SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

Geometry, Textures, and Workflow - Optimizing glTF

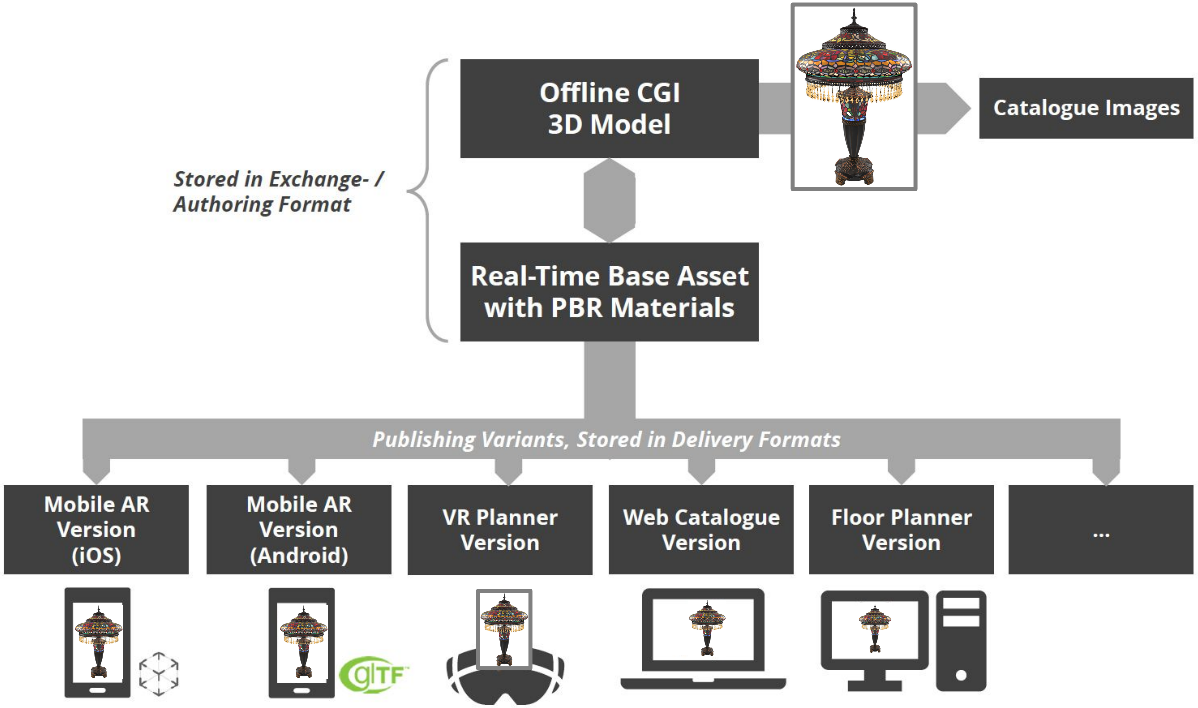
Tuesday, August 8, 2023 | Khronos BOF Series at SIGGRAPH 2023

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



Offline CGI 3D Model

Catalogue Images

Stored in Exchange- / Authoring Format

Real-Time Base Asset with PBR Materials

Publishing Variants, Stored in Delivery Formats

Mobile AR Version (iOS)

Mobile AR Version (Android)

VR Planner Version

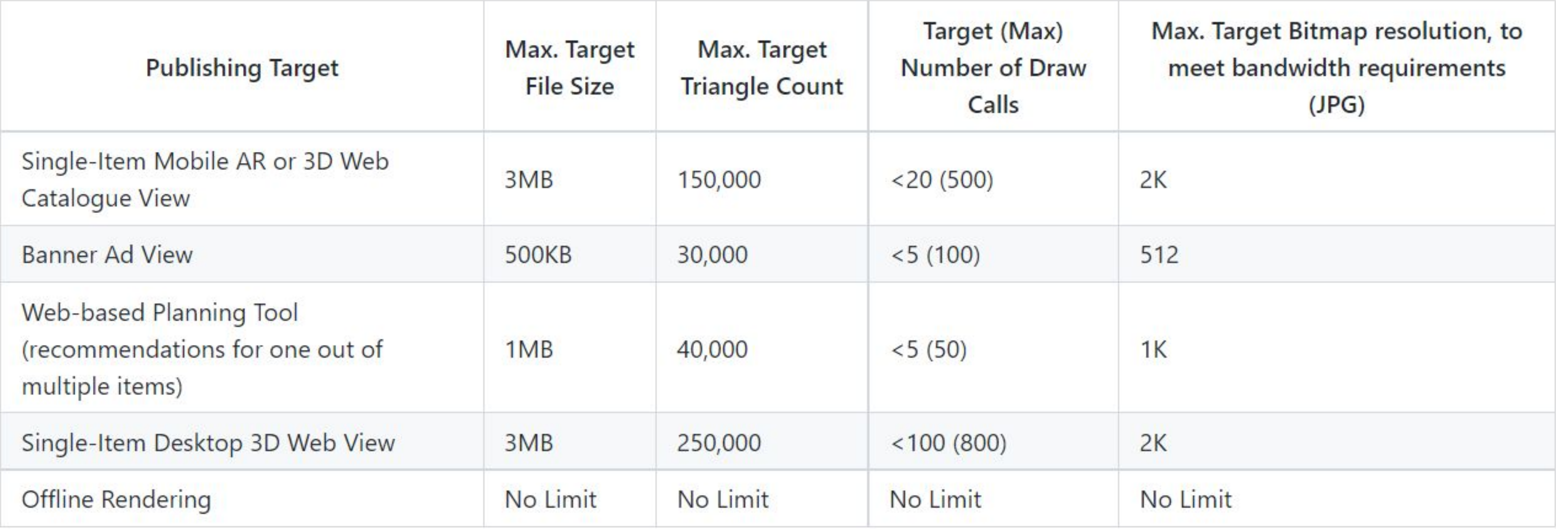
Web Catalogue Version

Floor Planner Version

...

Cp. **Khronos** [3D Commerce Asset Creation Guidelines](#) **v1.0** (section: [Publishing Targets](#))

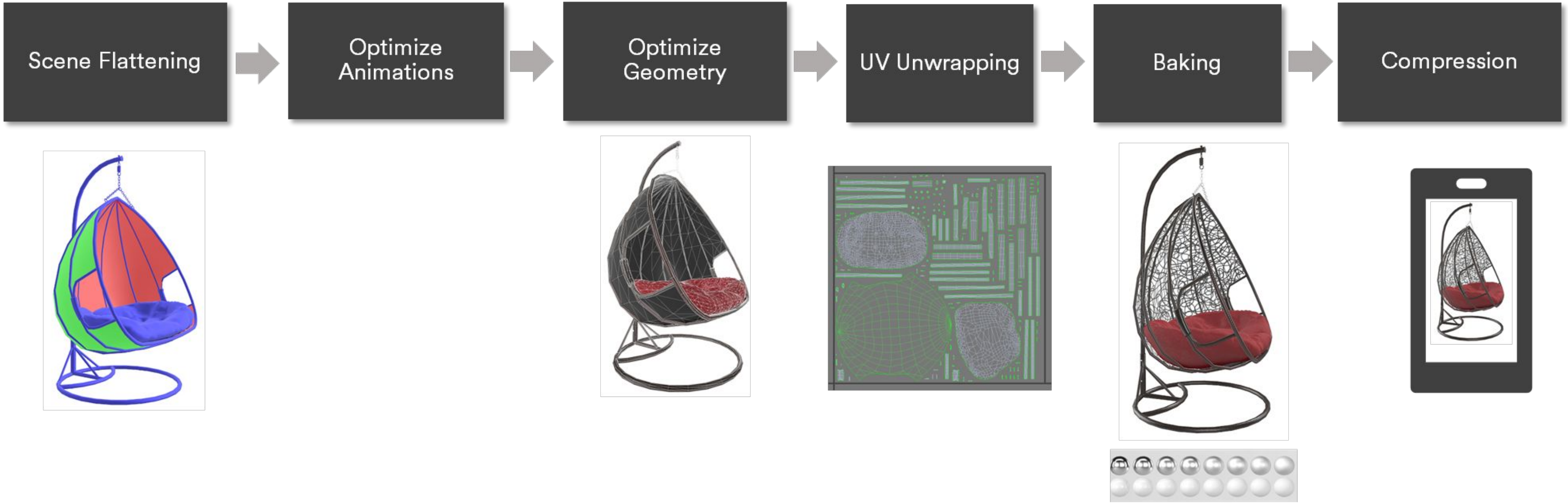
Hardware & Bandwidth Constraints



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

Cp. **Khronos** 3D Commerce Asset Creation Guidelines **v1.0** (section: Publishing Targets)

3D Asset Optimization Workflow



Scene Flattening → Optimize Animations → Optimize Geometry → UV Unwrapping → Baking → Compression

Images: Wayfair

Clustering

What?

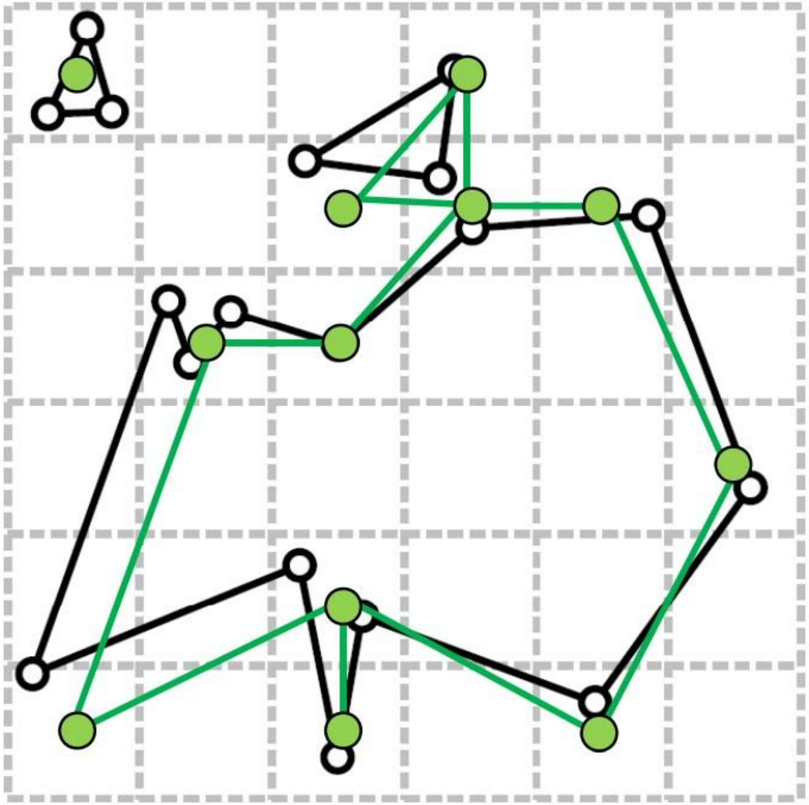
- Basic optimization technique originating in the 80’s
- Fast performance
- Can be parallelized

When?

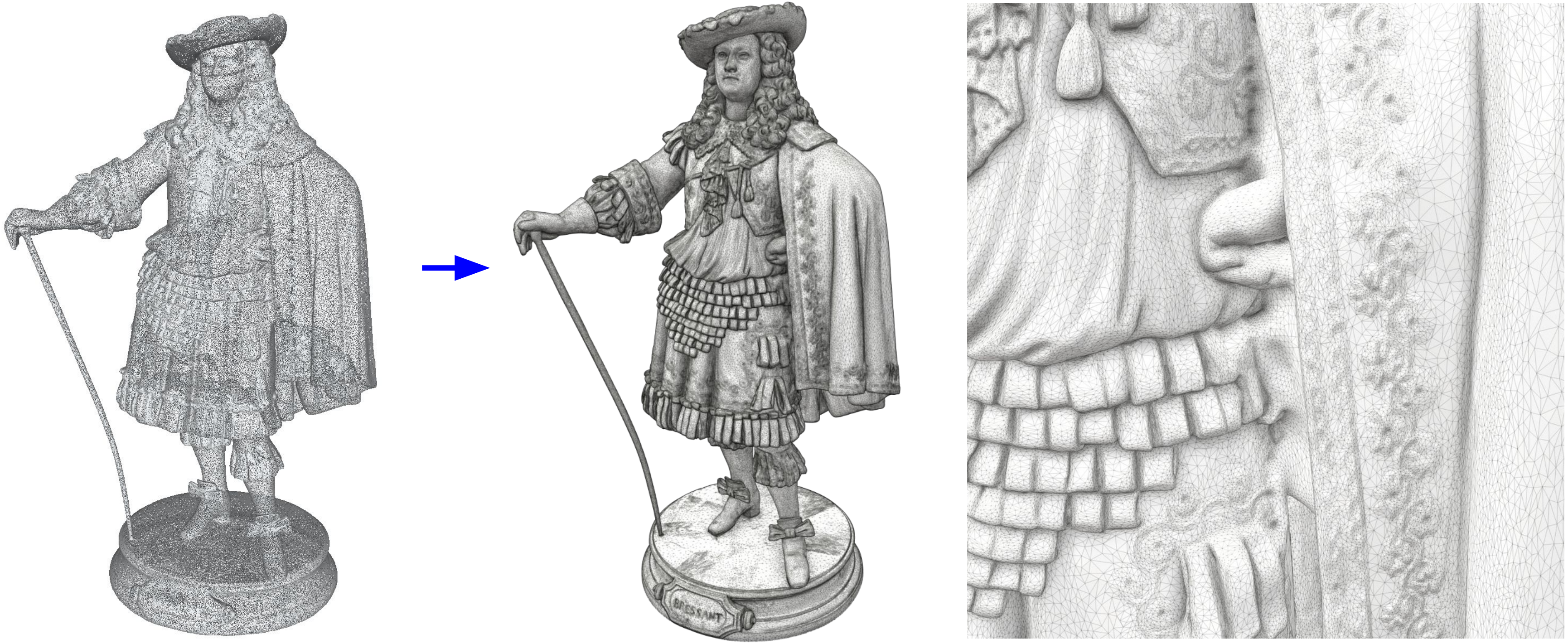
- Huge datasets, dense meshes
- Real-time computation
- Need to save memory

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?

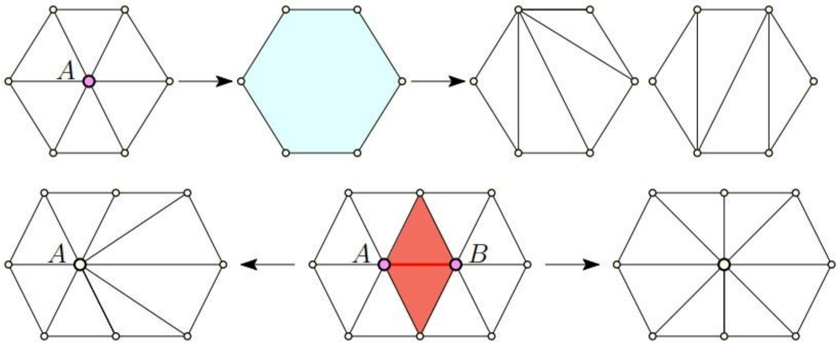
- Well established technique
- Sequencial, difficult to parallelize
- Possible to balance performance vs accuracy depending on needs

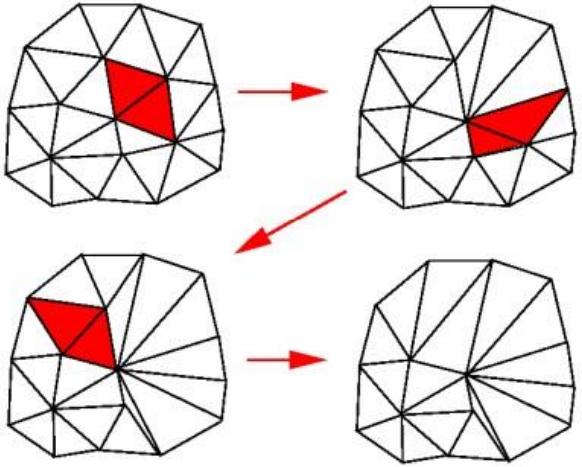
When?

- 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

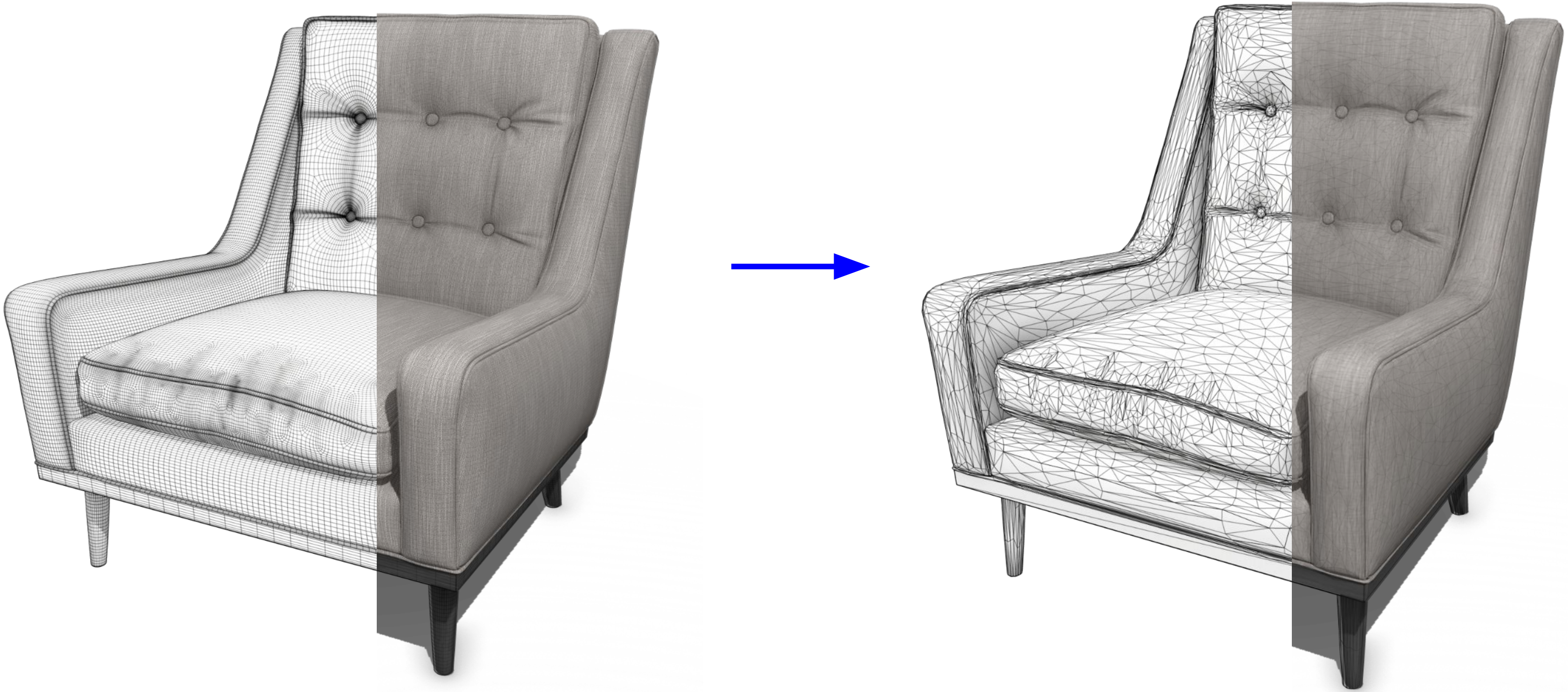
How?

- 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?

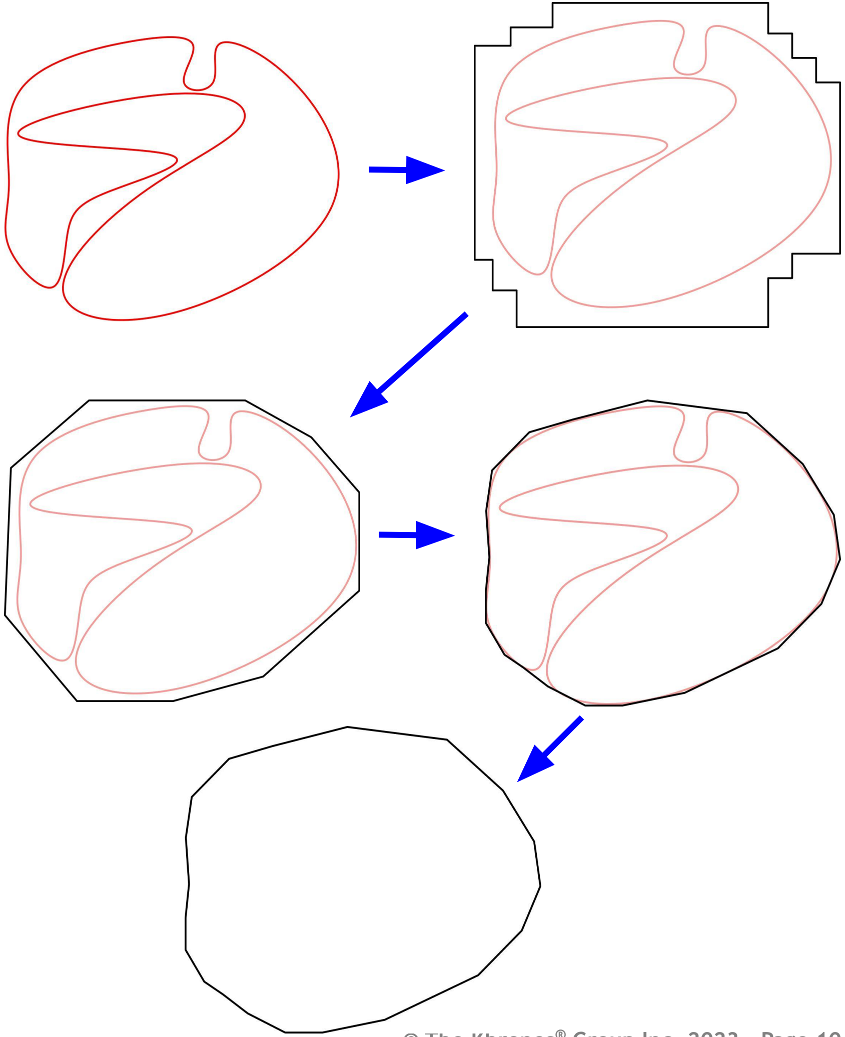
- Can mean two things
- Still being perfected
- Can be quite slow due to shrink wrapping process, accuracy needed

When?

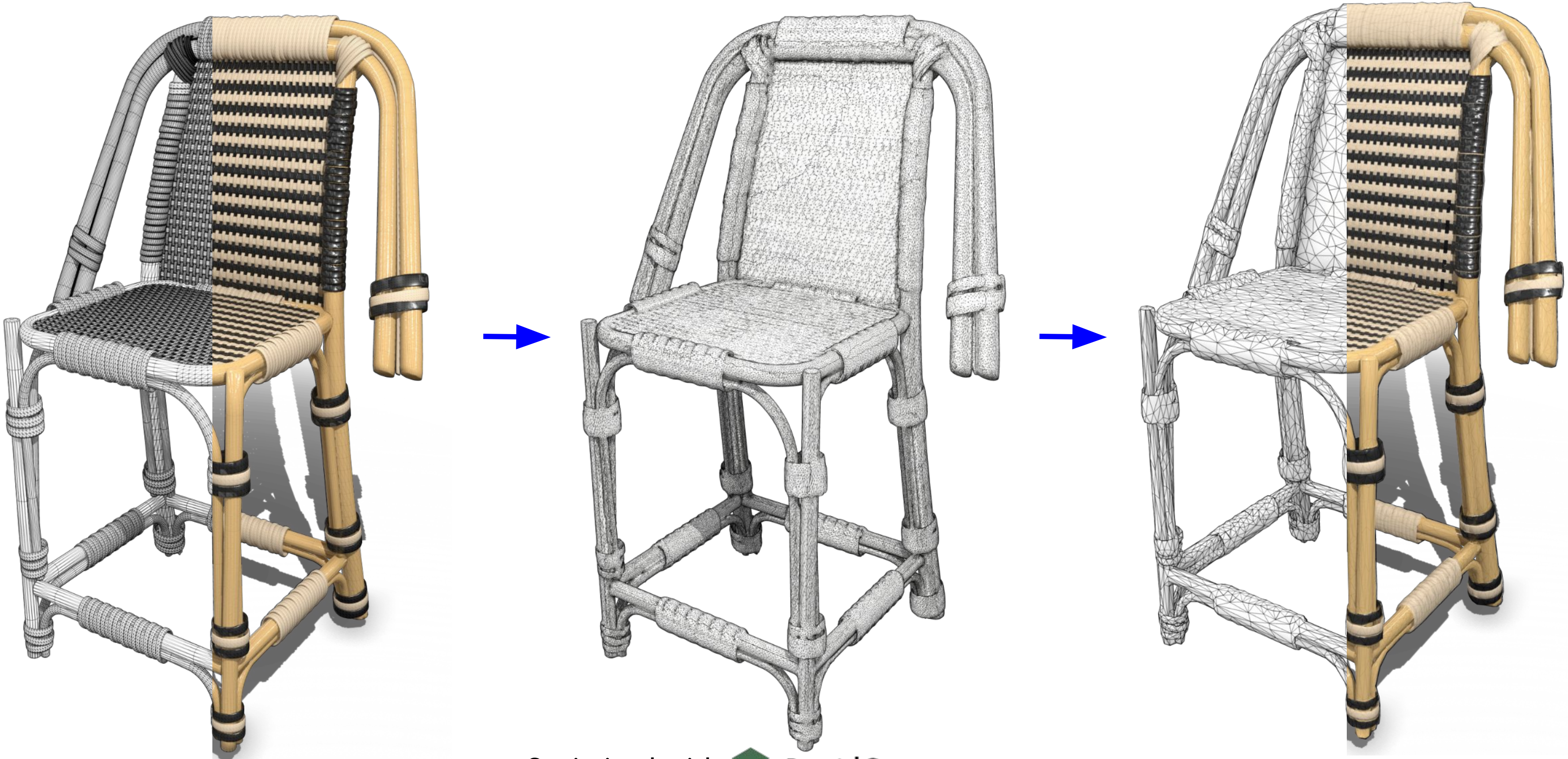
- Bad input
- High genus, complex assets
- Stacked layers
- Invisible geometry to remove
- Difficult to preserve data (e.g. UVs)

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

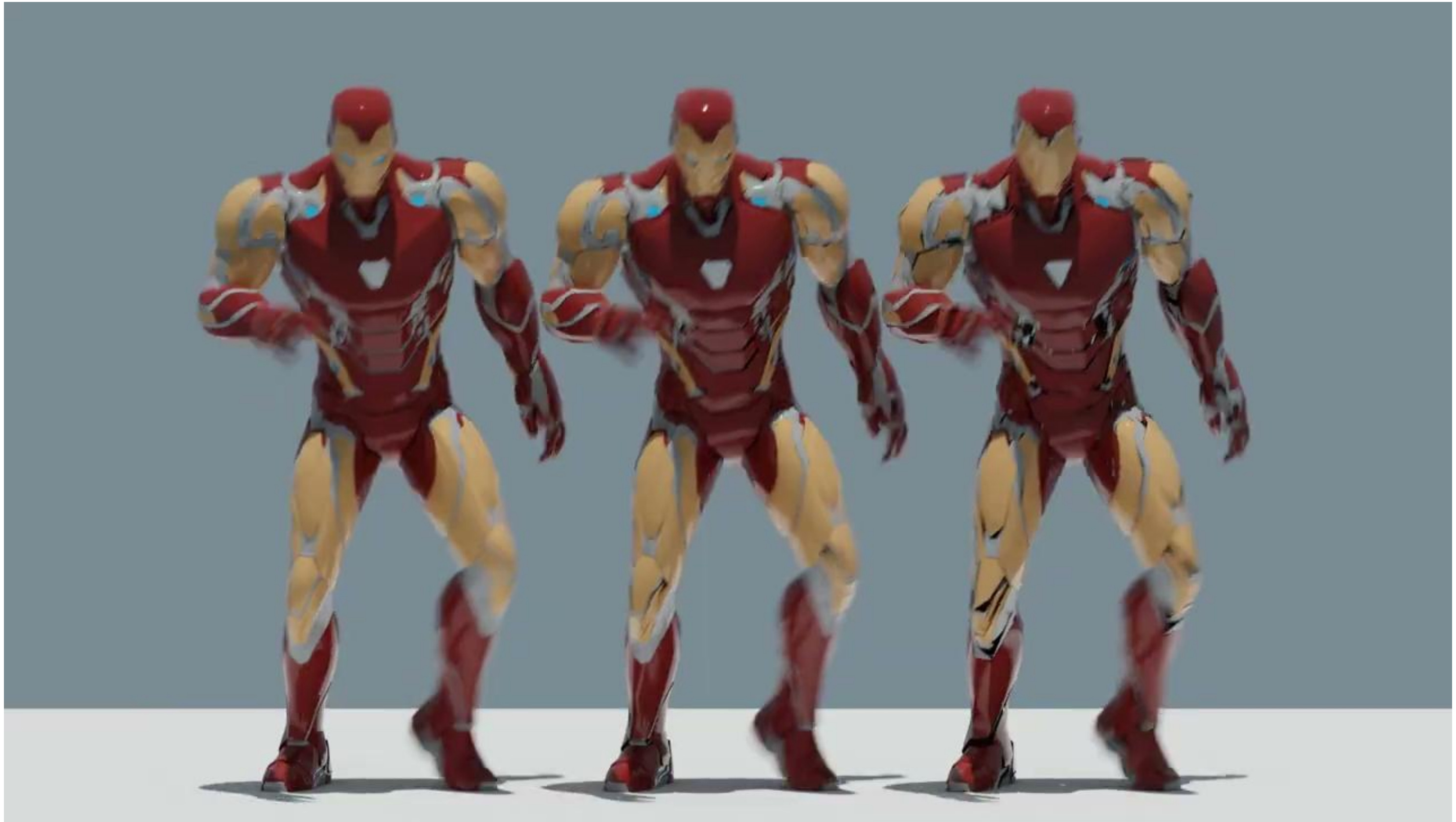
- Possible to simplify
- For interchange it's common to convert to high density linear curve due to different implementations of splines and others
- Usually very compressible data (10:1 ratio possible)
- Compression currently missing in glTF :(

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)

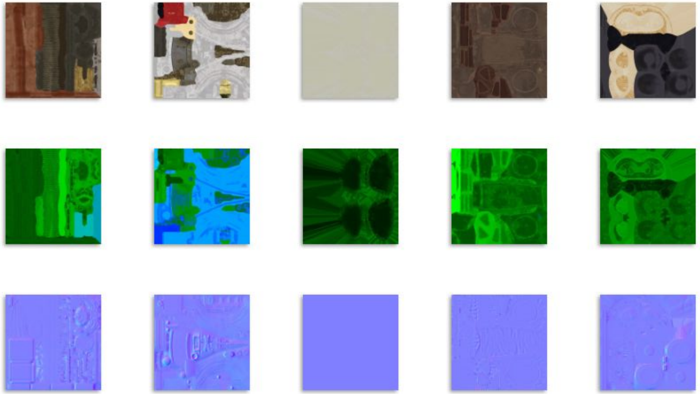
Optimizing Textures: Downscaling & Atlasing

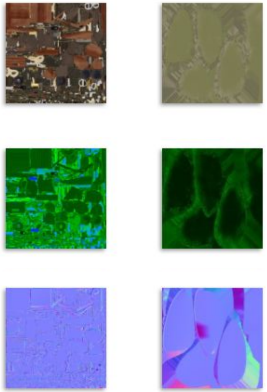
Input (10.90MB):

- 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**



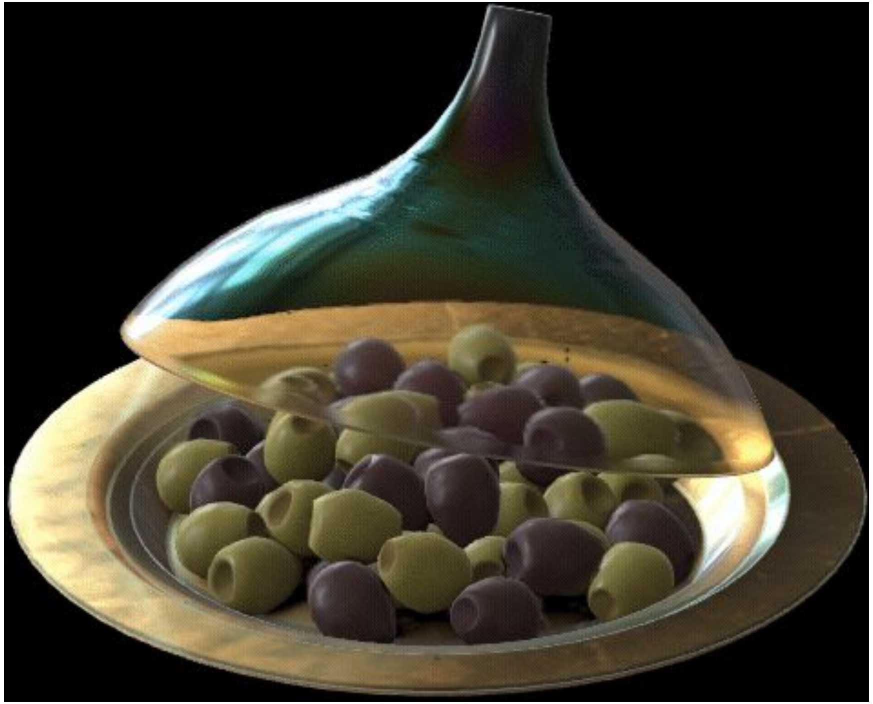


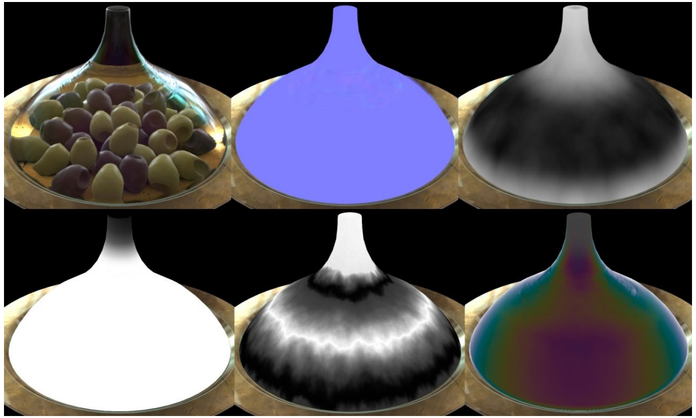
- Faster **transmission & less GPU memory**
- Faster **rendering:** less draws & texture switches
- Compressing **individual** texture maps -> Later (Andreas)

Example generated using https://rapidcompact.com

RapidCompact

Defining Realistic Materials: glTF Extensions



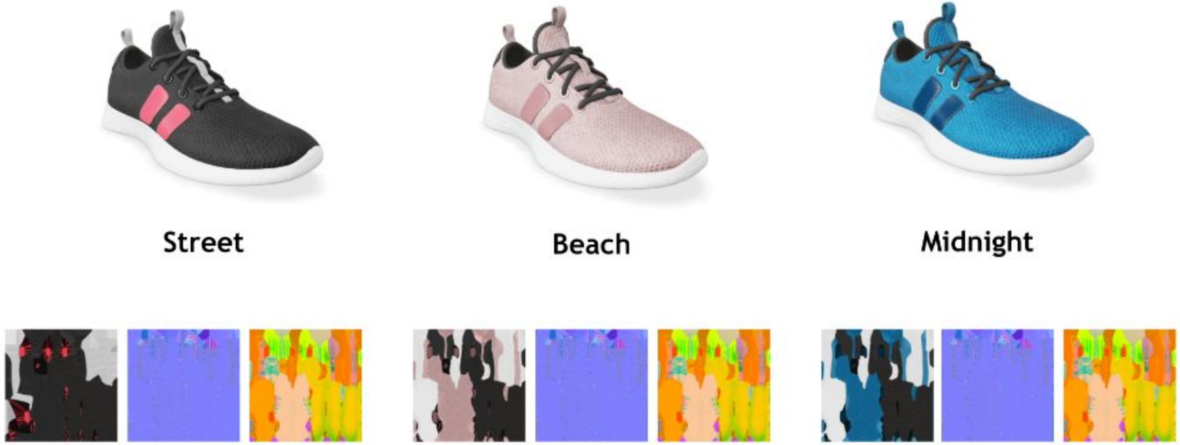


Extensions used:

`KHR_materials_transmission`, `KHR_materials_ior`, `KHR_materials_volume` (glass)

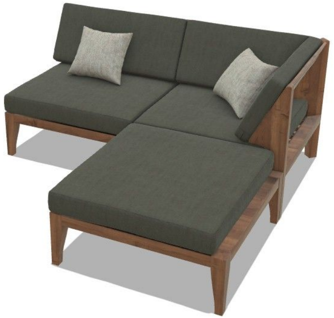
`KHR_materials_iridescence` (iridescence effect)

Optimizing Materials with KHR_materials_variants

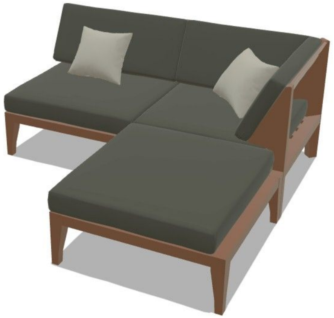


- Enables a single self-contained GLB file for all colorways of a product
- Enables sharing of **geometry** across variants in one file
- We can share **textures** (e.g., normals) across variants, too

Most "Aggressive" Optimization: Drop Textures



Textured, 6.50MB



Untextured ("DropTextures"), 28KB

- 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)

Example generated using https://rapidcompact.com

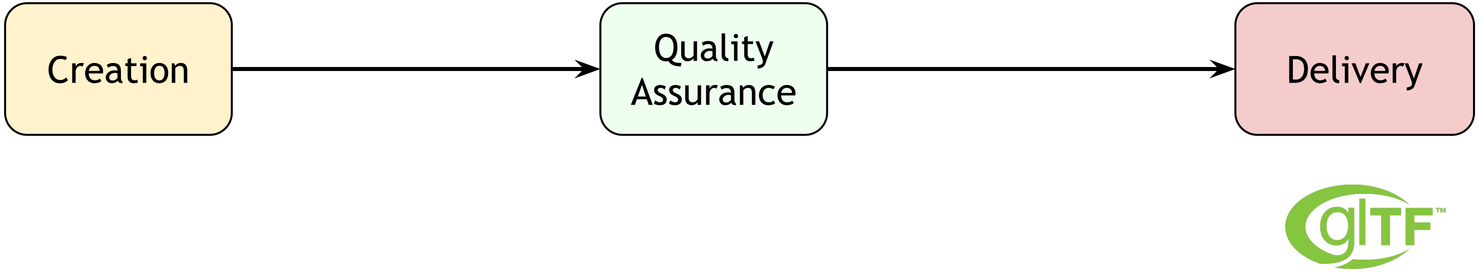
RapidCompact

Andreas Vasilakis

- Co-Founder and CEO @ Phasmatic
  - Photorealistic 3D eCommerce
  - WebGL/glTF Ray Tracing
- PhD in CS & 3D Graphics
- Postdoc Researcher & Adj. Professor

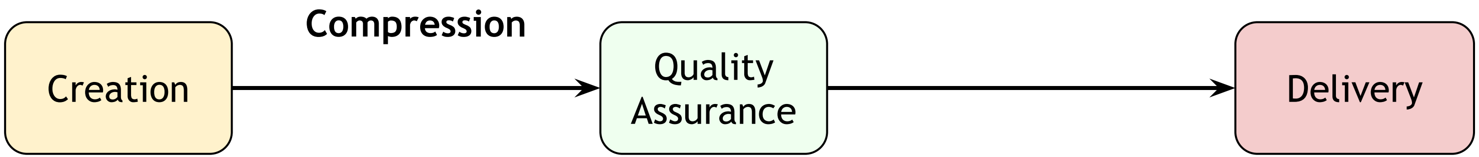
PHASMATIC

Creating, Optimizing & Validating 3D Content



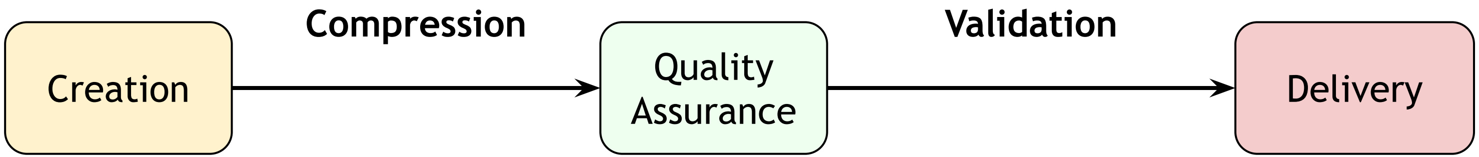
DO NOT MISS Eric’s talk about *“Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR”* at 11am

Creating, Optimizing & Validating 3D Content



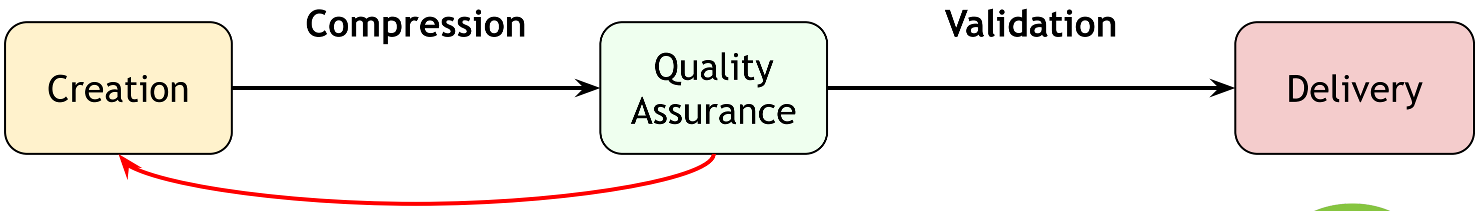
DO NOT MISS Eric's talk about *"Exploring the Artistic Frontier: Unleashing Creativity in 3D Models with glTF and PBR"* at 11am

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

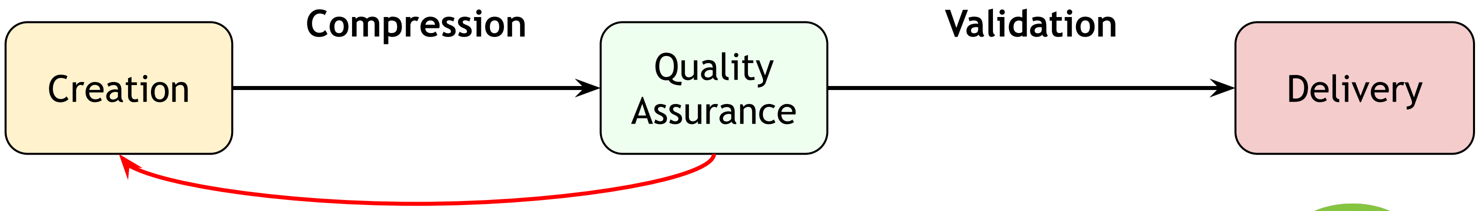
Creating, Optimizing & Validating 3D Content



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

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?

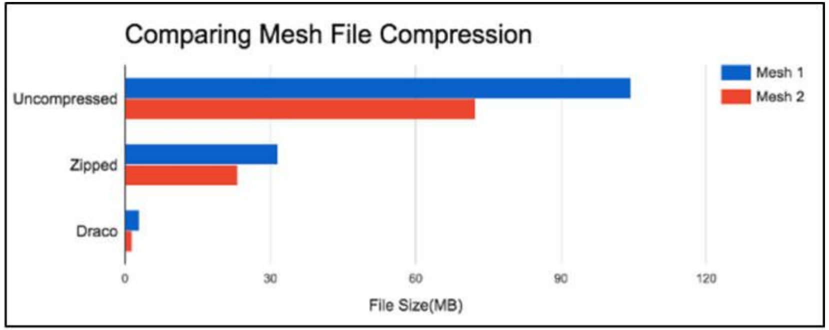
- **Can we do better?**
  - **Interactive online tools** to help artists!

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
3D DATA COMPRESSION

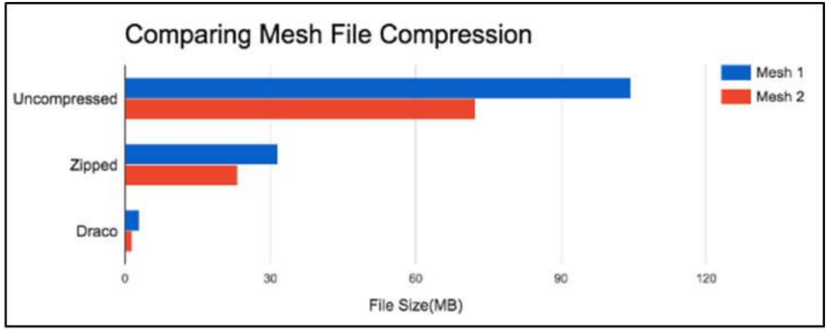


Typical Draco compression ratios

Enabling Efficient Geometry Size Reduction

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

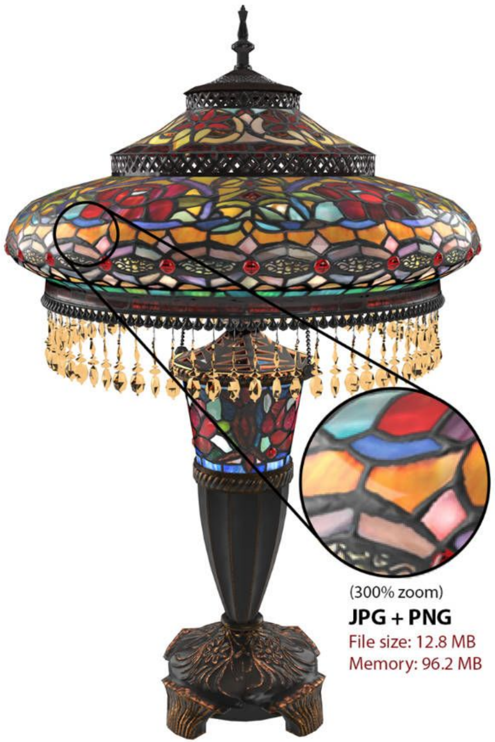
DRACO
3D DATA COMPRESSION



Typical Draco compression ratios

- **MESHOPT** provides compression and fast decoding for geometry, morph targets, and animations

Enabling Compact and Efficient glTF Textures

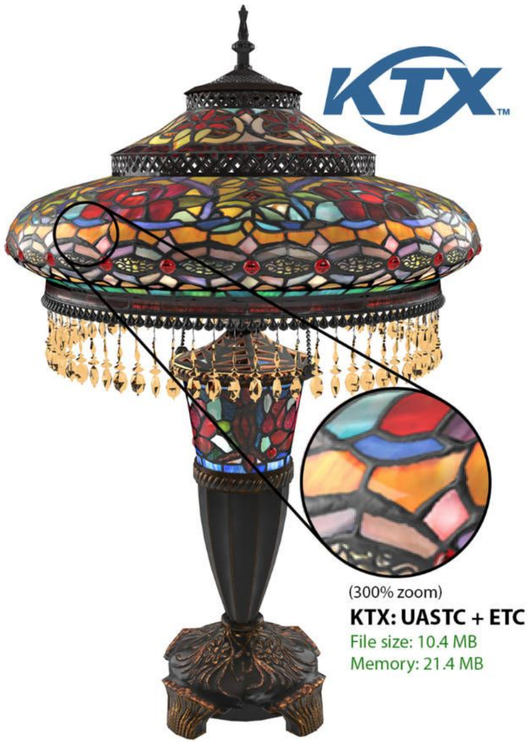


(300% zoom)

JPG + PNG

File size: 12.8 MB

Memory: 96.2 MB

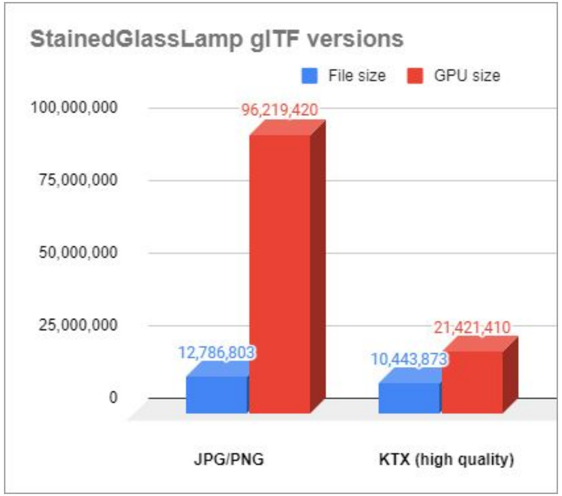


(300% zoom)

KTX: UASTC + ETC1S

File size: 10.4 MB

Memory: 21.4 MB

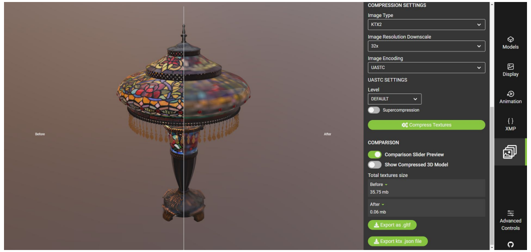


KTX Artist Guide

Khronos glTF-Compressor is here!

NEW

- **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



COMPRESSION SETTINGS

Image Type: KTX2

Image Resolution Downscale: 32x

Image Encoding: UASTC

UASTC SETTINGS

Level: DEFAULT

○ Supercompression

COMPARISON

● Comparison Slider Preview

○ Show Compressed 3D Model

Total textures size

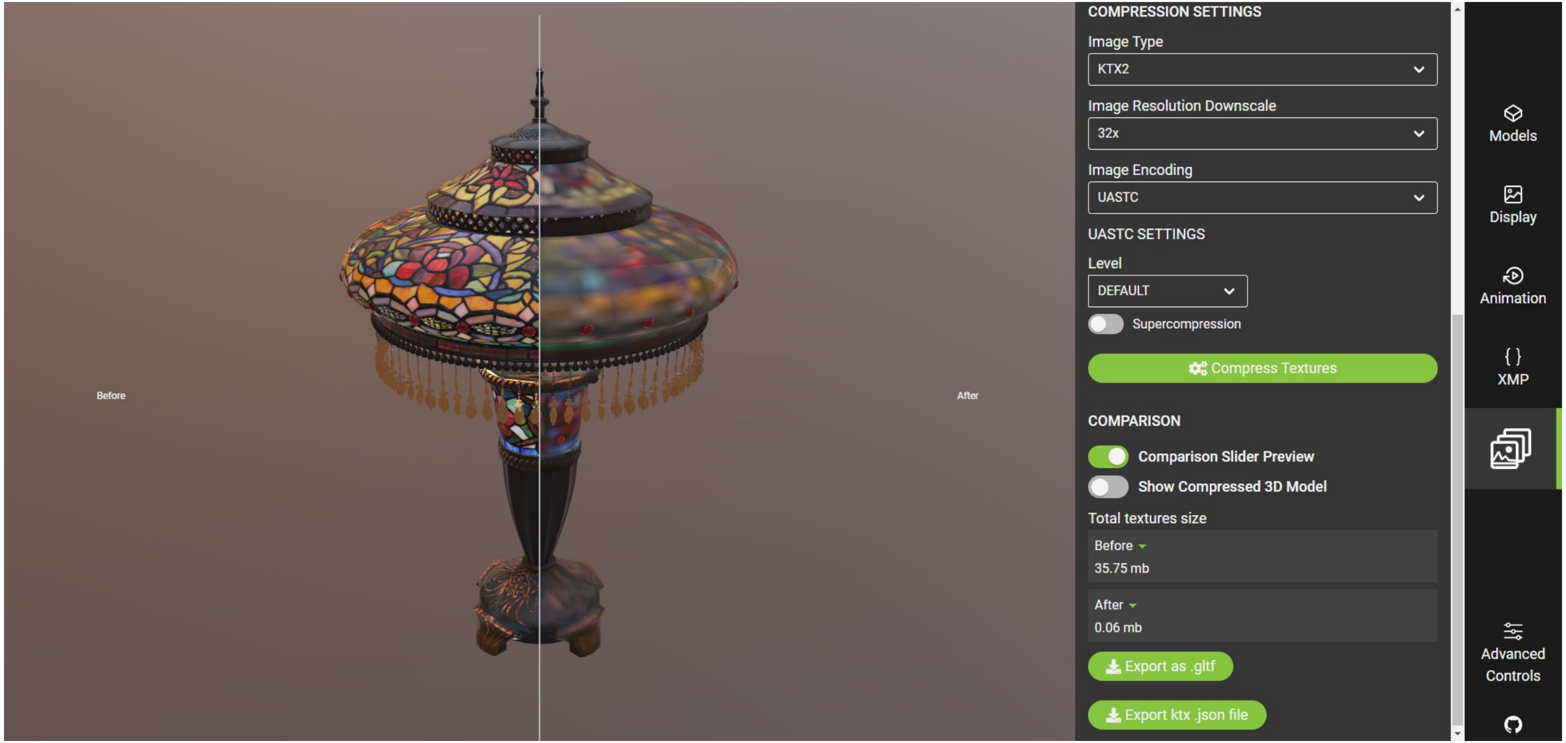
Before: 35.75 mb

After: 0.06 mb

- **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



COMPRESSION SETTINGS

Image Type: KTX2

Image Resolution Downscale: 32x

Image Encoding: UASTC

UASTC SETTINGS

Level: DEFAULT

○ Supercompression

Compress Textures

COMPARISON

● Comparison Slider Preview

○ Show Compressed 3D Model

Total textures size

Before: 35.75 mb

After: 0.06 mb

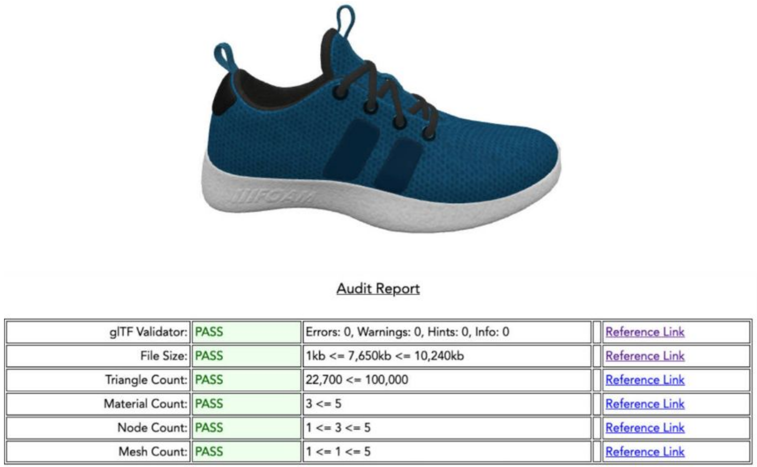
Export as .gltf

Export ktx .json file

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

Khronos glTF Asset Auditor

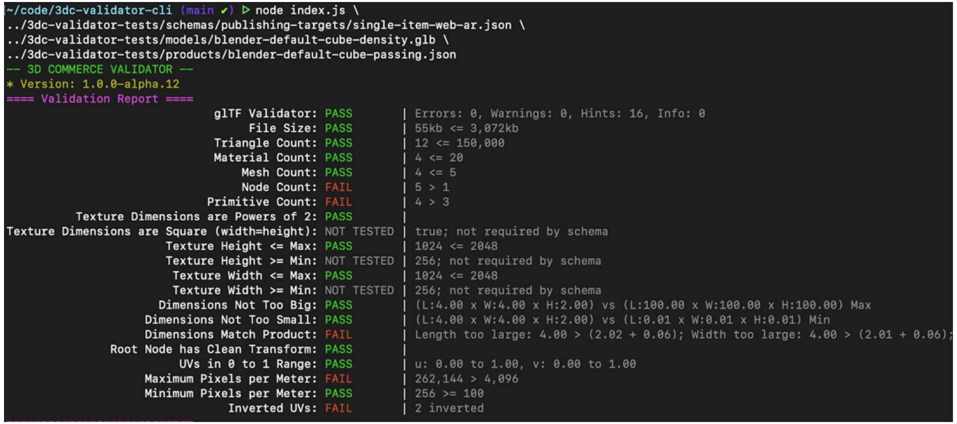
- Quickly check glTF asset for a specific use case, defined by an audit profile



Audit Report

glTF Validator:	PASS	Errors: 0, Warnings: 0, Hints: 0, Info: 0		Reference Link
File Size:	PASS	1kb <= 7,650kb <= 10,240kb		Reference Link
Triangle Count:	PASS	22,700 <= 100,000		Reference Link
Material Count:	PASS	3 <= 5		Reference Link
Node Count:	PASS	1 <= 3 <= 5		Reference Link
Mesh Count:	PASS	1 <= 1 <= 5		Reference Link

Web Browser



```
~/code/3dc-validator-cli (main ✔) ▷ node index.js \
../3dc-validator-tests/schemas/publishing-targets/single-item-web-ar.json \
../3dc-validator-tests/models/blender-default-cube-density.glb \
../3dc-validator-tests/products/blender-default-cube-passing.json
-- 3D COMMERCE VALIDATOR --
* Version: 1.0.0-alpha.12
==== Validation Report ====
                              glTF Validator: PASS       | Errors: 0, Warnings: 0, Hints: 16, Info: 0
                                   File Size: PASS       | 55kb <= 3,072kb
                              Triangle Count: PASS       | 12 <= 150,000
                              Material Count: PASS       | 4 <= 20
                                  Mesh Count: PASS       | 4 <= 5
                                  Node Count: FAIL       | 5 > 1
                             Primitive Count: FAIL       | 4 > 3
          Texture Dimensions are Powers of 2: PASS       |
Texture Dimensions are Square (width=height): NOT TESTED | true; not required by schema
                       Texture Height <= Max: PASS       | 1024 <= 2048
                       Texture Height >= Min: NOT TESTED | 256; not required by schema
                        Texture Width <= Max: PASS       | 1024 <= 2048
                        Texture Width >= Min: NOT TESTED | 256; not required by schema
                      Dimensions Not Too Big: PASS       | (L:4.00 x W:4.00 x H:2.00) vs (L:100.00 x W:100.00 x H:100.00) Max
                    Dimensions Not Too Small: PASS       | (L:4.00 x W:4.00 x H:2.00) vs (L:0.01 x W:0.01 x H:0.01) Min
                    Dimensions Match Product: FAIL       | Length too large: 4.00 > (2.02 + 0.06); Width too large: 4.00 > (2.01 + 0.06);
               Root Node has Clean Transform: PASS       |
                         UVs in 0 to 1 Range: PASS       | u: 0.00 to 1.00, v: 0.00 to 1.00
                    Maximum Pixels per Meter: FAIL       | 262,144 > 4,096
                    Minimum Pixels per Meter: PASS       | 256 >= 100
                                Inverted UVs: FAIL       | 2 inverted
```

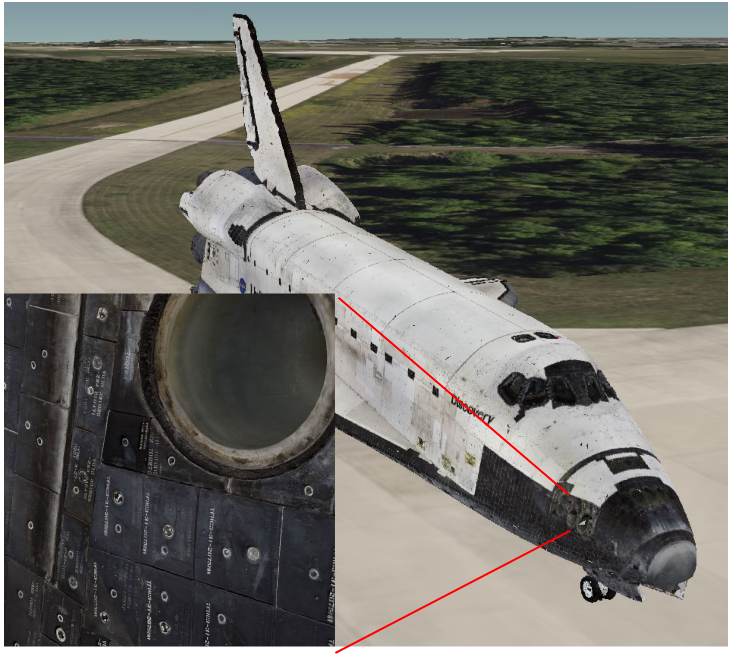
Command Line Interface

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

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

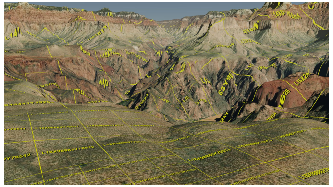


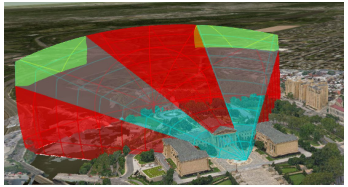
How do we get from a single model, to handling highly detailed scenes?

3DTiles™

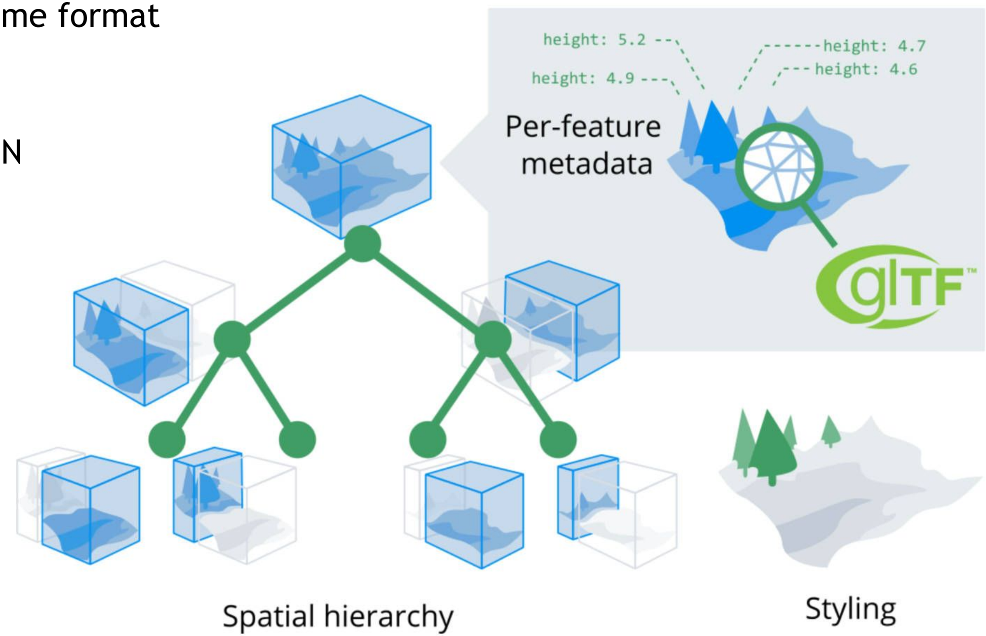
as a case study for glTF Optimization.

3DTiles Primer



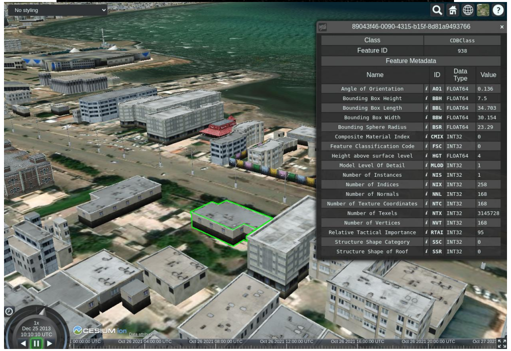


- **Open standard for streaming massive heterogeneous 3D geospatial data**
  - Terrain & imagery, 3D buildings, photogrammetry, point clouds, BIM models, interiors, etc.
  - Multiple source data types, one runtime format
- **Visualization + analysis**
- **Combine:**
  - Flexible spatial data structure 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**



3DTiles can be used for…

geospatial…



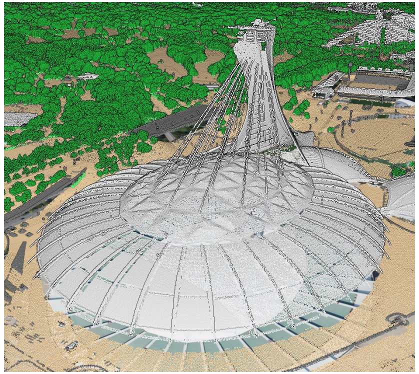
89043f46-0090-4315-b15f-8d81a9493766

Class	CDBClass
Feature ID	938

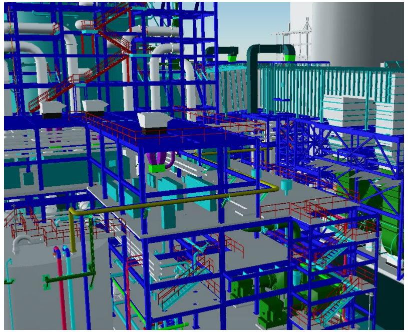
Feature Metadata

Name		ID	Data Type	Value
Angle of Orientation	i	AO1	FLOAT64	0.136
Bounding Box Height	i	BBH	FLOAT64	7.5
Bounding Box Length	i	BBL	FLOAT64	34.703
Bounding Box Width	i	BBW	FLOAT64	30.154
Bounding Sphere Radius	i	BSR	FLOAT64	23.29
Composite Material Index	i	CMIX	INT32	0
Feature Classification Code	i	FSC	INT32	0
Height above surface level	i	HGT	FLOAT64	4
Model Level Of Detail	i	MLOD	INT32	1
Number of Instances	i	NIS	INT32	1
Number of Indices	i	NIX	INT32	258
Number of Normals	i	NNL	INT32	168
Number of Texture Coordinates	i	NTC	INT32	168
Number of Texels	i	NTX	INT32	3145728
Number of Vertices	i	NVT	INT32	168
Relative Tactical Importance	i	RTAI	INT32	95
Structure Shape Category	i	SSC	INT32	0
Structure Shape of Roof	i	SSR	INT32	0

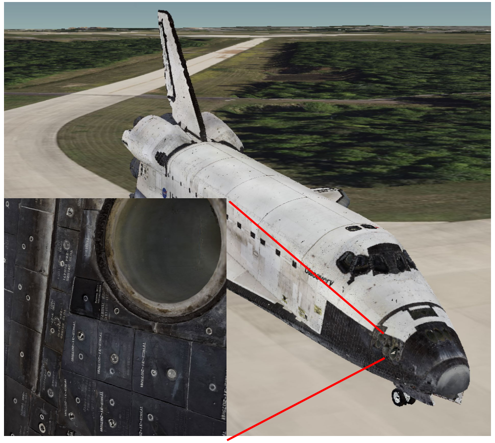
point clouds…



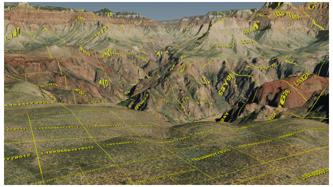
CAD/BIM…

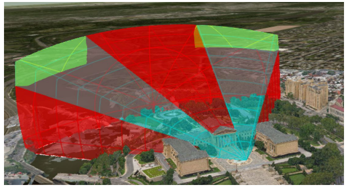


highly detailed models.

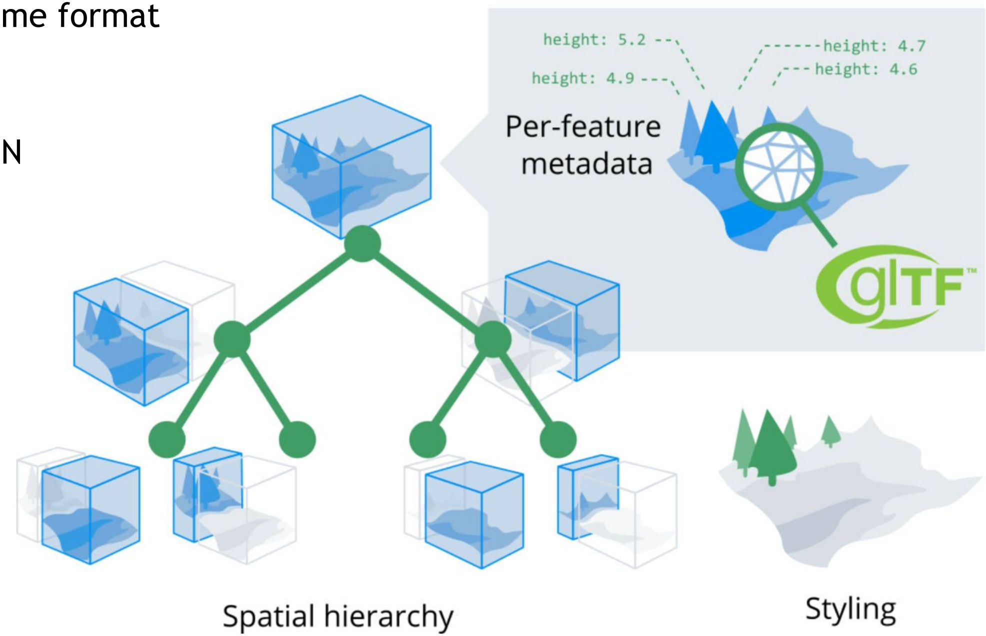


3DTiles Primer



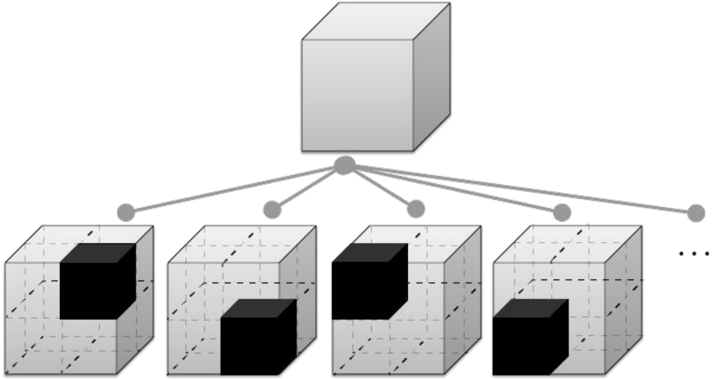


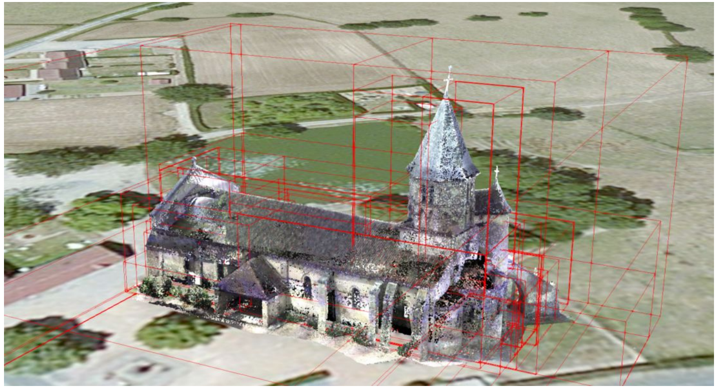
- Open standard for streaming massive heterogeneous 3D geospatial data
  - Terrain & imagery, 3D buildings, photogrammetry, point clouds, BIM models, interiors, etc.
  - Multiple source data types, one runtime format
- Visualization + analysis
- Combine:
  - Flexible spatial data structure in JSON
  - “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

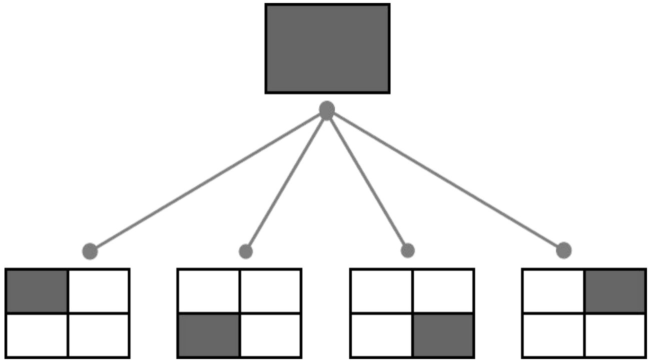
- Traditional method for subdivision in 3D graphics.
- Partitions space by recursive subdivision.
- Properties
  - Three-dimensional
  - Extends a quadtree by using three orthogonal planes to subdivide a space into 8 children.
- Supports (where needed!)
  - Non-uniform subdivision
  - Tight bounding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



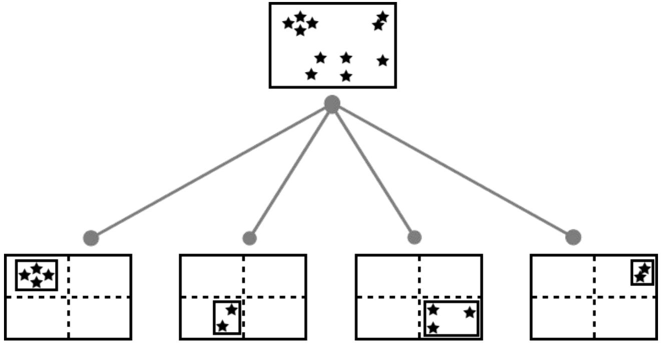


Quadtrees

- 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.

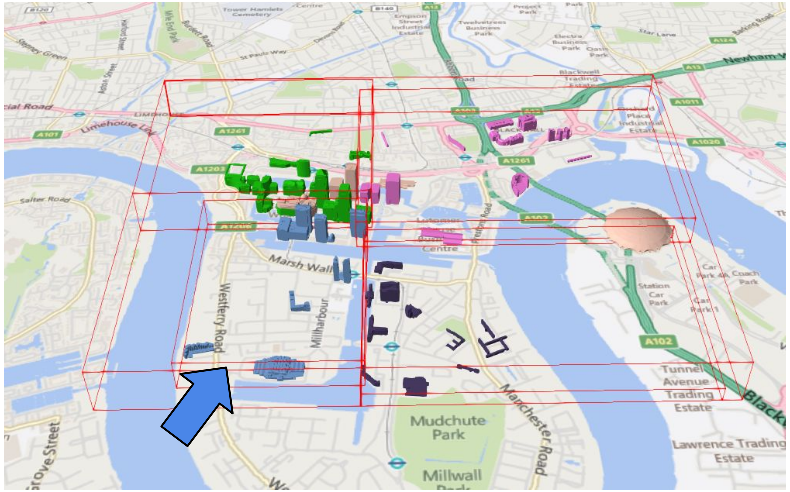


Classic quadtree subdivision

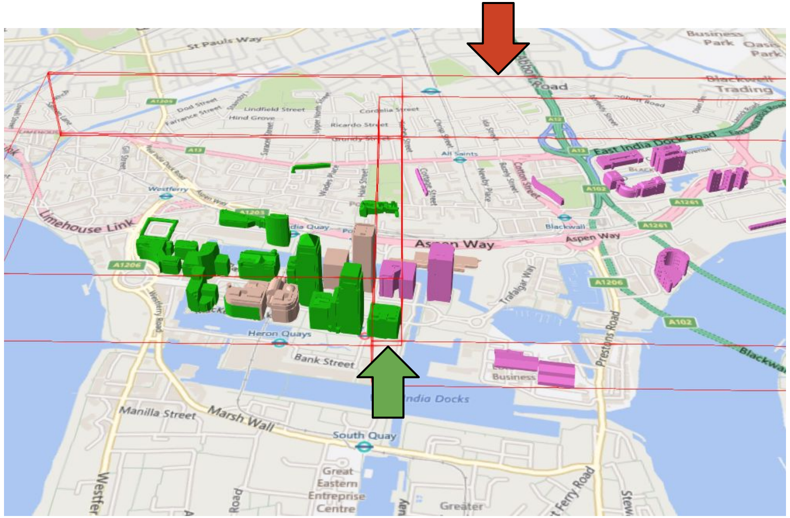


Quadtree with Bounding Volumes

Quadtrees



Quadtree with tight bounding volumes

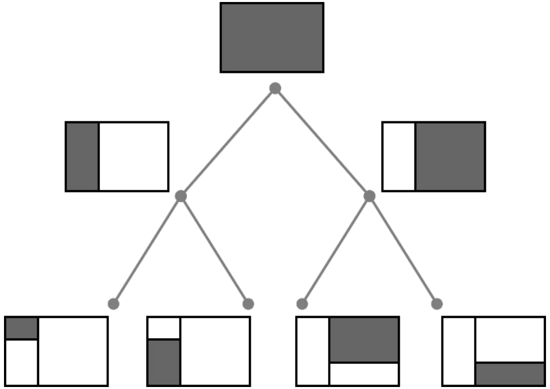


Quadtree with overlapping partitions

Image Sources: https://github.com/CesiumGS/3d-tiles/blob/draft-1.1/specification/README.adoc#spatial-data-structures

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



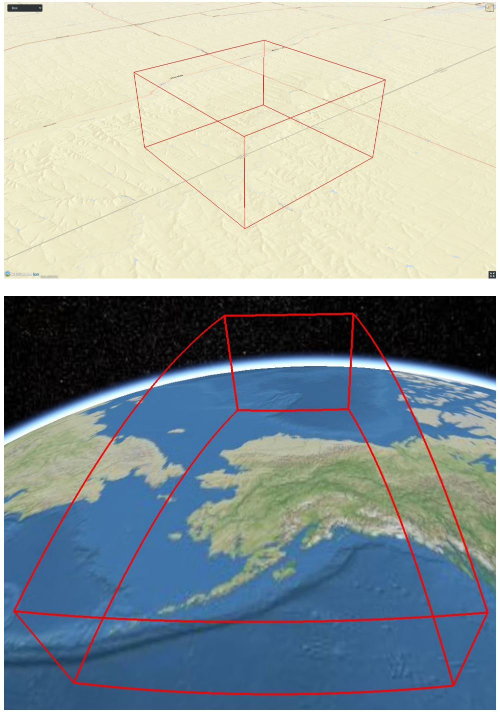
k-d tree. Note the non-uniform subdivision

You can use a combination of methods!

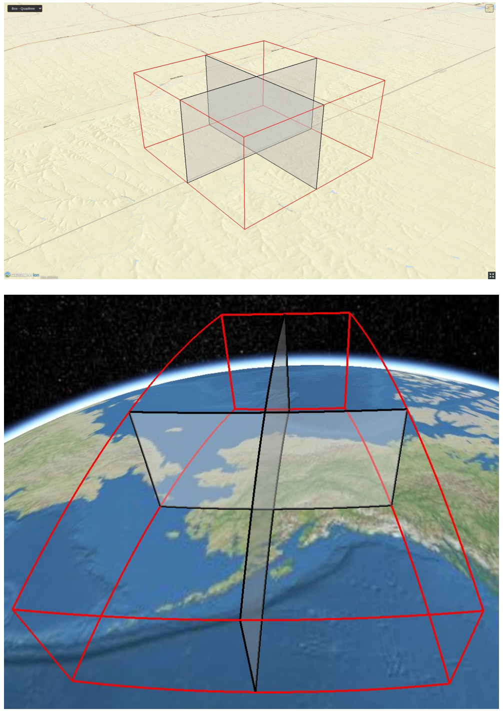
Combining methods

Bounding Volume Subdivision

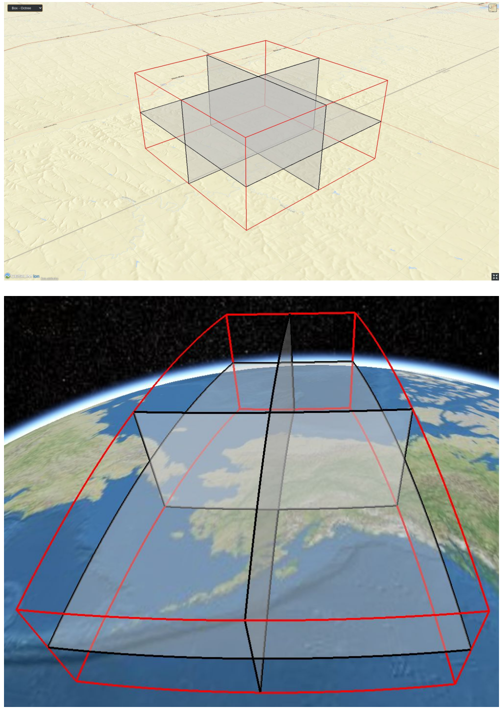
Root



Quadtree



Octree

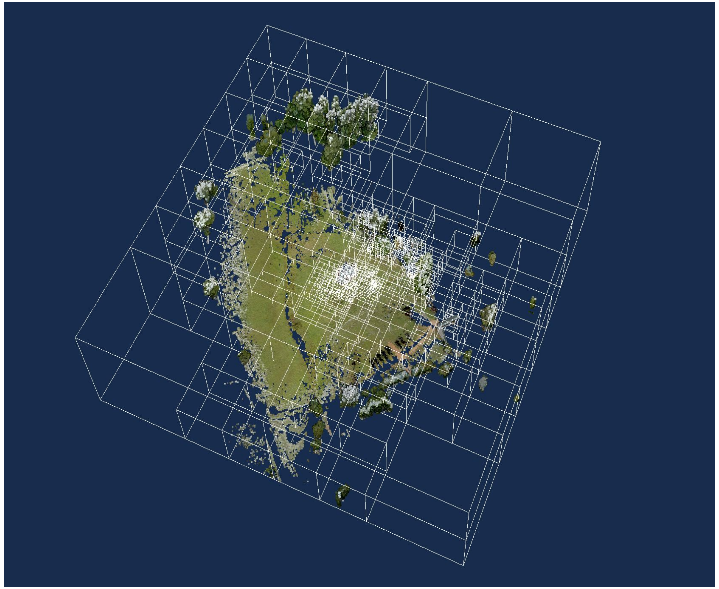


Combining methods

Choose the right subdivision

Octree in local coordinates

3D Tiles from terrestrial Lidar scan



Data source: Trimble

Quadtree in global coordinates

NYC buildings

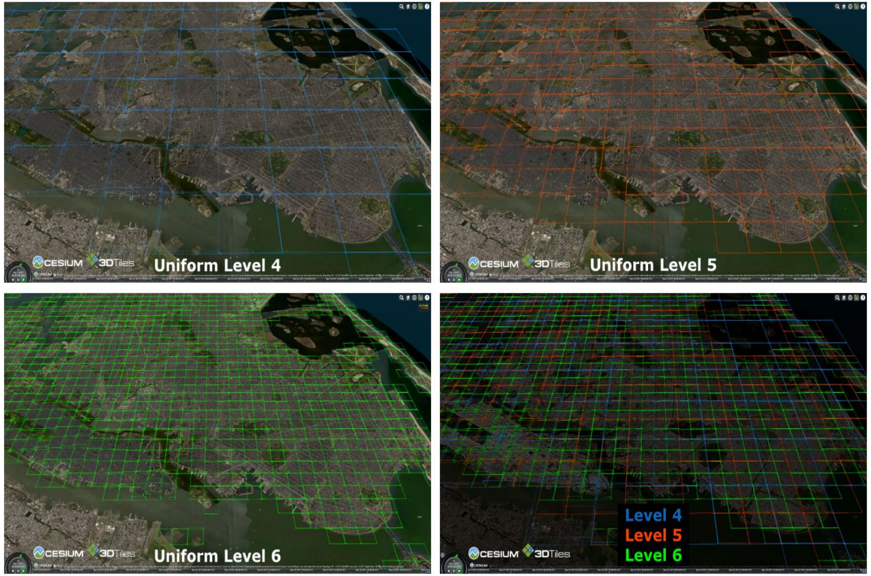


Figure 13. Bounding boxes for New York using Uniform Grid Tiling. Top Left: Uniform Level 4. Top Right: Uniform Level 5. Bottom Left : Uniform Level 6. Bottom Right : Composited view of Uniform Levels 4, 5, 6 for comparison.

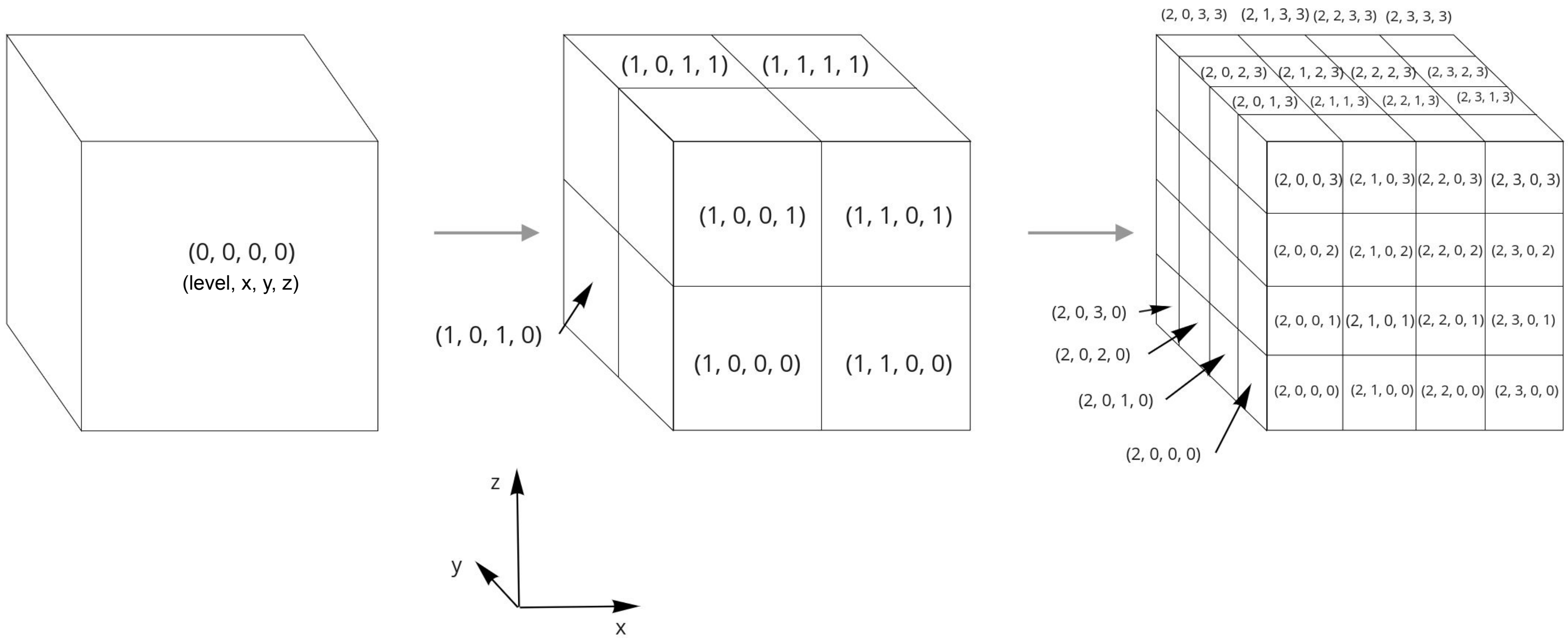
Combining methods

Choose the right subdivision

Discovery
United States
CESIUM

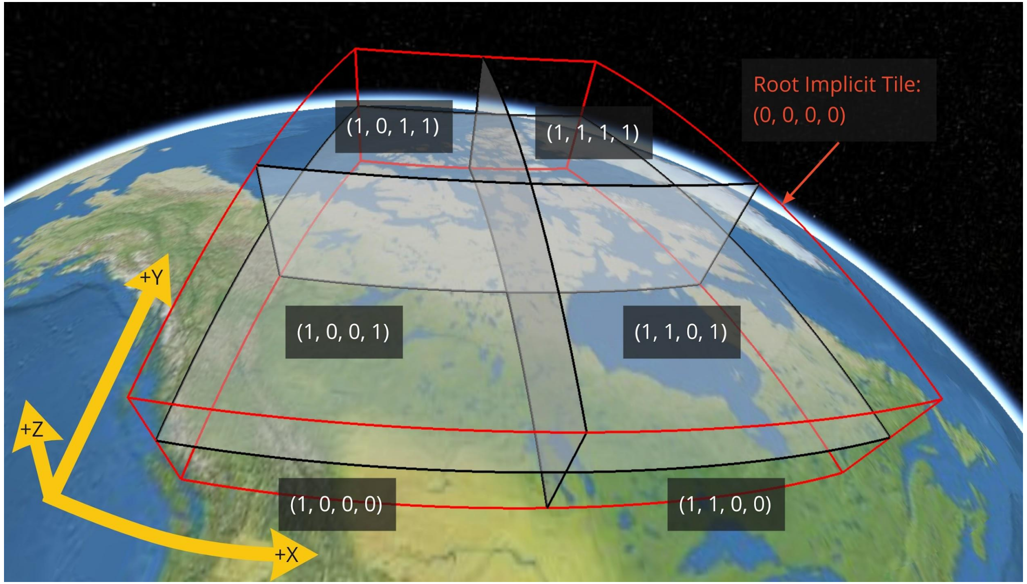
Combining methods

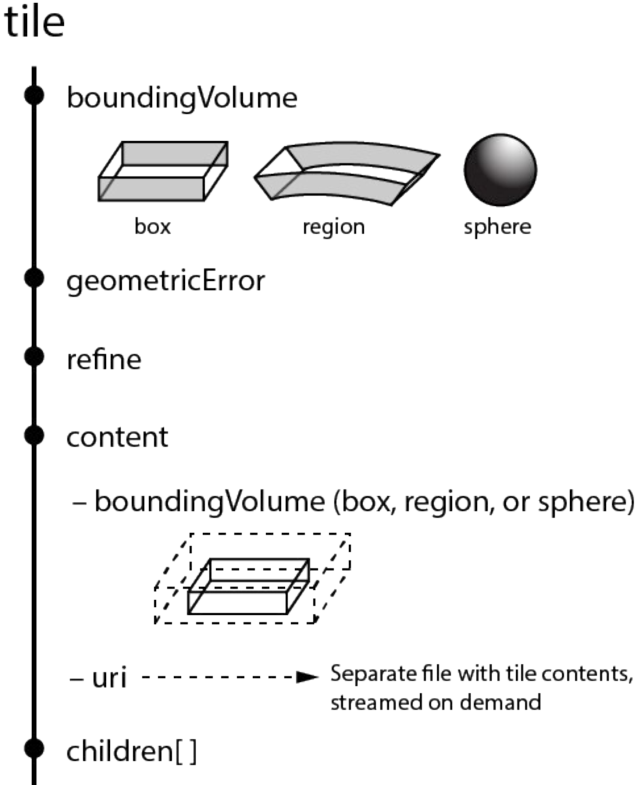
Handling coordinates: Box Bounding Volumes



Combining methods

Handling coordinates: Region Bounding Volumes





tile

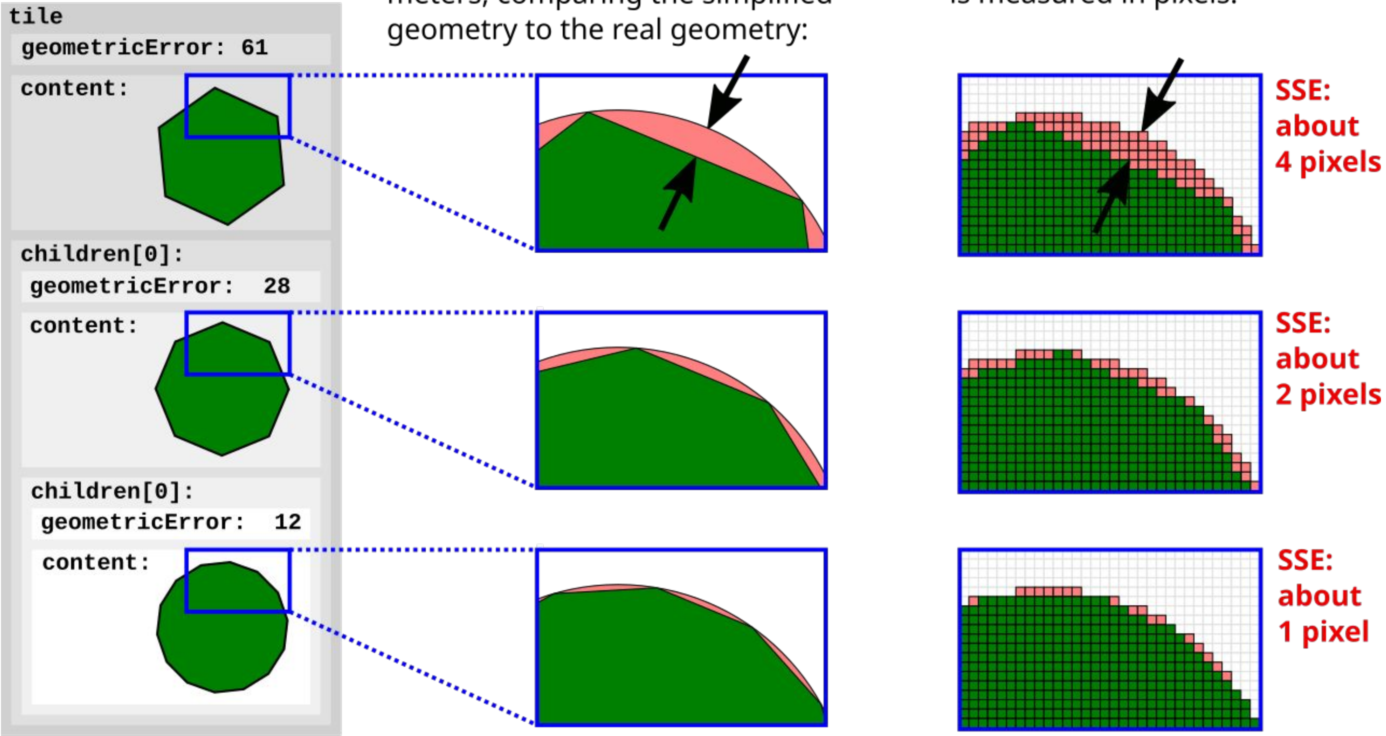
- boundingVolume
  - box
  - region
  - sphere
- geometricError
- refine
- content
  - – boundingVolume (box, region, or sphere)
  - – uri - - - - - - - - - - ► Separate file with tile contents, streamed on demand
- children[ ]

Combining methods

Geometric Error

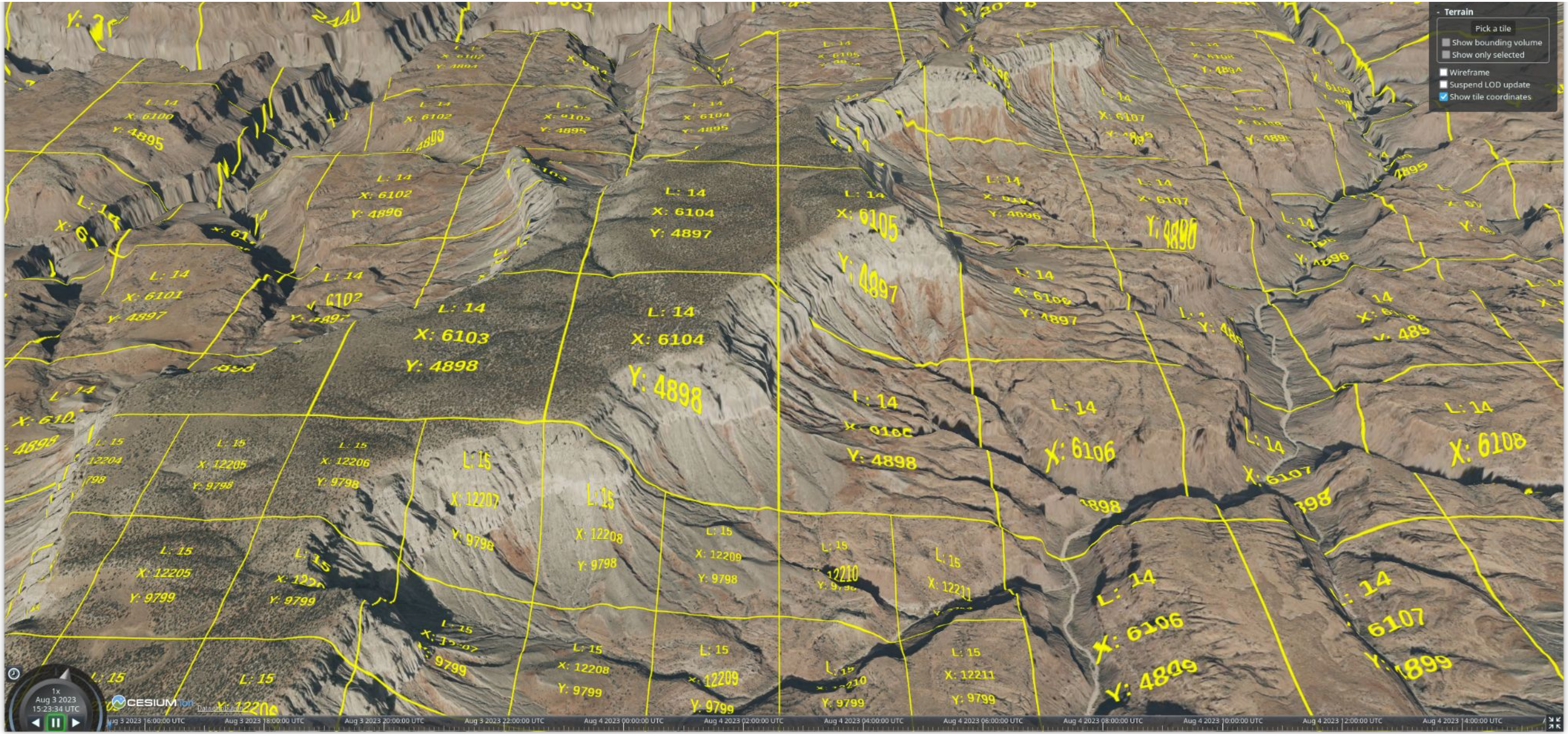
The geometric error is measured in meters, comparing the simplified geometry to the real geometry:

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



Combining methods

Mixing Levels of Detail



- Terrain

Pick a tile

□ Show bounding volume

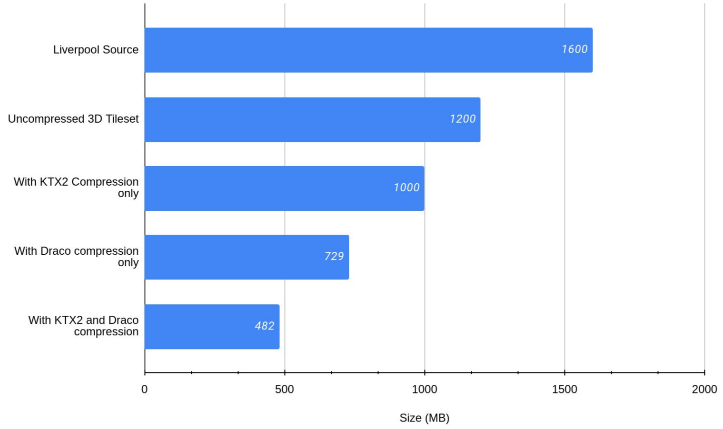
□ Show only selected

□ Wireframe

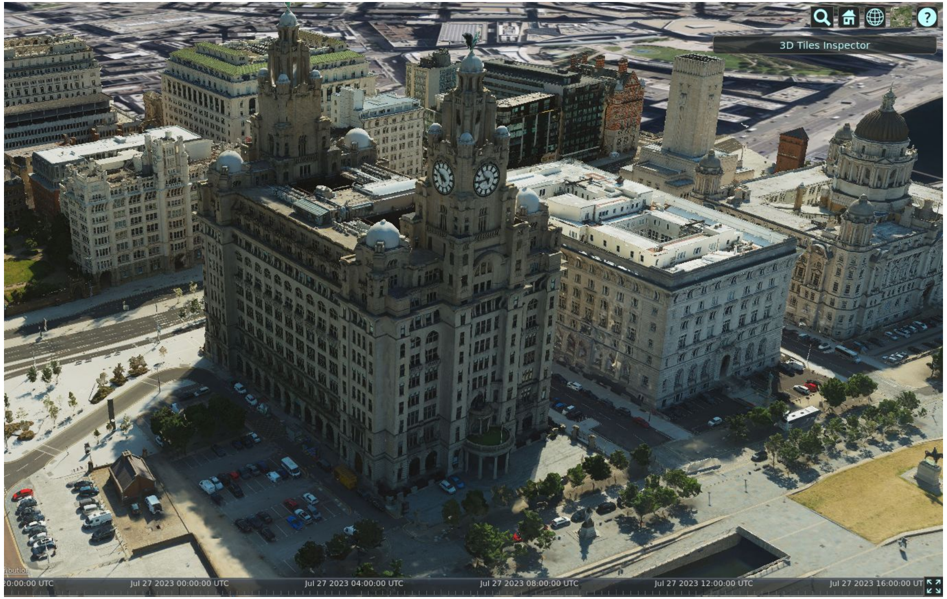
□ Suspend LOD update

☑ Show tile coordinates

Putting it all together



Liverpool Source	1.6GB
Uncompressed 3D Tileset	1.2GB
With KTX2 Compression only	1.0GB
With Draco compression only	729MB
With KTX2 and Draco compression	482MB



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
- 3D Tiles
  - You can use this for more than geospatial!
  - https://github.com/CesiumGS/3d-tiles
  - Community Projects
    - Vulkan Scene Graph loader for 3D Tiles
      - https://github.com/timoore/vsgCs
    - NASA-AMMOS 3D Tiles loader for three.js
      - https://github.com/NASA-AMMOS/3DTilesRendererJS
  - Reference Cards
    - https://github.com/CesiumGS/3d-tiles/tree/main/reference-cards
- Cesium
  - 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
  - Cesium for Unreal: https://github.com/CesiumGS/cesium-unreal
  - Cesium for Unity: https://github.com/CesiumGS/cesium-unity
  - Cesium for Omniverse: https://github.com/CesiumGS/cesium-omniverse

Ask the speakers

Max Limper
(DGG)

Pawel Nikiel
(DGG)

Adam Morris
(Cesium)

Andreas Vasilakis
(Phasmatic)