Geometry, Textures, and Workflow - Optimizing glTF

Tuesday, August 8, 2023 | Khronos BOF Series at SIGGRAPH 2023

Max Limper (DGG) | Adam Morris (Cesium)
Pawel Nikiel (DGG) | Andreas Vasilakis (Phasmatic)
Paweł Nikiel

- CTO @ DGG, Creators of RapidCompact
- Contributor to 3D Commerce & 3D Formats WGs
- Over 10 years of experience in 3D art, VR & AR development
- Focused on pipelines, automation & scale
- Everything interactive
- Meet me at the DGG booth (#837)
Many Publishing Targets

Cp. Khronos 3D Commerce Asset Creation Guidelines v1.0 (section: Publishing Targets)
# Hardware & Bandwidth Constraints

<table>
<thead>
<tr>
<th>Publishing Target</th>
<th>Max. Target File Size</th>
<th>Max. Target Triangle Count</th>
<th>Target (Max) Number of Draw Calls</th>
<th>Max. Target Bitmap resolution, to meet bandwidth requirements (JPG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Item Mobile AR or 3D Web Catalogue View</td>
<td>3MB</td>
<td>150,000</td>
<td>&lt;20 (500)</td>
<td>2K</td>
</tr>
<tr>
<td>Banner Ad View</td>
<td>500KB</td>
<td>30,000</td>
<td>&lt;5 (100)</td>
<td>512</td>
</tr>
<tr>
<td>Web-based Planning Tool (recommendations for one out of multiple items)</td>
<td>1MB</td>
<td>40,000</td>
<td>&lt;5 (50)</td>
<td>1K</td>
</tr>
<tr>
<td>Single-Item Desktop 3D Web View</td>
<td>3MB</td>
<td>250,000</td>
<td>&lt;100 (800)</td>
<td>2K</td>
</tr>
<tr>
<td>Offline Rendering</td>
<td>No Limit</td>
<td>No Limit</td>
<td>No Limit</td>
<td>No Limit</td>
</tr>
</tbody>
</table>

3D Asset Optimization Workflow

1. Scene Flattening
2. Optimize Animations
3. Optimize Geometry
4. UV Unwrapping
5. Baking
6. Compression

Images: Wayfair
Clustering

What?
- Basic optimization technique originating in the 80’s
- Fast performance
- Can be parallelized

When?
- Huge datasets, dense meshes
- Real-time computation
- Need to save memory

How?
- Divide space into fixed-size grid
- Find representative position based on all vertices in each cell
- Merge all vertices to the position
Clustering
Decimation

What?
- Well established technique
- Sequential, difficult to parallelize
- Possible to balance performance vs accuracy depending on needs

When?
- Good input topology or 3D scans
- Specific number of tris expected
- Preservation of data needed (e.g. UVs)
- Not suited for high genus or noisy mesh
- Doesn’t work well with overlaid layers

How?
- Calculate quadric error (or edge length)
- Find new positions with smallest offset from original curvature
- Easy to stack levels of simplification
- New method -> probabilistic quadrics
Decimation

Optimized with RapidCompact
Remeshing

What?
- Can mean two things
- Still being perfected
- Can be quite slow due to shrink wrapping process, accuracy needed

When?
- Bad input
- High genus, complex assets
- Stacked layers
- Invisible geometry to remove
- Difficult to preserve data (e.g. UVs)

How?
- Create voxel grid
- Refine
- Shrink wrap
- Finding closest points is risky
- Decimate at the end of process
Remeshing

Optimized with RapidCompact
Animation optimization

Animation curves
- Possible to simplify
- For interchange it’s common to convert to high density linear curve due to different implementations of splines and others
- Usually very compressible data (10:1 ratio possible)
- Compression currently missing in glTF :(

Additionally:
- Rigid animations - possible to simplify meshes
- Skinned animations - additional steps needed, need to rebind new mesh to existing skeleton
- Skeleton simplification - need to redirect weights to remaining bones
- Shape keys (morph targets) - after simplification need to figure out which vertex is affected by which target

IronMan CC-BY courtesy of 9A Films / Nihar Arora
Animation optimization

#Polygons: 640K / 20K / 5K

IronMan CC-BY courtesy of 9A Films / Nihar Arora
Max Limper

- Co-Founder and CEO @ DGG, Creators of RapidCompact
- Background in CS & 3D Graphics
- glTF Enthusiast & Contributor
- Meet me at the DGG booth (#837)
Optimizing Textures: Downscaling & Atlasing

**Input (10.90MB):**
- 5 materials (leather, mask, glasses, stand, misc)
- 15 different maps (5 x base color / ORM / normals)

**Output (2.59MB):**
- 2 materials (opaque, transparent)
- 6 different maps (2 x base color / ORM / normals)
- 2K/1K resolution for opaque 256/128 for transparent
- Adaptive to 3D size, ORM also has smaller resolution

- Faster transmission & less GPU memory
- Faster rendering: less draws & texture switches
- Compressing individual texture maps -> Later (Andreas)
Defining Realistic Materials: glTF Extensions

Extensions used:

KHR_materials_transmission, KHR_materials_ior,
KHR_materials_volume (glass)
KHR_materials_iridescence (iridescence effect)
Optimizing Materials with KHR_materials_variants

- Enables a single self-contained GLB file for all colorways of a product
- Enables sharing of geometry across variants in one file
- We can share textures (e.g., normals) across variants, too
Most “Aggressive” Optimization: Drop Textures

- Use representative untextured materials instead of textures
- For ultra-compact, fast-loading previews
- Aim: super-fast loading to show something to users of 3D Web / AR (while high-resolution content is loading in the background)

Textured, 6.50MB

Untextured (“DropTextures”), 28KB

Example generated using https://rapidcompact.com
Andreas Vasilakis

- Co-Founder and CEO @ Phasmatic
  - Photorealistic 3D eCommerce
  - WebGL/gltF Ray Tracing
- PhD in CS & 3D Graphics
- Postdoc Researcher & Adj. Professor
Creating, Optimizing & Validating 3D Content

DO NOT MISS Eric’s talk about “Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR” at 11am

© The Khronos® Group Inc. 2023 - Page 20
Creating, Optimizing & Validating 3D Content

Creation → Compression → Quality Assurance → Delivery

DO NOT MISS Eric’s talk about “Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR” at 11am
Creating, Optimizing & Validating 3D Content

DO NOT MISS Eric’s talk about “Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR” at 11am
Command Line Interfaces for Compression
- Not direct visual comparison
- Non-intuitive & user-friendly for artists
- Is compliant with use case specifications?

DO NOT MISS Eric’s talk about “Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR” at 11am
Creating, Optimizing & Validating 3D Content

- Command Line Interfaces for Compression
  - Not direct visual comparison
  - Non-intuitive & user-friendly for artists
  - Is compliant with use case specifications?

- Can we do better?
  - Interactive online tools to help artists!

DO NOT MISS Eric’s talk about “Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR” at 11am
Enabling Efficient Geometry Size Reduction

- DRACO provides advanced compression for mesh geometry

![DRACO logo]

![Graph showing mesh file compression]

Typical Draco compression ratios
Enabling Efficient Geometry Size Reduction

- **DRACO** provides advanced compression for mesh geometry

- **MESHOPT** provides compression and fast decoding for geometry, morph targets, and animations
Enabling Compact and Efficient glTF Textures

KTX Artist Guide
Khronos glTF-Compressor is here!

- Interactive tune texture compression size and quality
  - Intuitive Texture Selection
    - Complete image information display
    - Default selections based on image use
  - Flexible Texture Compression
    - JPG, PNG, WebP as well as KTX
    - Advanced KTX compression options
  - 2D and 3D Side-by-Side Live Texture Comparison
  - Optimized Assets Export

- Extends the capabilities of the Khronos open-source glTF Sample Viewer
- Future release will include Geometry Compression capabilities

Live Tool: glTF-Compressor-Release - Github: glTF-Compressor
Khronos glTF-Compressor: Live Demo

Live Tool: glTF-Compressor-Release - Github: glTF-Compressor
Khronos glTF Asset Auditor

- Quickly check glTF asset for a specific use case, defined by an audit profile

Web Browser

Command Line Interface

Adam Morris
- Staff Software Engineer @ Cesium
- M.S. in Human Computer Interaction from Iowa State University
- glTF enthusiast and Khronos 3D Formats Working Group member.
- Comprehensive XR background and experience.
Intelligent Organization

How do we get from a single model, to handling highly detailed scenes?
as a case study for glTF Optimization.
• Open standard for streaming massive heterogeneous 3D geospatial data
  - Terrain & imagery, 3D buildings, photogrammetry, point clouds, BIM models, interiors, etc.
  - Multiple source data types, one runtime format

• Visualization + analysis

• Combine:
  - Flexible spatial data structure in JSON
  - “Runtime ready” binary tile formats
  - glTF for 3D model formats
  - Vertex/polygon-level metadata
  - Declarative styling

• Started by Cesium in 2015
• OGC Community Standard since 2019
3D Tiles can be used for...

- point clouds...
- geospatial...
- CAD/BIM...
- highly detailed models.
3D Tiles: Primer

- Open standard for streaming massive heterogeneous 3D geospatial data
  - Terrain & imagery, 3D buildings, photogrammetry, point clouds, BIM models, interiors, etc.
  - Multiple source data types, one runtime format

- Visualization + analysis
- Combine:
  - Flexible spatial data structure in JSON
  - “Runtime ready” binary tile formats
  - glTF for 3D model formats
  - Vertex/polygon-level metadata
  - Declarative styling

- Started by Cesium in 2015
- OGC Community Standard since 2019
Octrees

- Traditional method for subdivision in 3D graphics.
- Partitions space by recursive subdivision.
- Properties
  - Three-dimensional
  - Extends a quadtree by using three orthogonal planes to subdivide a space into 8 children.
- Supports (where needed!)
  - Non-uniform subdivision
  - Tight bounding volumes
  - Overlapping children
Quadtrees

- Partitions space by recursive subdivision.
- Properties
  - Two-dimensional
  - Each node has exactly 4 children.
  - Regions may be square, rectangular, or even arbitrary shapes.
- Further optimizations
  - Bounding volumes around each child.
    - Efficient for sparse data sets.
  - "Loose" quadtrees
    - Children overlap, but coherence is preserved.
    - Useful to prevent specific 3D models from being split across partitions.
Quadtrees

Quadtree with tight bounding volumes

Quadtree with overlapping partitions

**k-d trees**

- Useful for subdividing sparse and non-uniform data sets.
- **Properties**
  - Binary tree where every node is a \( k \)-dimensional point in space.
  - Every node implicitly generates a hyperplane that divides space into two partitions called "half-spaces".
  - Has non-uniform subdivision, allowing a more balanced tree for sparse and non-uniform data sets.
- "The Curse of Dimensionality"
  - Largely irrelevant, since \( k \) is generally 2 or 3 in graphics.
- **Further optimizations**
  - Multi-way \( k \)-d trees

[http://www.crs4.it/vic/cgi-bin/bib-page.cgi?id=%27Goswami:2013:EMF%27](http://www.crs4.it/vic/cgi-bin/bib-page.cgi?id=%27Goswami:2013:EMF%27)

*k-d tree. Note the non-uniform subdivision*
You can use a combination of methods!
Combining methods

*Bounding Volume Subdivision*

- **Root**
- **Quadtree**
- **Octree**
Combining methods

*Choose the right subdivision*

Octree in local coordinates
3D Tiles from terrestrial Lidar scan

Quadtree in global coordinates
NYC buildings


Data source: Trimble
Combining methods

Choose the right subdivision
Combining methods

Handling coordinates: Box Bounding Volumes

(level, x, y, z)
Combining methods

Handling coordinates: Region Bounding Volumes

- tile
  - boundingVolume
  - geometricError
  - refine
  - content
    - boundingVolume (box, region, or sphere)
    - uri
      - Separate file with tile contents, streamed on demand
  - children[]
Combining methods

Geometric Error

The geometric error is measured in meters, comparing the simplified geometry to the real geometry:

The screen space error (SSE) is measured in pixels:

- SSE: about 4 pixels
- SSE: about 2 pixels
- SSE: about 1 pixel
Combining methods

Mixing Levels of Detail
Putting it all together

<table>
<thead>
<tr>
<th>Description</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liverpool Source</td>
<td>1.6GB</td>
</tr>
<tr>
<td>Uncompressed 3D Tileset</td>
<td>1.2GB</td>
</tr>
<tr>
<td>With KTX2 Compression only</td>
<td>1.0GB</td>
</tr>
<tr>
<td>With Draco compression only</td>
<td>729MB</td>
</tr>
<tr>
<td>With KTX2 and Draco compression</td>
<td>482MB</td>
</tr>
</tbody>
</table>
Notable glTF Extensions

● Useful for glTF optimization
  ○ KHR_draco_mesh_compression
  ○ KHR_mesh_quantization
  ○ EXT_mesh_gpu_instancing
  ○ EXT_meshopt_compression
  ○

● Notable mentions
  ○ MSFT_lod
    ■ Alternative method for adding Level of Detail to glTF files.
    ■ Lacks widespread support.
Where to learn more?

- **gltF**
  - [https://github.com/KhronosGroup/gltF](https://github.com/KhronosGroup/gltF)

- **3D Tiles**
  - You can use this for more than geospatial!
  - [https://github.com/CesiumGS/3d-tiles](https://github.com/CesiumGS/3d-tiles)
  - Community Projects
    - Vulkan Scene Graph loader for 3D Tiles
      - [https://github.com/timoore/vsgCs](https://github.com/timoore/vsgCs)
    - NASA-AMMOS 3D Tiles loader for three.js
      - [https://github.com/NASA-AMMOS/3DTilesRendererJS](https://github.com/NASA-AMMOS/3DTilesRendererJS)
  - Reference Cards

- **Cesium**
  - [https://cesium.com/learn/](https://cesium.com/learn/)

- **CesiumJS, Cesium for Unreal, Cesium for Unity, and Cesium for Omniverse are all open-source!**
  - CesiumJS: [https://github.com/CesiumGS/cesium](https://github.com/CesiumGS/cesium)
  - Cesium for Unreal: [https://github.com/CesiumGS/cesium-unreal](https://github.com/CesiumGS/cesium-unreal)
  - Cesium for Unity: [https://github.com/CesiumGS/cesium-unity](https://github.com/CesiumGS/cesium-unity)
  - Cesium for Omniverse: [https://github.com/CesiumGS/cesium-omniverse](https://github.com/CesiumGS/cesium-omniverse)
Ask the speakers

Max Limper (DGG)

Pawel Nikiel (DGG)

Adam Morris (Cesium)

Andreas Vasilakis (Phasmatic)