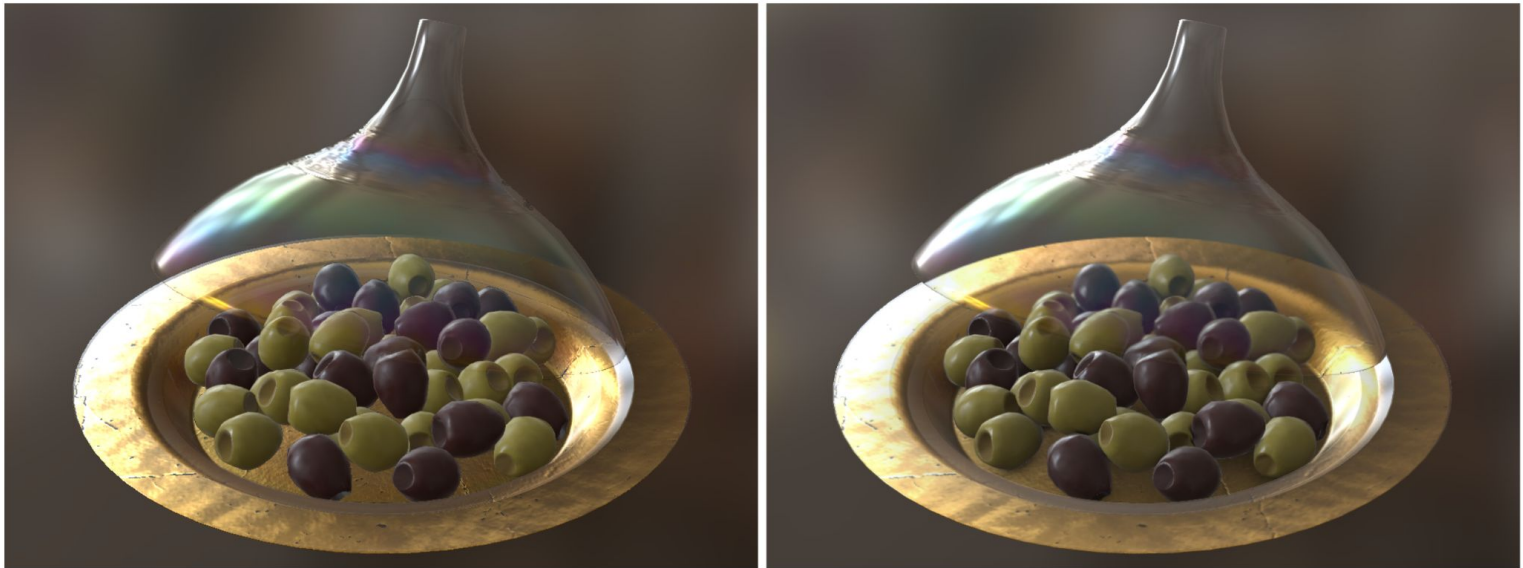


guc

a glTF to USD+MaterialX converter



Pablo Delgado

Motivation

- glTF PBR merged in MaterialX repo
- need strong link between both technologies

Project

- 8w development
- tested on ~300 assets
- Apache 2 license
- C API
- <https://github.com/pablode/guc>

Features

- no animation & skinning
- ND flattening & UsdShade inlining

Name	Status
KHR_lights_punctual	✓ Partial
KHR_materials_clearcoat	✓ Complete
KHR_materials_ior	✓ Complete
KHR_materials_sheen	✓ Complete
KHR_materials_specular	✓ Complete
KHR_materials_transmission	✓ Complete
KHR_materials_volume	✓ Partial

Conversion

1. export images
2. create UsdPreviewSurface & MaterialX materials
3. traverse scene graph depth-first
 - unique paths
 - instancing

gITF PBR

@proog128 / Tobias Häußler

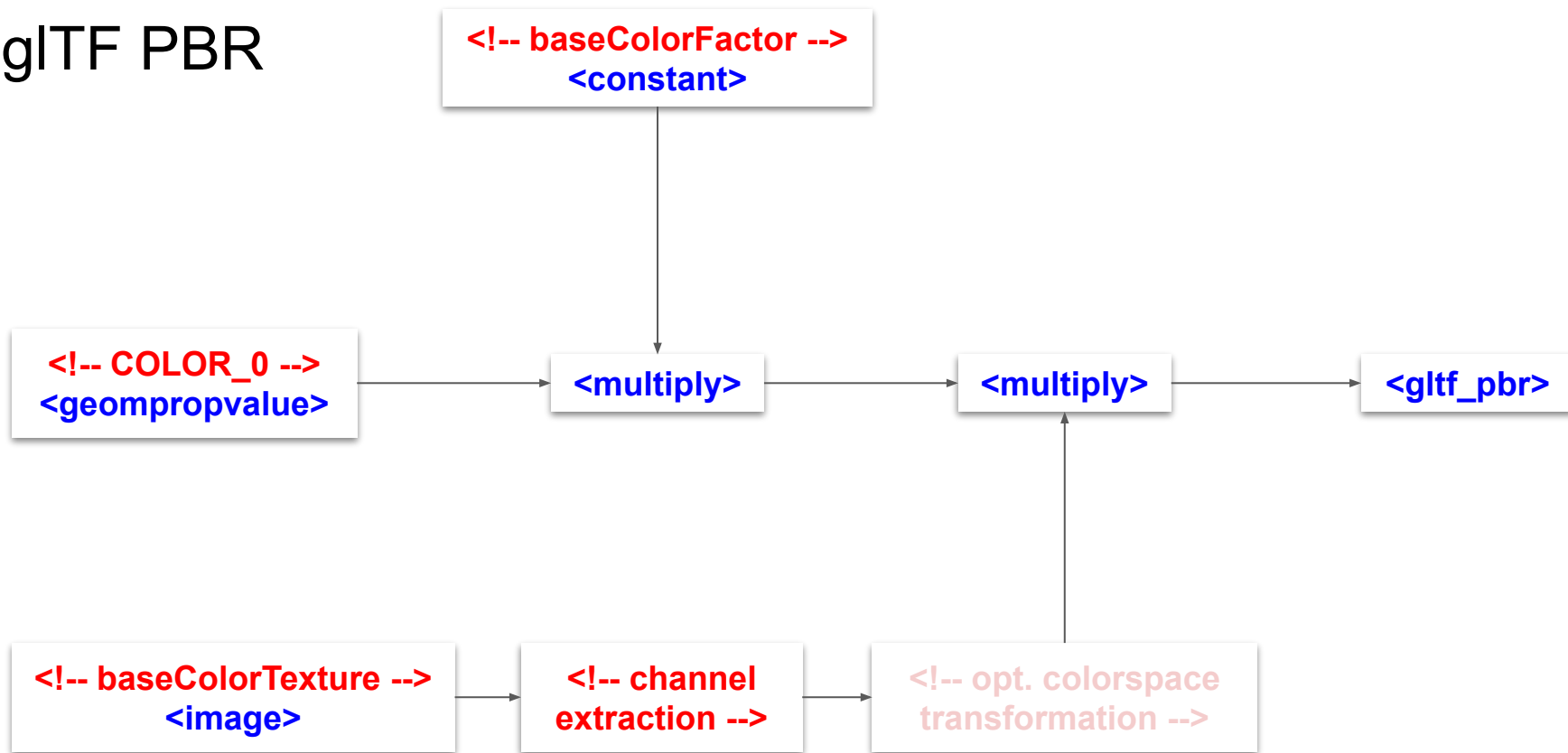
```
<nodedef name="ND_gltf_pbr_surfaceshader">
  <input name="base_color" />
  <input name="metallic" />
  <input name="roughness" />
  <input name="normal" />
  <input name="occlusion" />
  <input name="transmission" />
  <input name="specular" />
  <input name="specular_color" />
  <input name="ior" />
  <input name="alpha" />
  <input name="alpha_mode" />
  <input name="alpha_cutoff" />
  <input name="sheen_color" />
  <input name="sheen_roughness" />
  <input name="clearcoat" />
  <input name="clearcoat_roughness" />
  <input name="clearcoat_normal" />
  <input name="emissive" />
  <input name="emissive_strength" />
  <input name="thickness" />
  <input name="attenuation_distance" />
  <input name="attenuation_color" />
</nodedef>
```

UsdMtlx

```
<nodegraph name="NG">
  <image name="node1" />
  <!-- ... -->
</nodegraph>
<glTF_pbr name="SR" type="surfaceshader">
  <!-- ... -->
</glTF_pbr>
<surfaceMaterial name="MAT" type="material">
  <!-- ... -->
</surfaceMaterial>
```

- all nodes except surface in nodegraph

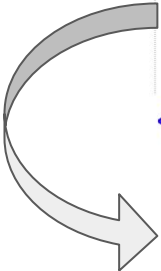
glTF PBR



UsdMtlx

`<surface>`

```
<nodegraph name="NG">
  <image name="node1" />
  <!-- ... -->
  <gltf_pbr name="SR" type="surfaceshader">
    <!-- ... -->
  </gltf_pbr>
</nodegraph>
```



- flattening requires surface extraction

Tangents

- glTF tangents have 4 components
 - `w` flips bitangent
- MaterialX tangents have 3 components
- solution: precalculate bitangents
- `custom <normalmap> node`

Output


F:\CesiumMilkTruck.usdc

File Edit View Window

/Materials/MaterialX/Materials/wheels

Navigation Show

Prim Name	Type	Vis
root		
Geom	Scope	V
Yup2Zup	Xform	V
Cesium_Milk_Truck	Xform	V
Materials	Scope	V
UsdPreviewSurface		
MaterialX		
Materials		
wheels	Material	
truck	Material	
glass	Material	
window_trim	Material	
Shaders		
NodeGraphs		



Hydra: GL

Render: 4.68 ms (217.92 FPS)
Playback: N/A

Camera: Free
Complexity: Low

? Search for prim by name Find Prim

Type	Property Name	Value
Ⓜ	inputs:alpha	1.0
Ⓜ	inputs:alpha_cutoff	0.5
Ⓜ	inputs:alpha_mode	0
Ⓜ	inputs:attenuation_color	(1, 1, 1)
Ⓜ	inputs:attenuation_distance	
Ⓜ	inputs:base_color	
Ⓜ	inputs:clearcoat	0.0

? Search for property by name Find Prop

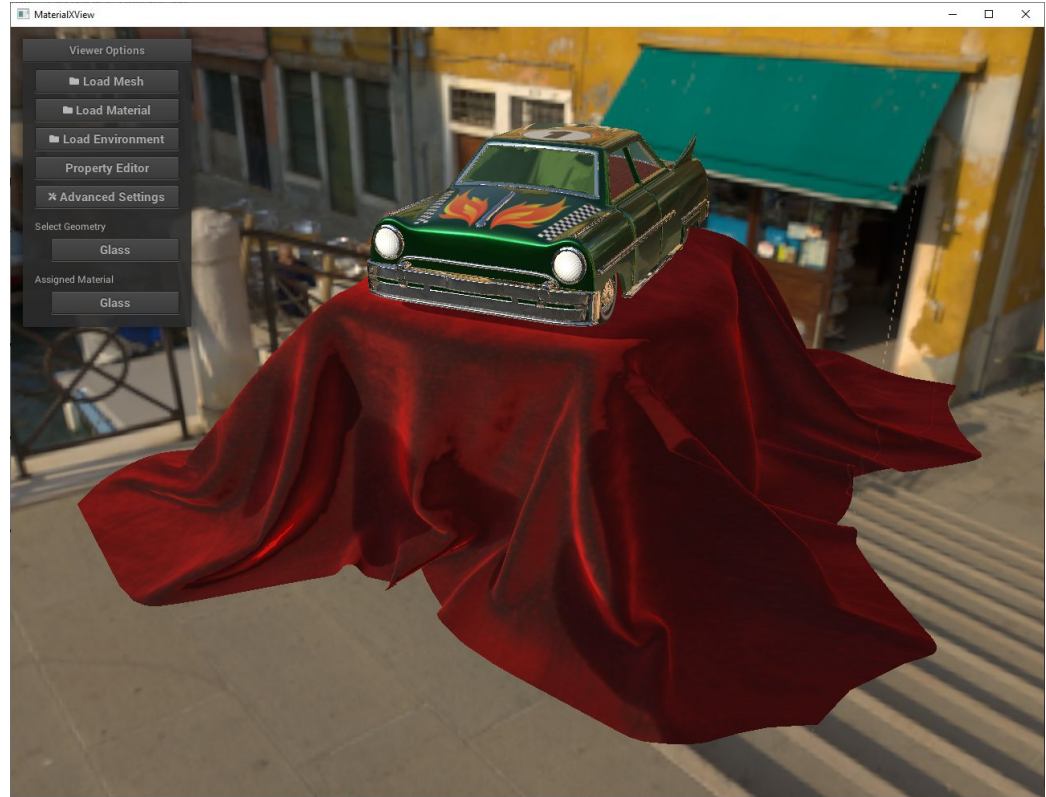
Value	Meta Data	Layer Stack	Composition
-------	-----------	-------------	-------------

0.0 Redraw On Frame Scrub Step Size 1.0 0.0 Play

Frame: 0.0

MaterialXView Compat

- #define, not user-facing
- manual material assignment
- vertex colors ignored



Future

- 1.0 release this or next year
- test suite

Questions / Comments?