3D Transmission Format Discussion

Neil Trevett
Khronos President
Vice President Mobile Content, NVIDIA
March 2015
Session Outline

- Outline of problem
  - Is that good outline of what AR developers actually need?

- What’s new/happening in the SDOs looking at this area since September 2014?
  - OGC, MPEG, Web3D, Khronos, W3C

- What next?
  - Discussion for planning next steps and potential collaboration
3D Needs a Transmission Format!

- Need to bridge the gap between tools and today’s GL based apps
  - Reduce duplicated effort in content pipelines
  - Common publishing format for content tools and services

- Browsers support loading of standard formats for many media types
  - With compression - for fast network transmission
  - With efficient client decode - especially important on mobile devices

- Why is 3D the last data type?
  - More degrees of freedom
  - Many different use cases and ways that 3D data may want to be transmitted

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>H.264</td>
<td>JPEG</td>
<td>?</td>
</tr>
<tr>
<td>Napster</td>
<td>YouTube</td>
<td>Facebook</td>
<td>!</td>
</tr>
</tbody>
</table>

An effective and widely adopted codec ignites previously unimagined opportunities for a media type
Building a Complete 3D Transmission Solution

- Three components need for complete solution
  - Scenegraph encoding (what is in the scene and how they relate)
  - REST3D APIs for Server/Client Negotiation (how do you want things to be sent)
  - 3D Asset payload - with variety of compression/streaming

1. Send Encoded scene graph

2. Use REST to negotiate asset selection, compression and streaming

3. Compress and Stream Assets
Rest APIs for 3D Asset Serving

• RESTful APIs - enables a web services approach
  - Eliminates need to bake all use cases into complex format
  - Formats and capabilities can be incrementally rolled out

• Streaming options queried and negotiated through REST e.g.
  - Request geometry in viewing frustum first - then the complete data set
  - Download geometry first then textures
  - Request highly compressed geometry (lossy) then lossless geometry
  - Request specific payload compression that app heuristically knows works well for its domain

• Rest3D Initiative - informal discussion group - chaired by Remi Arnaud
  - http://rest3d.wordpress.com/
  - How do we work with Rest3D?
  - Do we share goals and timeline requirements?
glTF = “JPEG for 3D”

• ‘GL Transmission Format’
  - Runtime asset format for WebGL, OpenGL ES, and OpenGL applications

• Compact representation for download efficiency
  - Binary mesh and animation data
  - Extension capability for future formats with compression and streaming

• Loads quickly into memory
  - JSON for scene structure and other high-level constructs
  - GL native data types require no additional parsing

• Full-featured
  - 3D constructs (hierarchy, cameras, lights, common materials, animation)
  - Full support for shaders and arbitrary materials

• Runtime Neutral
  - Can be created and used by any tool, app or runtime
glTF Internals

- **JSON to describe node hierarchy**
  - Platform neutral, run-time neutral, many processing libraries available
  - Node hierarchy refers to geometry, textures, materials, animations...

- **Texture Blob**
  - Can use existing standard image compression formats
  - ATSC tuned for textures and can be used directly by GPUs

- **Shaders Blob**
  - Use GLSL shaders today - compression is not normally an
  - Vendor portable PBR (physically-based rendering) materials is an interesting topic

- **Vertices Blob**
  - Default is native GL typed array format
  - 3D asset streaming and compression as options - MANY choices
Texture Compression is Key

- Texture compression saves precious resources
  - Network bandwidth, device memory space AND device memory bandwidth

- Developers need the same texture compression EVERYWHERE
  - Otherwise portable apps - such as WebGL need multiple copies of same texture

---

**Texture Compression**

- **DXTC/S3TC**
  - Windows

- **PVRTC**
  - iOS

- **ETC1**
  - Mandated in Android Froyo (400M devices)

- **ETC2 / EAC**
  - MANDATED in OpenGL ES 3.0
  - OpenGL 4.3

- **ASTC**
  - OpenGL ES 3.0 and OpenGL 4.3 extensions -> Core once proven

---

**Quality**

- **Royalty-free**
  - BUT only optional in ES.
  - Only 4bpp | 3 channel
  - No alpha support

- **ETC2 / EAC**
  - MANDATED in OpenGL ES 3.0
  - OpenGL 4.3
  - Royalty-free
  - Best quality.
  - Independent control of bit-rate and # channels
  - 1 to 4 channel
  - 1-8bpp in fine steps

- **ETC1**
  - Mandated in Android Froyo (400M devices)
  - Backward compatible with ETC1
  - ETC2: 4bpp | 3 channel
  - EAC: 4 (8) bpp | 1(2) channel
  - COMBINED: RGBA 8bpp | 4 channel
  - Does not have 1-2 bit compression WITH ALPHA

- **Pervasive Deployment**
  - 2008-2010
  - 2012-2013
  - 2014-
Khronos ASTC - Universal Texture Standard

- Adaptive Scalable Texture Compression (ASTC)
  - Quality significantly exceeds S3TC or PVRTC at same bit rate
- Industry-leading orthogonal compression rate and format flexibility
  - 1 to 4 color components: R / RG / RGB / RGBA
  - Choice of bit rate: from 8bpp to <1bpp in fine steps
- ASTC is royalty-free and so is available to be universally adopted
  - Will be mandated in Android M
- Now built into many GPUs
  - GPUs can render DIRECTLY from compressed textures in memory
Shaders -> Material Descriptions

• Physically Based Rendering (PBR) is hot topic in 3D graphics
  - Describe the *properties* of each material - not a literal shader
  - Easy for content creators
  - Realism of materials can scale to the platform capabilities

• Reach out to existing projects to include in transmission format?

NVIDIA MDL (material description language)  Open Shading Language


http://www.openshading.com/
Asset Compression/Streaming

• Need range of geometry compression options - Lossy vs. Lossless
  - Gzip is baseline for compression efficiency

• MANY Techniques and Experiments Underway: e.g.
  - MPEG has open-source and royalty-free mesh compression algorithms
    - 3D Mesh Coding Progressive Streaming (3DMC), Bones Based Animation (BBA)
  - X3D Binary Compression
  - WebGL-Loader used by Google body
  - X3DOM: Efficient binary meshes
  - OGC 3D Portrayal
    - REST APIs for request section of city model and send PNG of view if low-powered client
Many Ways of Compressing 3D Assets

Towards a 3D transmission format for the Web

Kristian Sons
kristian.sons@dfki.de
DFKI, Intel VCI

Philipp Stusallek
philipp.stusallek@dfki.de
DFKI, Intel VCI

May 24, 2013

ABSTRACT

Recent advances in the internet technology landscape raised the demand for efficient transmission of large amounts of binary data. This includes HTML5 video and audio streaming, data to do audio processing using the upcoming Web Audio API [11], and 3D related data to used by the WebGL API [18]. However, unlike video, audio or images, 3D data is very information and complexly structured. X3D’s main challenge is to establish efficient 3D data transmission for the Web.


X3D Binary Compression Capabilities and Plans

Motivation

Lots of work has already been accomplished using the X3D Efficient Binary Encoding (EIE) standard. X3D has numerous coherent approaches already available that meet author requirements for a general Web-based X3D data. These approaches are combined, extensive data compression with the priority of the XML format. We are now ready to demonstrate and standardize multiple interoperable improvements for a final standard.


- Annual review of goals and development capabilities at the Web3D Conferences and the X3D conference.


© Copyright Khronos Group 2015 - Page 12
glTF and Compression Benchmarking

• Benchmarking 3D compression formats for potential glTF extensions
  - Baseline is GZIP

• MPEG royalty-free Scalable Complexity compresses Meshes, Skinning, Animations
  - 3D Mesh Compression codec **MPEG-SC3DMC**
  - **Open3DGC** JavaScript and C/C++ implementation
  - WebGL Open3DGC loader [https://github.com/fabrobinet/glTF-webgl-viewer](https://github.com/fabrobinet/glTF-webgl-viewer)

• WebGL-loader is Google lightweight compression format for WebGL content

<table>
<thead>
<tr>
<th>Format</th>
<th>CAD Models (Mbytes)</th>
<th>3D Scanned Models (Mbytes)</th>
<th>MPEG dataset (Mbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>1310 (100%)</td>
<td>736 (100%)</td>
<td>600 (100%)</td>
</tr>
<tr>
<td>Gzip</td>
<td>336 (26%)</td>
<td>204 (28%)</td>
<td>157 (26%)</td>
</tr>
<tr>
<td>WebGL-loader</td>
<td>219 (17%)</td>
<td>117 (16%)</td>
<td>103 (17%)</td>
</tr>
<tr>
<td>WebGL-loader + Gzip</td>
<td>80 (6%)</td>
<td>38 (5%)</td>
<td>26 (4%)</td>
</tr>
<tr>
<td>Open3DGC</td>
<td>67 (5%)</td>
<td>22 (3%)</td>
<td>22 (4%)</td>
</tr>
</tbody>
</table>

Open3DGC is 5x-9x more efficient than GZIP and 1.2x-1.5x more efficient than webgl-loader
Transmission Format Liaison Catchup

- **Separable, individual projects at each SDO**
  - Projects normatively reference projects from other SDOs

- glTF drafts publicly available for feedback and requirements gathering

- glTF can refer to **ROYALTY FREE** mesh and animation compression technology from MPEG
  - Creates MPEG glTF loader

- glTF can refer to mesh and animation compression technology from Web3D
  - Create X3D/X3DOM glTF loaders

+ Other SDOs and companies
Discussion

- Is glTF a good substrate for the other SDOs?
  - JSON hierarchy description is not rocket science - it’s a simple substrate
  - Everyone was doing their own version anyway - saving everyone’s time
  - Even this simple step giving good loading speed-ups
  - Very open to input and feedback - getting positive feedback so far

- How do we drive definition of Rest APIs
  - Work with Rest3D or create different initiative? Where?

- How do we drive to closure on materials descriptions and texture formats
  - A number of industry solution available

- How do we propose/select streaming/compression technologies?
  - Need royalty-free ratification at an SDO to be candidate
  - Industry-wide call for proposals?
  - What is the selection process?
  - How many? Definitely > 1!

- Suggest Requirements -> Candidates -> Benchmarking -> Selection
  - Group with diverse perspectives, e.g. AR Community, could be ideal venue