AE500020

Leveraging Open Standards & Guidelines for Mutliplatform Asset Sharing

Learning Objectives

- **The need open standards & guidelines**
- **glTF open standard:** reduces the pain of producing multiple configurations of the same graphic for deployment across various user platforms
- **3D Asset Guidelines:** to ensure content creators can create efficiently; tool-agnostic and can be used alongside Autodesk software
- **KTX texture compression:** for efficient texture files; compression options and how/when to apply them
- **Metadata schema:** to ensure consistency about information embedded in 3D assets
- **Rendering and viewing:** recommended practices and presets for creating, sharing, viewing, and collaborating on various platforms including desktop, mobile, and AR

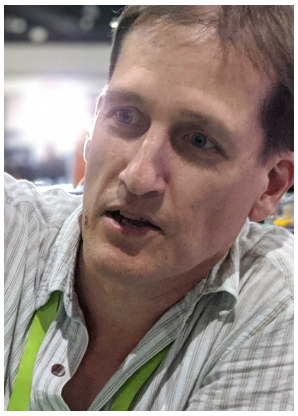
Description

The Khronos Group, the open consortium spearheading industry alignment on the creation, management and display of 3D content, understands the pain of producing multiple configurations of the same graphic for deployment across various user platforms.

Khronos 3D Commerce Working Group has developed a set of realtime 3D asset guidelines, tools and programs to ensure that artists and content creators efficiently author 3D assets and models that can be reliably deployed for real-time rendering and designers can view and share 3D assets consistently on diverse end-user delivery platforms. The guidelines are 3D tool-agnostic and can be used alongside Autodesk's software to bring product designers and engineers an easy-to-understand set of recommended practices and presets for creating, sharing, and viewing 3D assets and collaborating on various platforms including desktop, mobile, and AR.

This talk will explore the ongoing work within Khronos including the 3D Assets Guidelines, descriptive Metadata schema to ensure consistency about information embedded into 3D assets, an upcoming certification program for model viewers that ensure consistent viewing experience on every platform, and more. Presenters will point out decision points and explain how designers, engineers and architects can apply these best practices. Chairs of the Khronos 3D Commerce Working Group and 3D Formats Working Group will explore how these guidelines have been applied outside of a retail context by entities including the Smithsonian Institute. The presenters will discuss how guidance for product design diverges from the 3D asset use case for e-commerce, art or education.

Speakers



Eric Chadwick
Staff Technical Artist, Wayfair

- Pipeline technical director, real-time 3D & rendering workflows
- Regular contributor in Khronos 3D Commerce & 3D Formats working groups
- 20 years in game development, 4 year AU veteran

Nathaniel Hunter
Chief Executive Officer, Dreamview, Inc

- Co-Chair of 3D Commerce Asset Creation TSG
- 30 years in content creation with Autodesk software, spanning film, TV, commercials, games, products, and internet technology
- B2B online platform serving retailers, suppliers, manufacturers and brand owners
- Scalable and standardized management, creation and distribution of real-time and photo-real product content

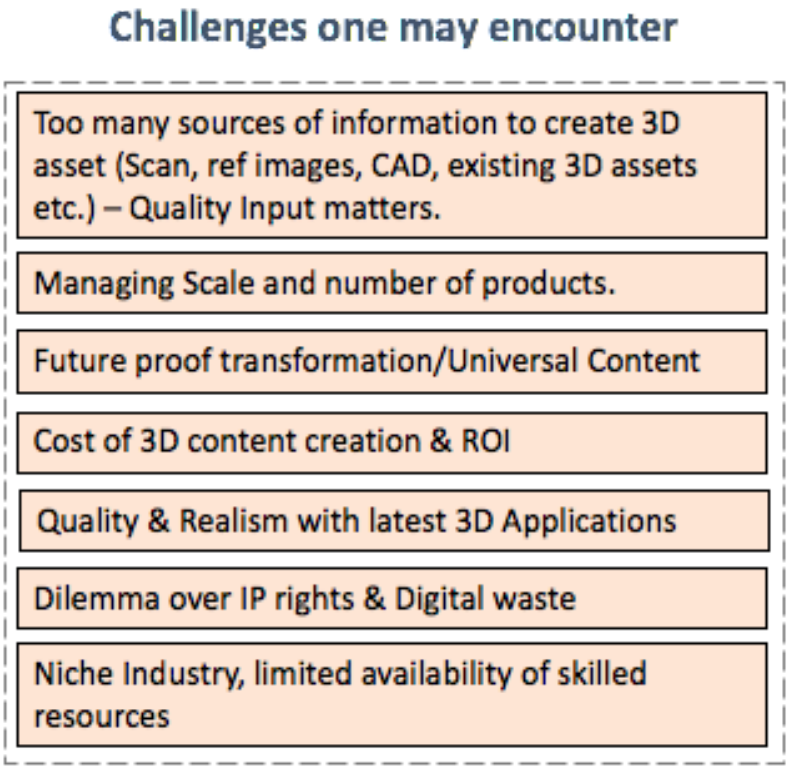
The Need For Open Standards & Guidelines

This talk focuses on how we use glTF to publish real-time 3D content for e-commerce purposes; basically selling products online. glTF is a great standard for web 3D, and we'll walk you through the best practices we've learned through our own professional work.

First and foremost, we need to ask the question, why are Open Standards and Guidelines even needed? These are just a few examples of challenges that are being faced daily.

Common challenges in 3D content creation

Without standards and guidelines, there are numerous issues.



These are just a few examples of challenges that are being faced daily.

Without Standards, Chaos ensues…

Quality and consistency is impossible to manage. A prime example of these issues that continually crop up is the varying asset quality delivered by any number of vendors around the world working with brands and retailers in the creation of this 3D content. WIthout standards or guidelines, the results from 3 different companies, with 3 different approaches, will likely result in 3 different deliverables.

Original Reference

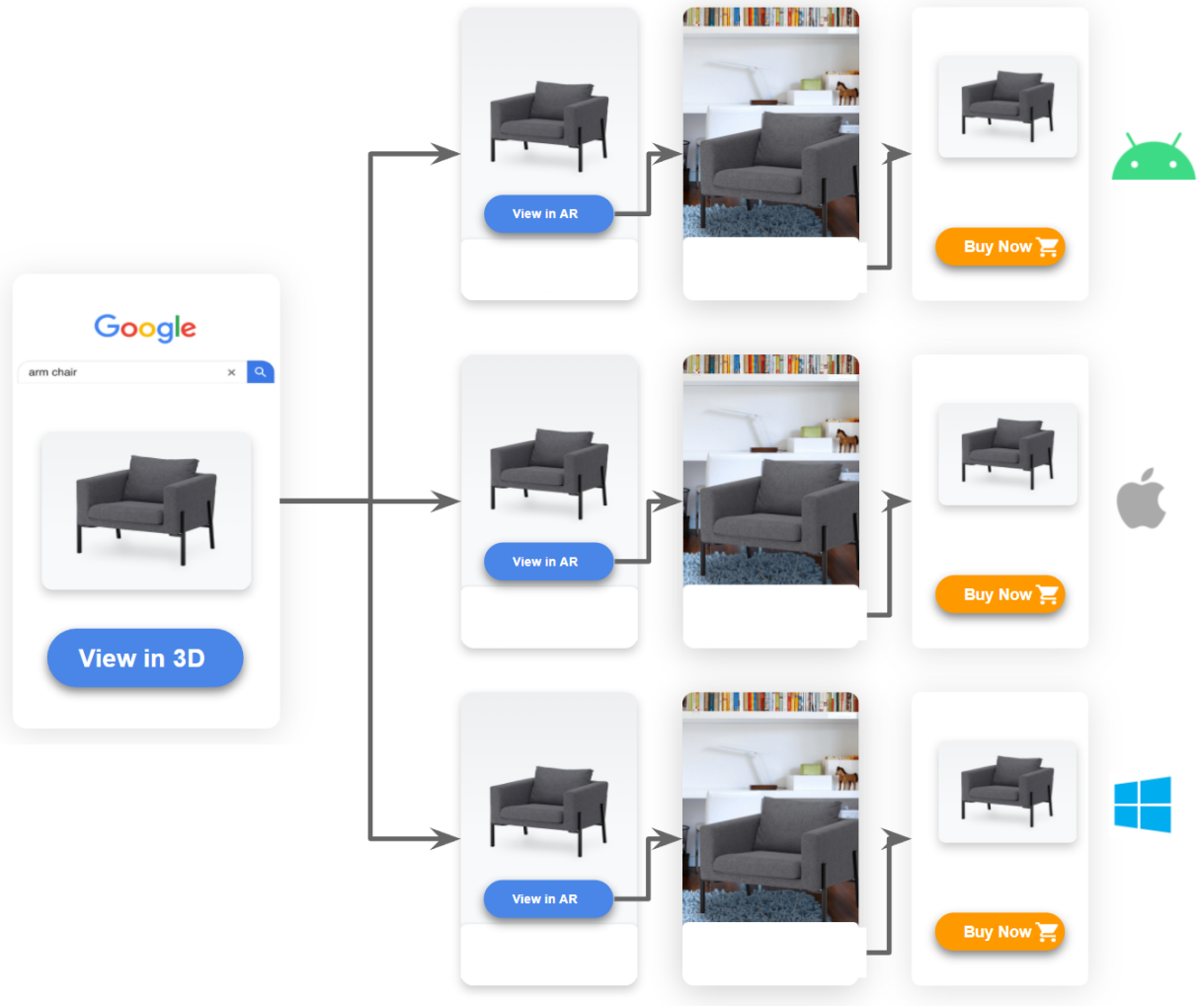
Vendor A

Vendor B

Vendor C

…. even across different platforms

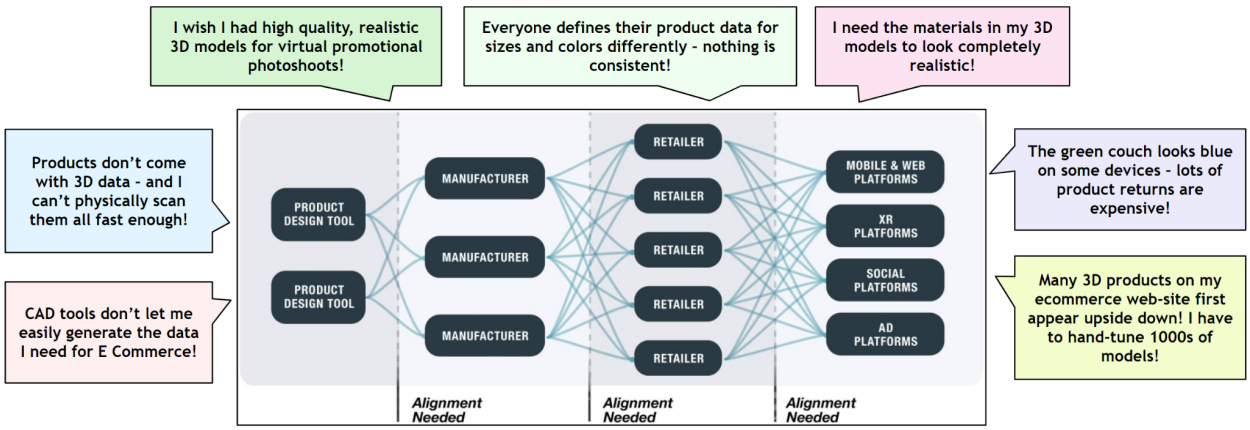
Inconsistent visual quality across experiences. This compounds even at the level of an asset that is finished and delivered across multiple marketplaces, websites and technology platforms. From android, to iOS, to the web, assets without standards at the viewer level can showcase less than desirable results.



Complexity of Requirements

There are multiple layers of concerns and friction points. Boiling these issues down however is not trivial, with a multitude of complex concerns and friction points across industries.

So whats the solution?



Complex retail pipeline with hundreds of companies and thousands of products
Many friction points - tooling, technical and commercial

An open standard solution

There are multiple layers of concerns and friction points. Enter 3D commerce, an initiative inside of Khronos.

3DCommerce™

Create specifications and guidelines to align the 3D asset workflow from product design through manufacturing, through each stage of retail to end-user delivery platforms

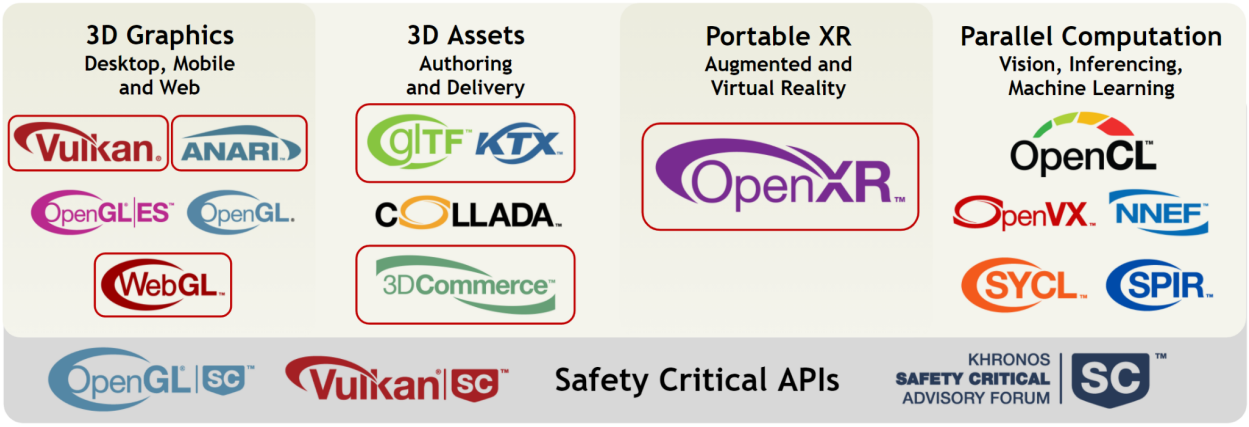
The Khronos 3D Commerce Mission

Boiled down to the most simplistic description. To render everything, everywhere. But how to go about this?

Khronos Active Initiatives

3D commerce is an initiative built into The Khronos Group, with a focus on creating

Royalty-free open standards for 3D graphics, Virtual and Augmented Reality, Parallel Computing, Machine Learning, and Vision Processing

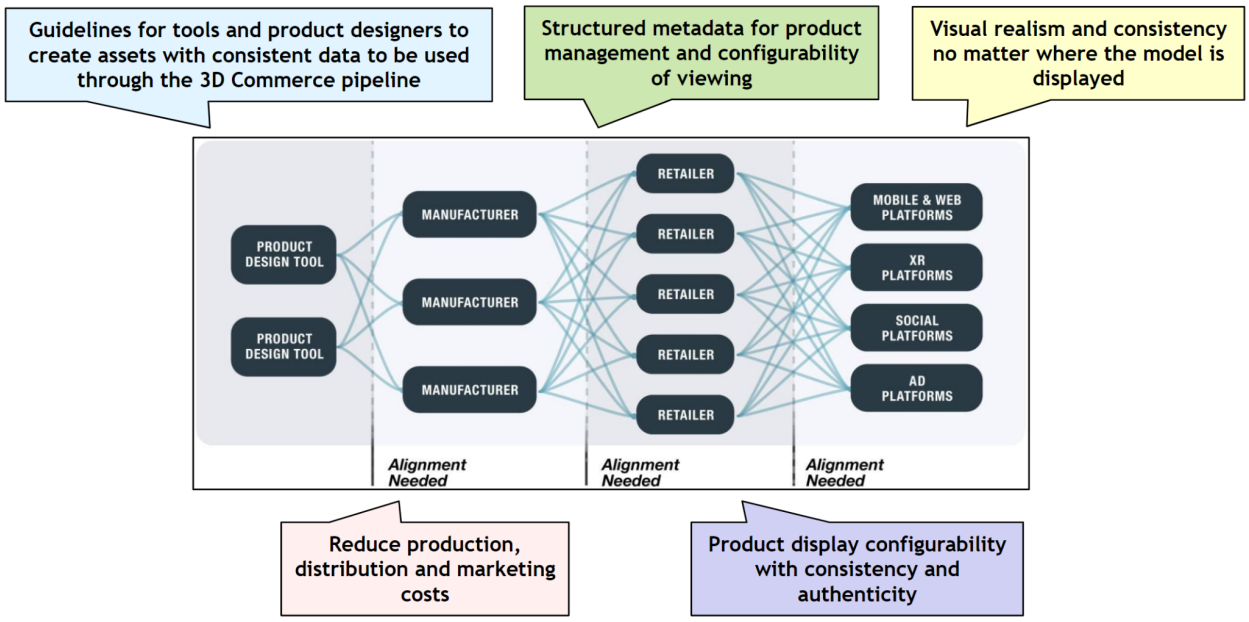


Who is part of Khronos?

Diverse Technology and E-Commerce companies working together. Khronos is supported and implemented by the leaders across technology, creativity, and business. This is just a small sampling of the member companies that not only strive to create, but also adhere to these global standards and guidelines.

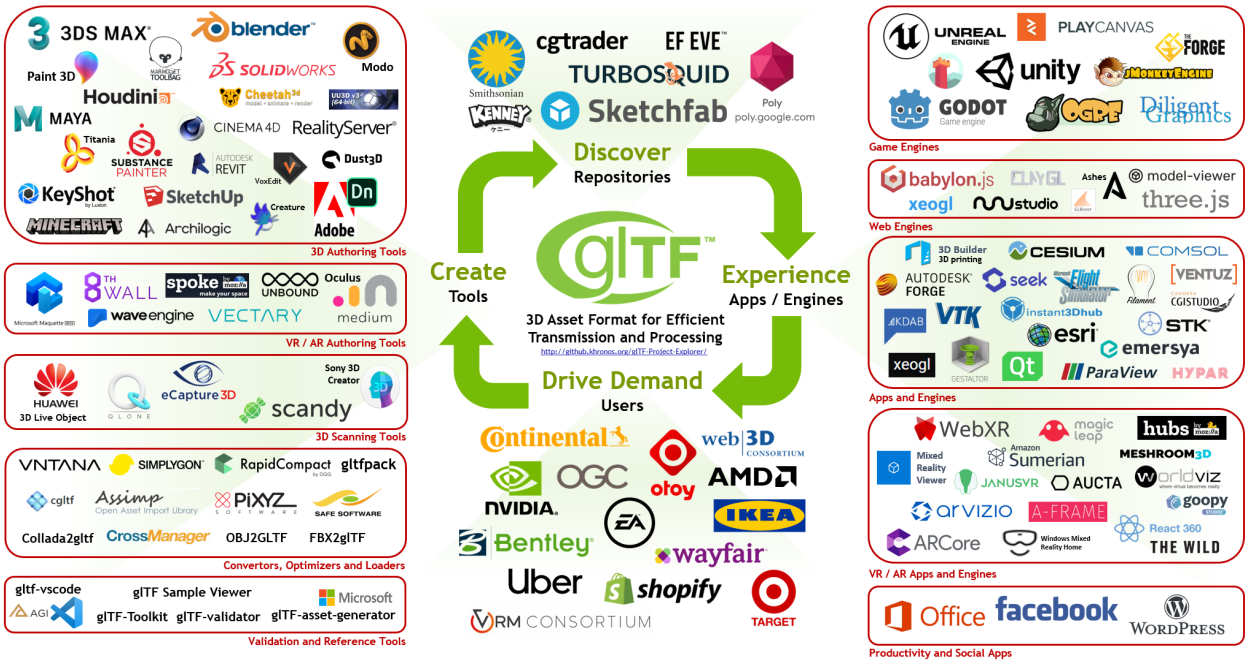
Khronos 3D Commerce Goals

Create Open Standards & Guidelines for Multiplatform Asset Sharing. The goals for 3D Commerce are the very topic of this conversation. And for this, I'm going to pass the baton along to Eric to go into more depth on how we're accomplishing this on a day to day basis.



glTF Open Standard

So let's do a quick overview of what glTF is all about. glTF stands for OpenGL Transmission Format, a royalty-free specification for efficient transmission and loading of 3D scenes and models.



This is a crazy slide, but it really shows how widely adopted this format is, across so many industries. glTF is managed by the Khronos Group, a consortium of companies coming together to develop and extend open source standards. glTF is purpose-built to be an efficient transmission format for 3D content, and as you can see it has many different use cases. There's a strong groundswell of support here.

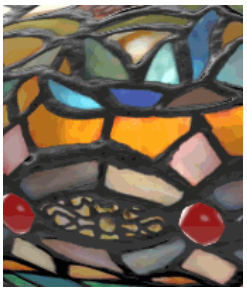
glTF for High Quality Assets

Material support for refraction, sheen, clearcoat, SSS, etc.
https://github.com/KhronosGroup/glTF-Sample-Models

glTF is able to handle high-fidelity content. In the 3D Formats working group, which oversees improvements to the glTF ecosystem, we've recently released several improvements to materials handling, for complex surfacing. Sheen for fabrics, clearcoat and colored specular handling, real-time refraction for colored and foggy glass, volumetric translucency with attenuation. glTF can handle a wide variety of visually rich materials.

Base Color
Metal
Rough
Emissive
Occlusion
Normal

KHR_materials_transmission
KHR_materials_sheen
KHR_materials_clearcoat



glTF Wide Rendering Support

Same asset works in both rasterizers (realtime web) and pathtracers (offline progressive).

The glTF standard is built with multi-platform rendering in mind. You can take the same asset and render it in both a real-time web engine, and an offline pathtracer, and you'll get very similar results. In these examples, the model is rendered in a realtime rasterizer on the left, and it's raytraced on the right. It's the same model.

If you look closely at the right, you should see more evidence of self-reflections and refractions. This takes many minutes to resolve in a progressive pathtraced render. Whereas the realtime render on the left is running in milliseconds, and it's taking various shortcuts to be performant.

The key thing here is, it's the same 3D asset. Meaning essentially, author once, render everywhere.

Babylon.js Sandbox

Dassault Systèmes Enterprise PBR Sample Renderer

glTF for Variants/Options

KHR_materials_variants and glTFVariantMeld. In the 3D Commerce working group, we've published an extension for options. glTF is great for showing multiple configurations of the same base product.

This is pretty easy to set up, just export a glTF model for each variant, then combine them together into a single asset, which then tosses out redundant data. You end up with an efficient file. You don't need five versions of the couch geometry; you only need to store and transmit the material changes.

3D Asset Guidelines

How do we get to these results that are so consistent?

Real-Time Asset Creation Guidelines

Best practices and modeling standards for high-quality, efficient 3D assets.

Asset Creation, the foundational layer to all product visualization, requires a consistent approach that can be reproduced over and over, to insure the compatibility of products no matter where they come from, or what workflows create them.

To this end, the Asset Creation Guidelines, which are freely available, map out a pathway for consistency across Geometry, materials, textures, rendering and lighting, as well as publishing targets and file formats that can be easily accessed by both the content creators as well as those that request the creation.

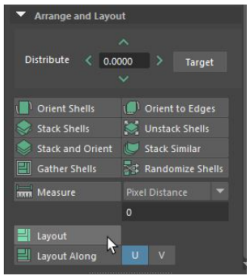
This information is available today at https://github.com/KhronosGroup/3DC-Asset-Creation

3DCommerce™

Real-time Asset Creation Guidelines

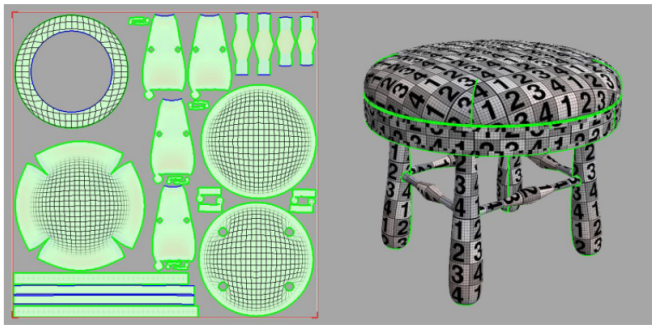
- Editors and Alumni
- Content
- Asset Creation Guidelines Summary
  - Executive summary
  - File Formats and Asset Structure Summary
  - Coordinate Systems and Scale Summary
  - Geometry Summary
  - UV Coordinates Summary
  - Materials Summary
  - Textures Summary
  - Rendering & Lighting Summary
  - Levels of Detail Summary
  - Publishing Targets Summary
  - glTF & USDZ Summary

Within Maya, using the **Layout **command in the Arrange and Layout rollout of the UV Editor will also help produce similar UV packing results.



(C)2020, TurboSquid. License: CC BY 4.0 International
Figure 4.27

By using these tools, the UVs will be organized, scaled and packed in such a way as to provide even texel density to all UV shells all within the UV space. In most cases, the tools will **not **reorient shells to align them, and it will be the responsibility of the artist to manually re-orient UV shells to their desired rotational angle before packing. The results of such a UV packing effort can be seen below.



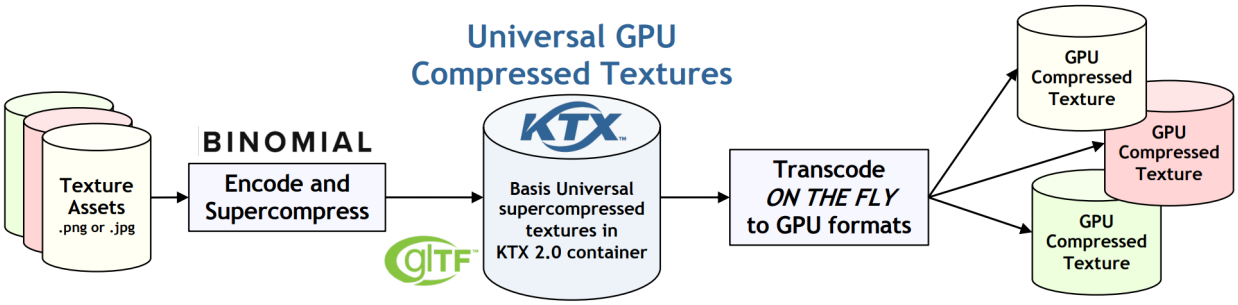
KTX2 Texture Compression

Let's talk about compression. How do we get efficient transmission of high-fidelity assets to our end customers?

KTX2 is Universal Texture Compression

KTX (Khronos Texture) is an efficient container format for supercompressed GPU textures

- https://www.khronos.org/KTX
- Start from PNG/JPG textures
- Encode into Basis Universal
- Transcoded on load into native GPU formats
- Textures stay compressed on the GPU



KTX2 provides a great way to compress textures; they're often the greatest contributor to large file sizes. We've had compressed texture formats for a long time; think JPG and PNG. But 3D models depend on textures being loaded into the GPU for rendering; which presents some unique challenges.

We have many different GPUs out there, and they each support different compressed texture formats. KTX2 solves this by compressing to a universal format, which is then transcoded on the fly into whatever format is best for a given GPU. So, the textures stay compressed in memory.

KTX2 Results

What does that look like? Here's a complex asset with detailed high-resolution textures. A lot of these are 2k. On the left in red we have the original asset with JPGs and PNGs, and on the right in green we have two versions compressed with KTX2. The key thing to notice here is not only the quality being preserved, but the differences in memory consumption on the GPU. It's really significant. Let's load a live demo to see the results in real time.

example glTFs | demo: higher quality | demo: smaller file

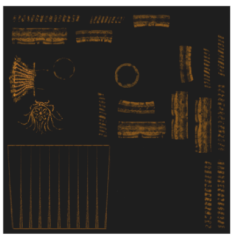
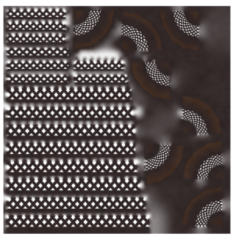
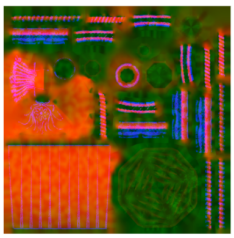
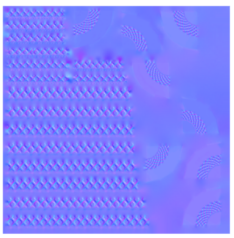
On the left is the model with JPG and PNG textures, and on the right with KTX2. At normal viewing distance, they're virtually indistinguishable.

At Wayfair we're coming at this from an e-commerce perspective; how do our customers normally see these models? At this distance, there's really no difference.

And even if we zoom in close, the differences are really hard to spot. We're getting really high-quality results, for about a fifth of the GPU memory cost. And the transmission size is also a bit smaller.

KTX2 Compression Types

- ETC1S for color textures with little to no loss of visual quality
- UASTC for ORM and normal maps which often need higher precision

Use Case	Baseline Khronos Compressed Texture Format	Format with selected modes	LZ-Based Supercompression	Supercompressed bits per pixel (*typical* rates for 24-bit textures)	Compressed Image Comparison (*typical* rates for 24-bit textures)
Highest Quality	ASTC (8bpp includes alpha)	UASTC	Optional RDO Encoding + zstd	6bpp	PNG 6bpp
Smallest Size	ETC1 (4bpp, 8bpp with alpha)	ETC1S	Custom BasisLZ	1.0bpp	JPG 1.5bpp

There are two main compression methods in KTX2. ETC1S and UASTC. ETC1S is the smaller, and it excels at textures with homogenous colors.

What does that mean? Think photographic content, or typical color maps. When the RGB color channels have similar values, ETC1S does a great job.

UASTC is a bit larger, but you need it for textures with lots of variation across the color channels. ORM for example; which packs Occlusion, Roughness, and Metalness into three RGB channels; with ORM the channels are usually quite different. This is a challenge for ETC1S, but UASTC does a great job, at the cost of a slightly larger size.

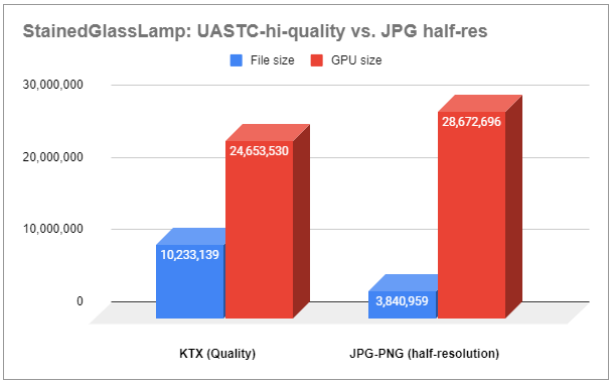
KTX2 Versus PNG/JPG

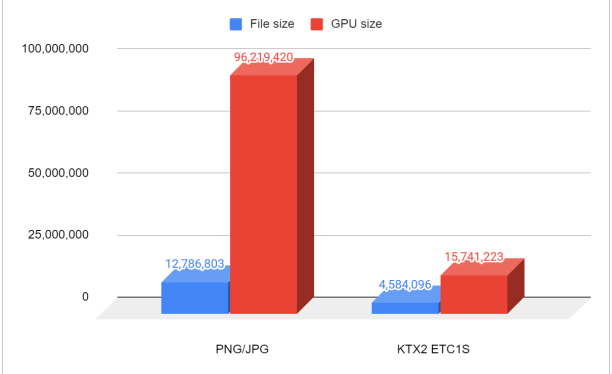
- PNG/JPG must be uncompressed for GPU
- Half-resolution PNG/JPG is still larger on GPU
- KTX2 not always smaller transmission size, but definitely smaller in memory

Look at the tall red bar in the chart on the left. PNG and JPG textures have to be uncompressed completely as they're non-native to the GPU. However KTX2 is transcoded directly into GPU-supported codecs, and they stay compressed.

On the right is a graph showing a common ask, why can't you just downsize all the textures instead of using KTX2? If you halve the resolution you're not really getting the same results as you get with KTX2. Sure the file size is smaller (the blue bars), but the GPU size is larger, and don't forget all your textures are a lot blurrier too, because you're using half-resolution for the JPGs and PNGs.

The KTX2 files are still at their original resolutions here. So usually it's worth it to supercompress, rather than downsample.





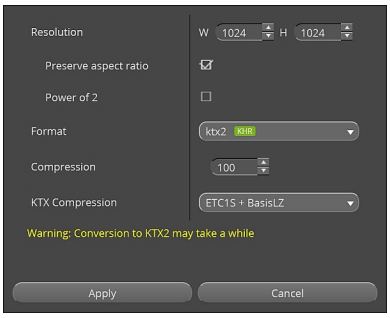
KTX2 Tools

We have a decent number of tools at our disposal for making KTX2 files.

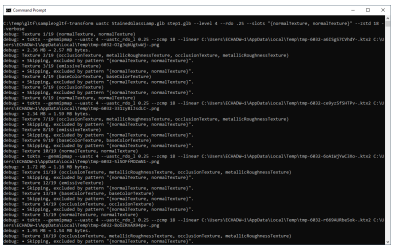
On the left we have some great graphical tools that really help demystify the process, and give you instant visual feedback.

On the right we have command-line tools, which give you the ultimate control. It's really up to you which work best in your situation. So that's a basic rundown of texture compression in glTF and how it helps.

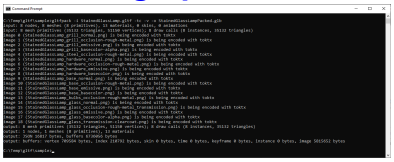
Gestaltor



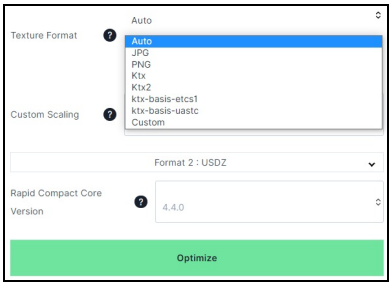
glTF-Transform



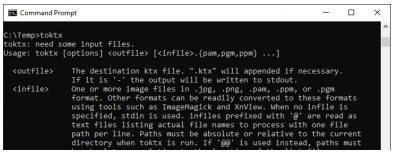
gltfpack



RapidCompact



toKTX



What tools do you want?

We want to hear from you!
khronosdevs.slack.com

Metadata Schema

Beyond the creation guidelines that define how to create the 3D models and vi, the data that transfers with these models becomes very important, so defining a Metadata Schema was a core requirement.

Metadata Schema Requirements

Understanding the needs across multiple asset types.

- Technical specification group primarily focused on model metadata.

- Why?
    - Reduce effort required for deploying new models provided by third parties.
    - Pair descriptive data to models that can follow the model along the pipeline.

- Goal
    - Provide a base standard, with the ability to extend on top.
    - Provide a library of well defined fields, to reduce ambiguity.

Metadata Schema Version 1.0

Defining the foundation for metadata

- Focus on industry agnostic fields.
    - Examples:
        - description
        - creator information
        - etc.

- glTF is the primary target format for Version 1.0

- Remain flexible to allow…
    - extensions on top of Version 1.0.
    - future adoption for other model formats, if desired.
    - additional extensions for specific and unique use cases.

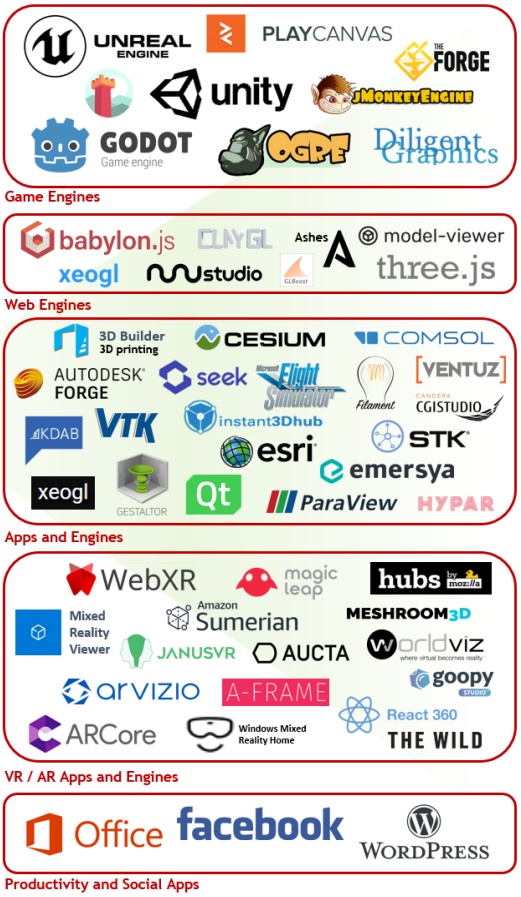
Rendering & Viewing

So let's talk a bit about getting your assets out there in the real world, and in front of clients' eyeballs.

glTF Renderers & Viewers

Choose your renderer based on what your clients want.

- Game engines for standalone experiences; interaction
- Web engines for website integration
- Apps for collaboration & industry workflow
- VR/AR for immersion and integration
- Social for traffic & engagement



As we mentioned earlier, glTF enjoys wide support across the 3D ecosystem. What you use depends on where you want your customers to consume your content.

Game engines are great for standalone rendering, providing easy ways for your clients to interact. Web engines are great for integrating directly into your sites. Apps are useful for in-product viewing, multi-user collaboration, and supporting specific industry workflows.

VR and AR are of course great for immersion and interaction; specifically for us at Wayfair it's really helpful for our customers to preview products in their spaces with Augmented Reality, to flip through material options, and to walk around to see them from multiple angles.

Social is also a key use-case for real-time 3D content; we've seen clear evidence that engagement really spikes when 3D content enters the picture.

Levels of Detail (LODs)

Downsample the source asset to work with specific targets:.

Another key presentation issue is making sure your 3D content runs smoothly. There's a lot of different hardware out there; so optimize your content to run well on your target platforms.

Here are some specs from the Asset Creation Guidelines; remember these are not hard rules either, they're simply targets to aim for. We know assets with these specs tend to render efficiently on these types of platforms.

▪ Desktop 3D:	250k triangles	2k textures	3mb file size
▪ Mobile AR/web:	150k triangles	2k textures	3mb file size
▪ 3D Room Planner:	40k triangles	1k textures	1mb file size
▪ 3D Banner Ad:	30k triangles	512 textures	500kb file size

Choose your main platform

Tune content to look best on your primary platform. All other platforms will look a bit different; glTF standards insure the model still renders OK.

So even though glTF enjoys wide support and a clear standard, there are still going to be differences in how renderers will present your models. See this chair for example, the fabric details are softer or more contrasty depending on the viewer. This is because renderers have different priorities; they each choose what they're optimizing for.

So to get the most bang for your buck, it helps to tune your content to look best where you are presenting it the most. By all means, test the content in as many places as you can. But optimize for where it matters most.

glTF Sample Viewer

Babylon.js Sandbox

Filament Viewer

<model-viewer>

<model-viewer> - Fidelity Test Results