

**K H R O N O S**  
G R O U P™



**Neil Trevett**  
**Vice President Mobile Ecosystem, NVIDIA**  
**President, Khronos Group**

# Khronos Connects Software to Silicon

Open Consortium creating  
ROYALTY-FREE, OPEN STANDARD  
APIs for hardware acceleration

Defining the roadmap for  
low-level silicon interfaces  
needed on every platform

Graphics, compute, rich media,  
vision, sensor and camera  
processing

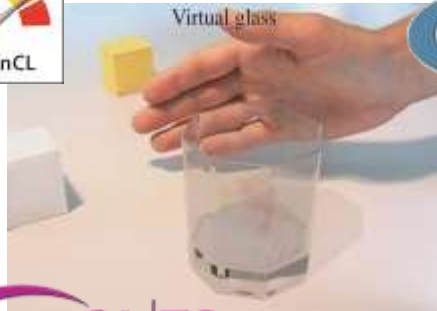
Rigorous specifications AND  
conformance tests for cross-  
vendor portability

*Acceleration APIs  
BY the Industry  
FOR the Industry*



**Well over a BILLION people use Khronos APIs  
Every Day...**

# Khronos Standards



## Visual Computing

- 3D Graphics
- Heterogeneous Parallel Computing



## 3D Asset Handling

- 3D authoring asset interchange
- 3D asset transmission format with compression



Over 100 companies defining royalty-free APIs to connect software to silicon



## Sensor Processing

- Vision Acceleration
- Camera Control
- Sensor Fusion






## Acceleration in HTML5

- 3D in browser - no Plug-in
- Heterogeneous computing for JavaScript



# 3D Needs a Transmission Format!

- **Compression and streaming of 3D assets becoming essential**
  - Mobile and connected devices need access to increasingly large asset databases
- **3D is the last media type to define a compressed format**
  - 3D is more complex - diverse asset types and use cases
- **Needs to be royalty-free**
  - Avoid an 'internet video codec war' scenario
- **Eventually enable hardware implementations of successful codecs**
  - High-performance and low power - but pragmatic adoption strategy is key

Audio	Video	Images	3D
MP3	H.264	JPEG	?
 <i>napster.</i>			!

An effective and widely adopted codec ignites previously unimagined opportunities for a media type

# What is glTF?

- **‘GL Transmission Format’**
  - Runtime asset format for WebGL, OpenGL ES, and OpenGL applications
- **Efficient Representation = Small Size AND Minimal Load Processing**
  - JSON for scene structure and other high-level constructs
  - Binary mesh and animation data
  - Little or no processing to drop glTF data into client application
- **Runtime Neutral**
  - Can be created and used by any app or runtime
- **Khronos is prototyping standards-based pipeline**
  - Conditioning of COLLADA assets into glTF for WebGL applications



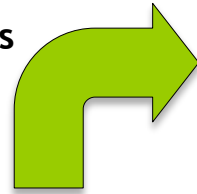


# Why Should I Care about glTF?

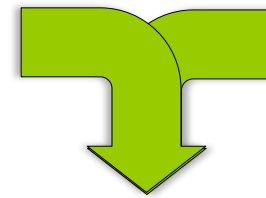
- **No open, comprehensive, vendor- and engine- neutral format for publishing 3D data to web and mobile apps**
  - COLLADA - designed for interchange not playback - XML-based
  - Autodesk FBX - proprietary
  - Engine-specific formats (e.g. Three.js) on their own trajectories
- **3D authoring wheel keeps getting re-invented**
  - Developers scramble to find exporters/importers, custom pipelines per-project
  - Basic features (e.g. cameras and lighting) are often hand-coded when they could and should be created by artists in a 3D tool
- **Need a common publishing format for 3D ecosystem**
  - A “JPEG for 3D”
  - Enable any asset server to serve assets to diverse clients
  - Scale market opportunities for 3D content, tools and services

# COLLADA and glTF Ecosystem

OpenCOLLADA  
Importer/Exporter  
and COLLADA  
Conformance Tests  
On GitHub



COLLADA2GLTF  
Translator



Other  
authoring  
formats

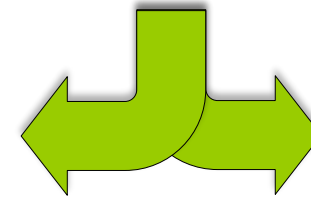
Web-based  
Tools



Pervasive WebGL  
deployment



Three.js glTF Importer.  
Rest3D initiative



# COLLADA Developments

- **COLLADA specifications now on GitHub!**
  - Encouraging accessibility and feedback
- **COLLADA 1.4.2**
  - Spec update in progress
  - Gathering feedback from community
  - Easier to be compliant by relaxing some aspects of the schema
- **COLLADA-CTS - Conformance Test Suite on GitHub**
  - Improved with recent Blender support
  - <https://github.com/KhronosGroup/COLLADA-CTS>
- **OpenCOLLADA - open source reliable importer/exporters**
  - OpenCOLLADA: newest exporter for Maya 2014
  - Exporter for Max 2015 is now working!
  - <https://github.com/KhronosGroup/OpenCOLLADA>





# COLLADA2GLTF

- COLLADA2GLTF open-source converter is gaining robustness and momentum
  - This pipeline is critical to easy glTF adoption
  - [Binaries](#) are available on GitHub for easy use
  - Significant contributions from Tom Fili, Analytical Graphics, Inc.
  - Community contributions are welcome!

# three.js glTF loader

- Three.js glTF loader project on Github
  - <https://github.com/KhronosGroup/glTF/tree/master/loaders/threejs>
- Most glTF features are already supported
  - Triangle meshes (optionally uses THREE.BufferGeometry for faster loading)
  - Materials - diffuse, specular, emissive, ambient, textures, environment maps
  - Cameras - perspective and orthographic,
  - Lights - ambient, spot, point, directional
  - Matrix transforms, Scene structure, Animation (Key frame/articulated only)
  - Shaders (uses “common profile” techniques - lighting models such as Phong and Lambert, and their parameters, are mapped to existing Three.js material types)
- Features TBD, in progress
  - Skinned animations and morphs
  - Arbitrary GLSL shaders (via THREE.ShaderMaterial)

# glTF Specification Process

- **Open specification; Open process**
  - Specification driven by COLLADA working group at Khronos
- **Spec work being done completely in the open**
  - All features backed up by multiple implementations in code
- **Specification**
  - <https://github.com/KhronosGroup/glTF/blob/master/specification/README.md>
- **glTF repo with sample code and specification JSON schema**
  - <https://github.com/KhronosGroup/glTF>

# glTF Status

- glTF 0.8 schema available on GitHub
  - [github.com/KhronosGroup/glTF](https://github.com/KhronosGroup/glTF)
  - Getting very close to glTF 1.0 - most likely no major breaking changes in 1.0
- Specified & implemented features:
  - Node hierarchy, Mesh, Animation (key frames, skinning)
  - Camera, Lights, Materials (Blinn, Phong, Lambert)
- Features on the way:
  - Morphing, Video, Mesh Compression, Cube maps
  - Multi-pass still being considered
- Next steps
  - Encourage support for more input formats
  - Direct glTF export from tools

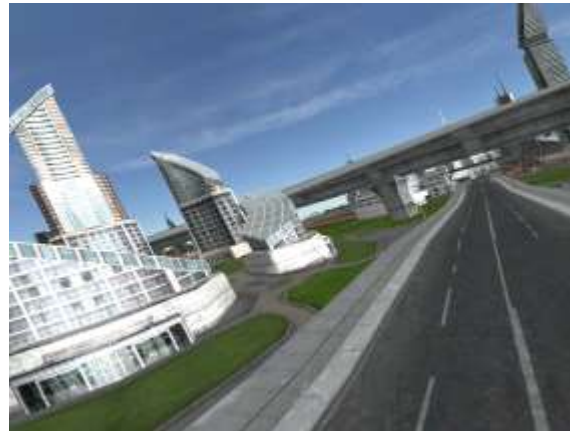
# glTF Adoption!

three.js loader

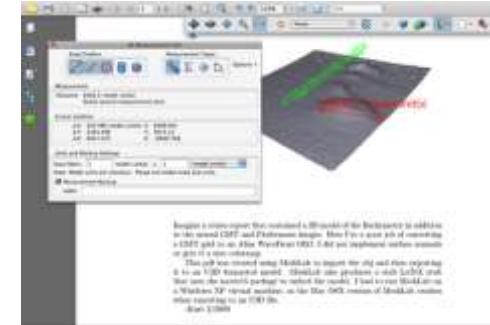


rest3d viewer

Cesium Engine



Montage Viewer



[a.mo.bee]

# glTF and 3D Asset Compression

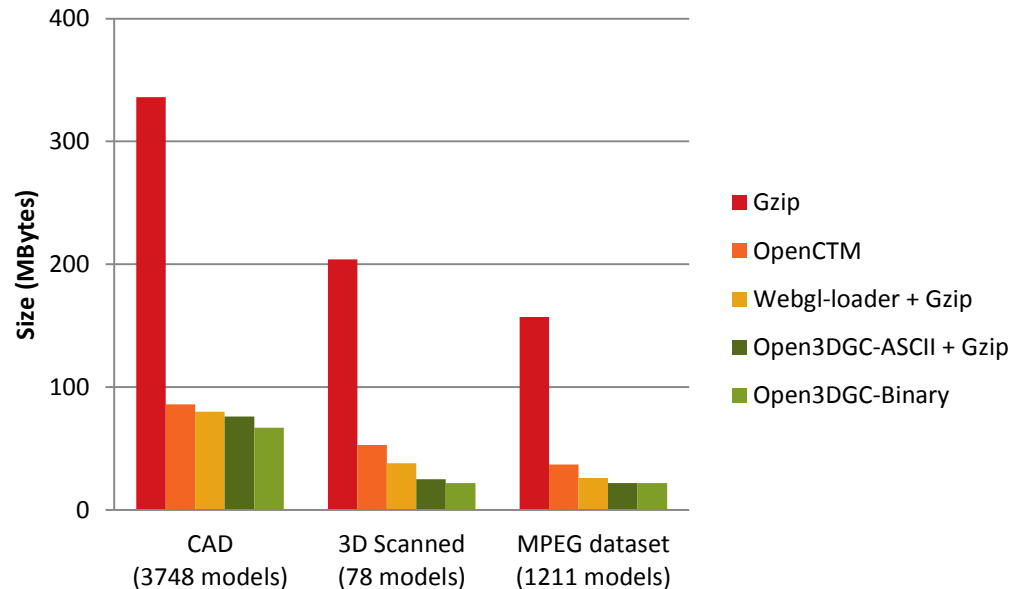
- Integrating and benchmarking 3D geometry compression formats with glTF
  - Baseline is GZIP
- Scalable Complexity 3D Mesh Compression codec [MPEG-SC3DMC](#)
  - Royalty-free graphics compression technology from MPEG (MIT License)
  - Open3DGC is efficient JavaScript and C/C++ implementation
  - Convertor using Open3DGC to compress 3D Meshes, Skinning, Animations
  - <https://github.com/amd/rest3d/tree/master/server/o3dgc>
- WebGL-loader is Google lightweight compression for WebGL content
- OpenCTM uses LZMA compression





# Initial Compression Results

- Compression Efficiency
  - [Gzip](#) (default level=6)
  - [OpenCTM](#) (default settings)
  - [Open3DGC](#) and [Webgl-loader](#)
    - Positions on 14 bits
    - Normals and texCoords on 10 bits



Open3DGC is 5x-9x more efficient than Gzip  
1.3x-2.4x more efficient than OpenCTM and  
1.2x-1.5x more efficient than webgl-loader

# 3DGC Decode Times

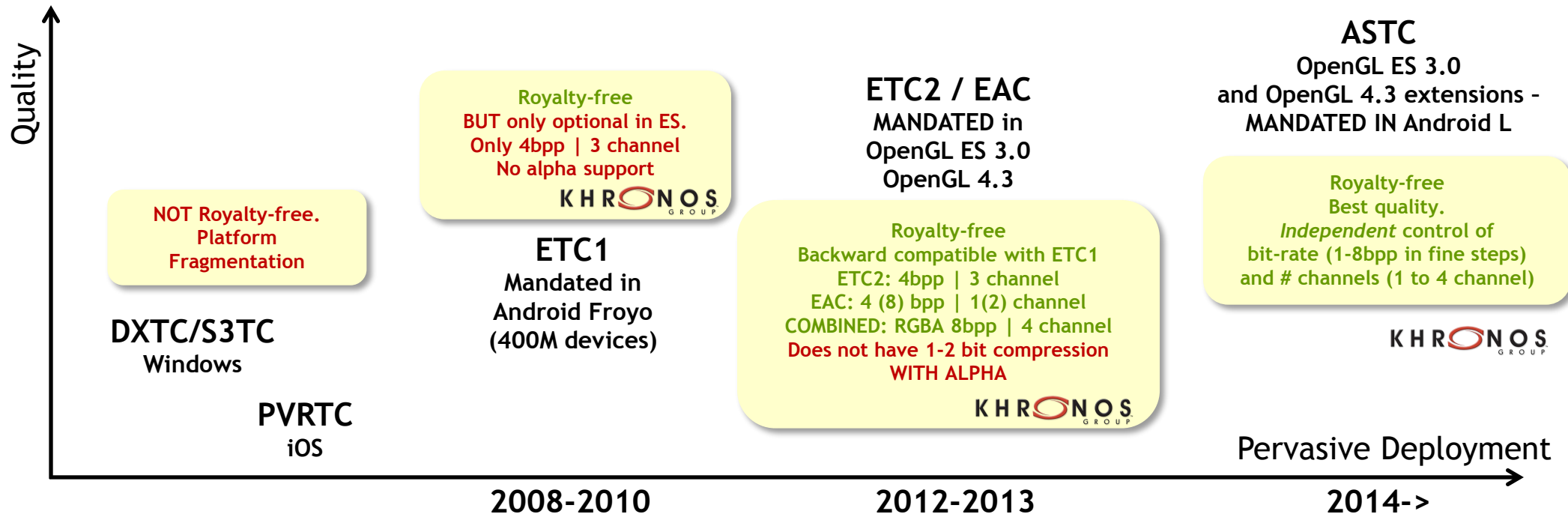
- JavaScript Decoding Speed
- Desktop machine
  - Windows® 64-bit, 8GB RAM, Chrome
  - AMD Phenom™ II X4 B95 CPU @ 3.0GHz
- Smart phone
  - Samsung Galaxy S4
  - Android 4.2.2
  - Chrome

	Number of triangles	Desktop decoding time (ms)	Smart phone decoding time (ms)
“Hand”	100K	130	1045
“Dilo”	54K	85	768
“Octopus”	34K	65	457

Decoding speed will become even more critical with dense 3D meshes generated by 3D digitization technologies (e.g. 3D scanners)  
3D Codec can be accelerated by WebCL Kernels or (eventually) hardware

# 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



# 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**
  - Shipping as OpenGL/OpenGL ES extension today for industry feedback



Original  
24bpp



8bpp



ASTC Compression  
3.56bpp



2bpp

# With Consumer Capture - 3D Will Go Social!

