glTF 2.0 Export in InstantUV

Max Limper | Fraunhofer IGD

@mlimper_cg
Context: InstantUV 3D Scan Optimization

Original 3D Scan

- High-Res. Input
- Decimate
- Low-Poly Mesh
- Segment
- Mesh + UV Seams
- Unwrap
- Mesh + UVs
- Bake Maps
- Textured Mesh
- Finalize for GPU
- Web-ready 3D Model

Web-ready 3D Model
Export Format?

• Must support different texture maps
  - Albedo
  - Normals
  - Occlusion
  - Metallic
  - Roughness

• Must be compact & fast to load (min. processing)

• Must align well with Web technologies

• Must be open (well-specified, no proprietary SDK)
gltf 2.0 – choose your renderer

web-ready 3d model

babylon.js

cesium

sketchfab

three.js
The Devil is in the Details

• “Ready to Render“ = Everything must be well-defined

• Example: Tangent Space Normal Maps
  - UV Origin?
  - Tangent Spaces?
  - Handedness?
  - Possible Flipping of Components from Texture?

• glTF specifies **everything** (in contrast to OBJ)!
Example: Tangent Space Normal Maps

InstantUV Export to Sketchfab, Default Parameters, MatCap Rendering

OBJ Version (Y Flipped)  glTF Version
Takeaways

• glTF 2.0 is our choice for ready-to-render 3D export

• Great progress and massive adaption with glTF 2.0

• Enabled us to serve different renderers without much effort