
open
Source
days '25
#ASWF
/* ACADEMY SOFTWARE FOUNDATION

open source days '25

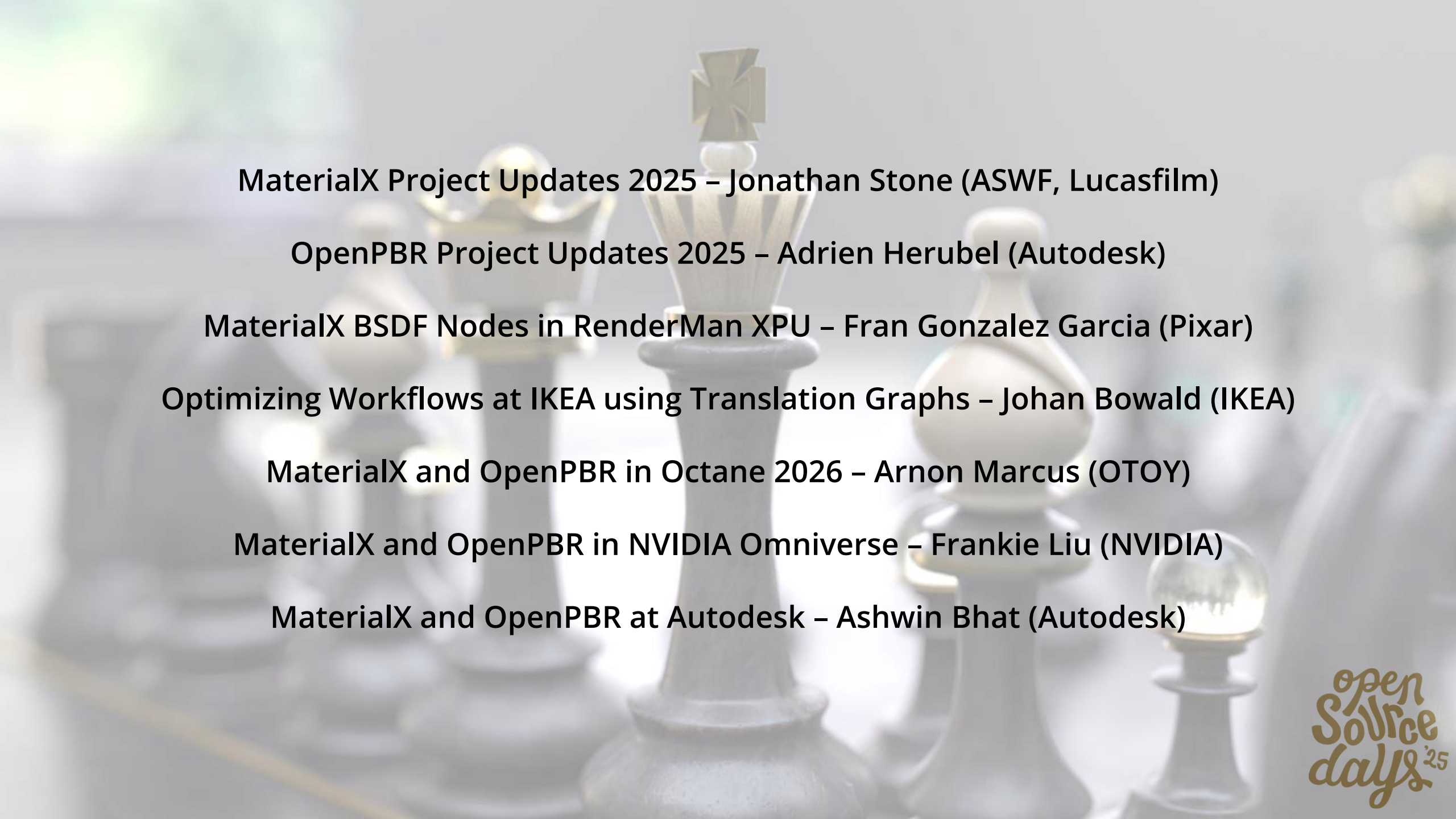
/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

MaterialX and OpenPBR Town Hall

August 5th, 2025

#ASWF



MaterialX Project Updates 2025 – Jonathan Stone (ASWF, Lucasfilm)

OpenPBR Project Updates 2025 – Adrien Herubel (Autodesk)

MaterialX BSDF Nodes in RenderMan XPU – Fran Gonzalez Garcia (Pixar)

Optimizing Workflows at IKEA using Translation Graphs – Johan Bowald (IKEA)

MaterialX and OpenPBR in Octane 2026 – Arnon Marcus (OTOY)

MaterialX and OpenPBR in NVIDIA Omniverse – Frankie Liu (NVIDIA)

MaterialX and OpenPBR at Autodesk – Ashwin Bhat (Autodesk)

open source days '25

MaterialX Project Updates 2025

Jonathan Stone, ASWF, Lucasfilm ADG

MaterialX 1.39.2 Release

- Chiang Hair BSDF (NVIDIA)
- Generic Color Ramps (Autodesk)
- Improved Worley Noise (SideFX)
- Disney Principled shading graph
- Shader generation optimizations

MaterialX 1.39.3 Release

- Major collaboration between the MaterialX and OpenUSD teams
- MaterialX 1.39.3 is the default version in USD 25.05 and beyond
- Provides access to OpenPBR Surface in USD applications

ASWF Dev Days

- Provides dedicated mentorship to new ASWF contributors
- Merged 20 contributions to MaterialX over the past two events
- Next event will be on September 25, see https://www.aswf.io/dev-days/ for details

Upcoming Work

- MaterialX 1.39.4 Release Candidate
  - Many new features and optimizations, highlighted by our speakers today
- Integration of the NanoColor spaces in MaterialX
- Generation of Slang shading code from MaterialX graphs

Join the Conversation

- Visit www.materialx.org to learn more about the project
- Visit https://www.aswf.io/get-involved/ to join the conversation
- Visit https://www.aswf.io/dev-days/ to participate in ASWF Dev Days

open Source days '25

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

OpenPBR Project Update 2025

Adrien Herubel, Autodesk

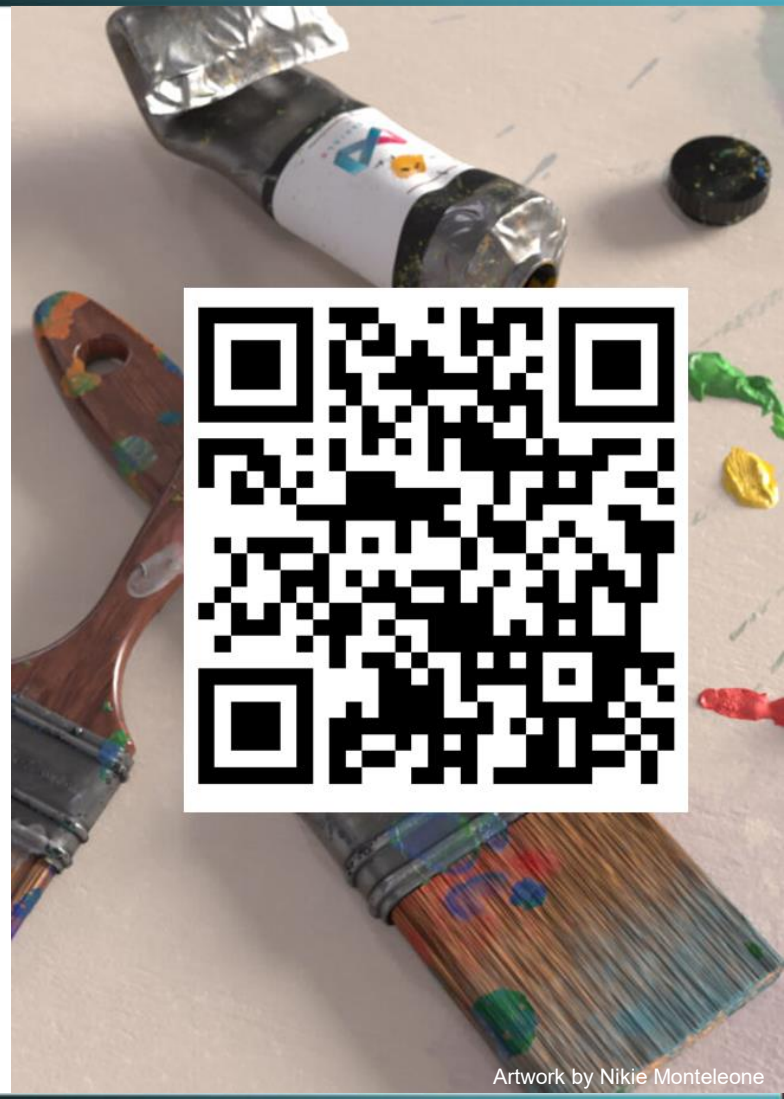
MATERIALX

openpbr

#ASWF

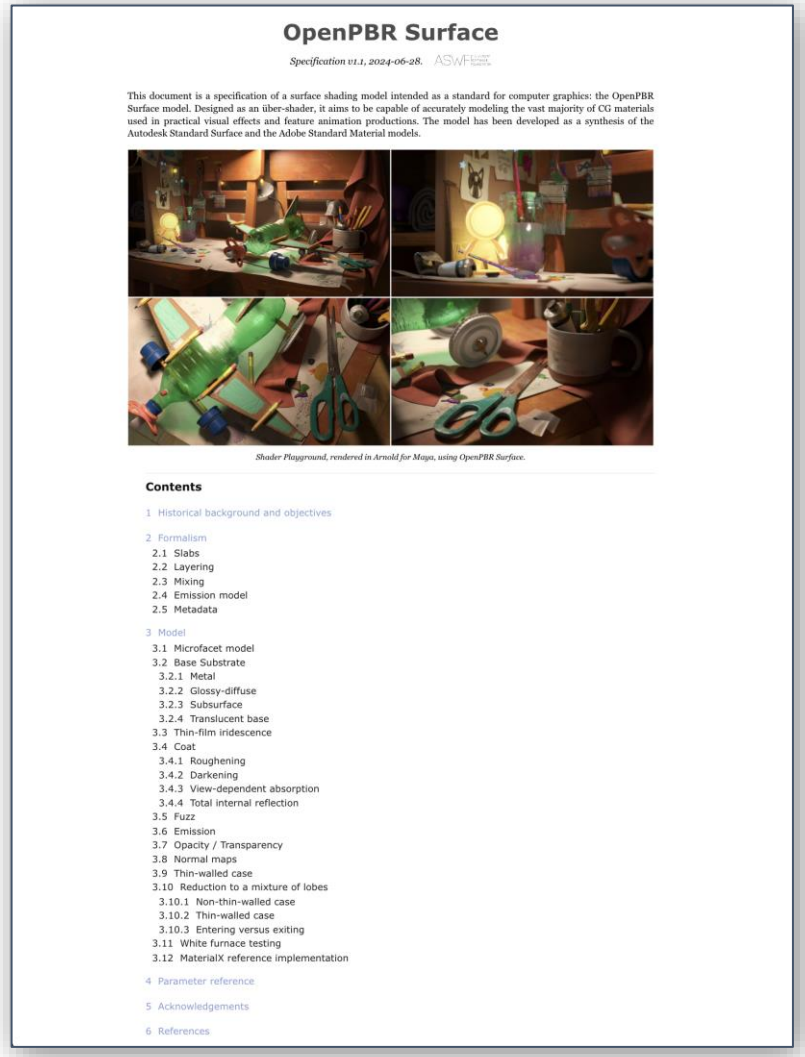
Introduction

- The OpenPBR project
- 2024-2025 Achievements
- What we are working on now
- What is next
- OpenPBR at SIGGRAPH



The OpenPBR project

- A new surface appearance model
  - Physically based & artist friendly
  - Open governance
  - Backed by the industry
- The OpenPBR specification
  - Written for implementers
  - Precise description of the expected appearance
  - Parameter reference
- The OpenPBR reference implementation
  - MaterialX graph



OpenPBR Surface

Specification v1.1, 2024-06-28. ASWF

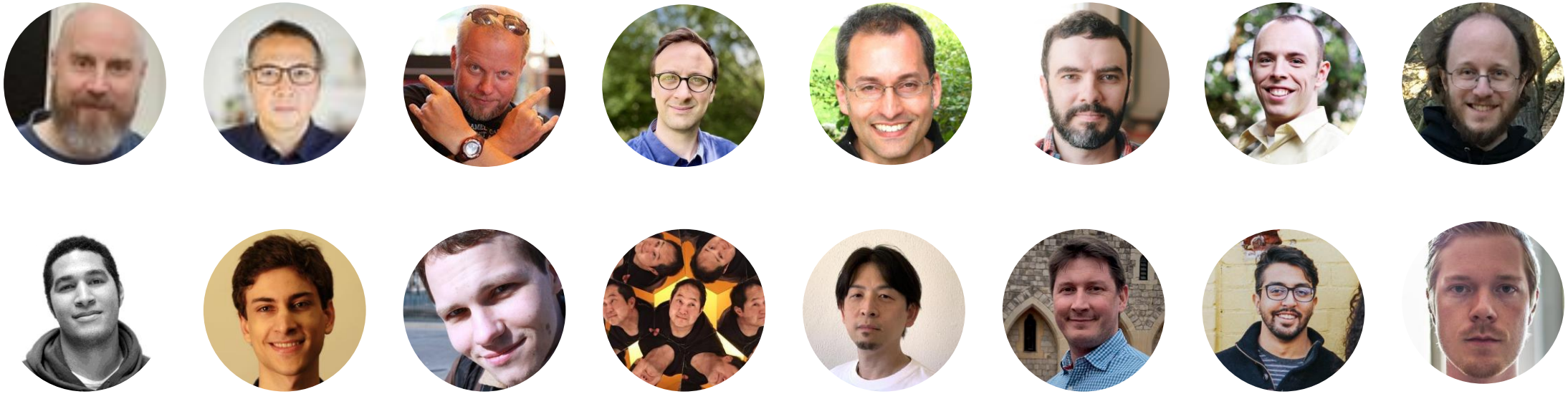
This document is a specification of a surface shading model intended as a standard for computer graphics: the OpenPBR Surface model. Designed as an über-shader, it aims to be capable of accurately modeling the vast majority of CG materials used in practical visual effects and feature animation productions. The model has been developed as a synthesis of the Autodesk Standard Surface and the Adobe Standard Material models.

Shader Playground, rendered in Arnold for Maya, using OpenPBR Surface.

Contents

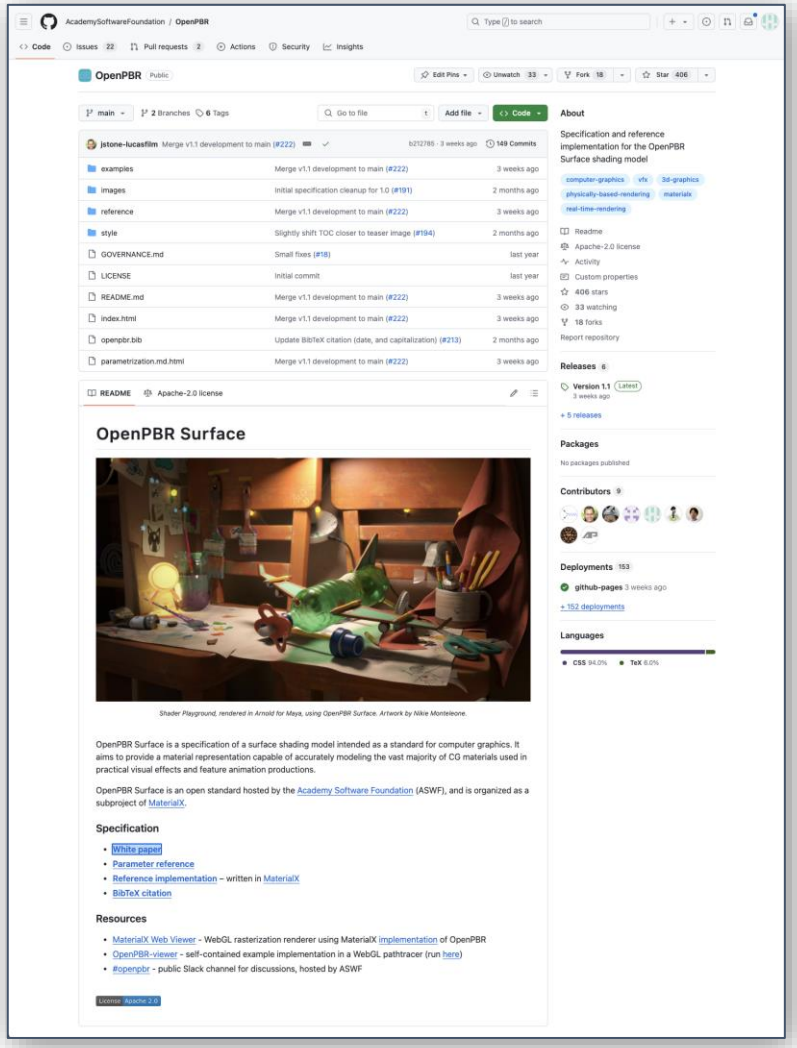
Contributors

- Contributors from Adobe, Apple, Autodesk, Epic Games, ILM, NVIDIA, SideFX, …



How to contribute

- On GitHub : Issues and PRs
  - https://github.com/AcademySoftwareFoundation/OpenPBR
- ASWF Slack
  - #openpbr
- ASWF Zoom OpenPBR meeting
  - Bi-weekly, alternating with the MaterialX meeting



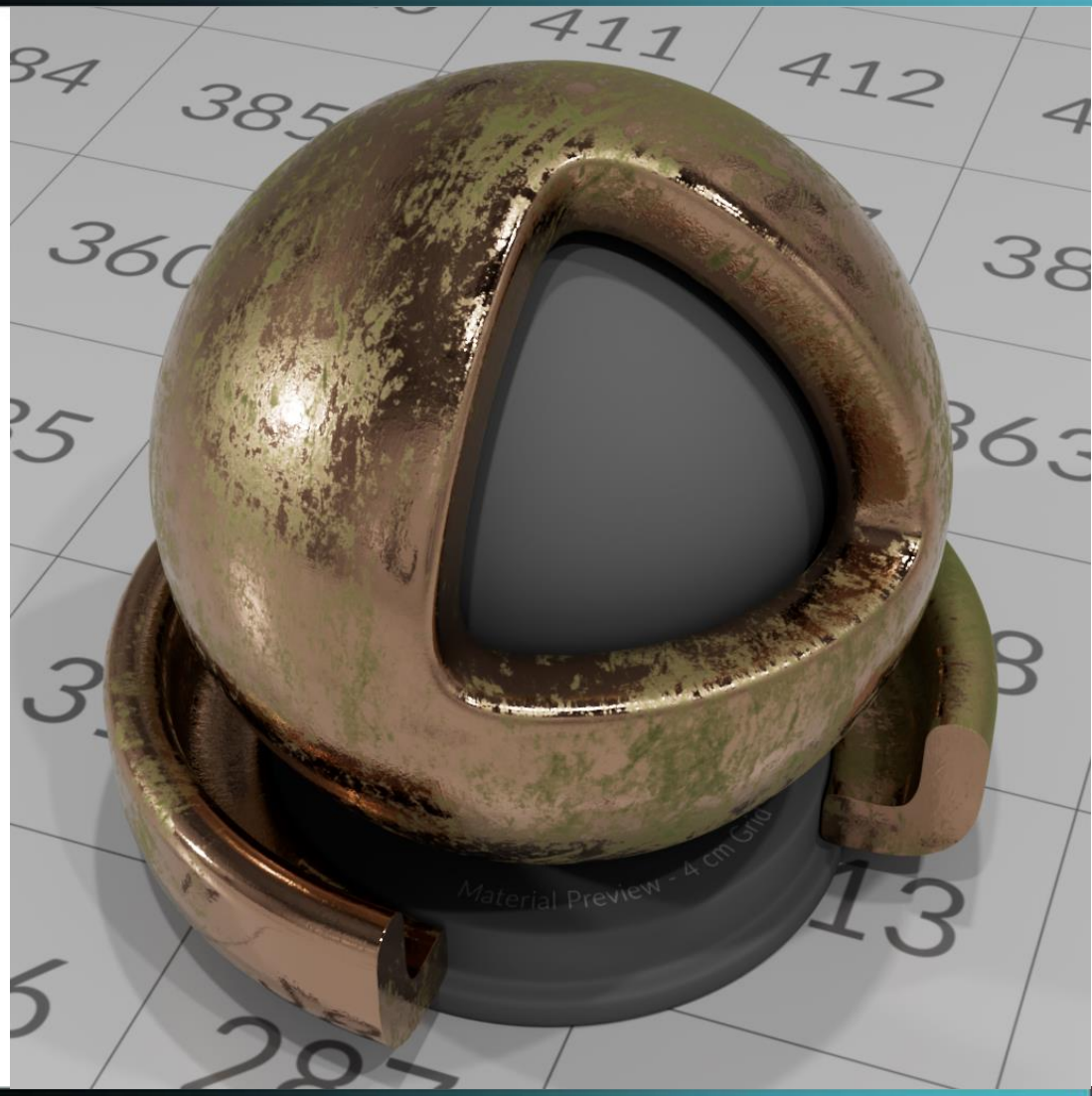
2024-2025 Achievements

- OpenPBR 1.1 in MaterialX and USD
  - MaterialX 1.38.10-openpbr and USD 24.02
  - MaterialX 1.39 and USD 25.05
- Adoption
  - Default material in Autodesk Maya, 3ds Max, and Arnold
  - Omniverse
  - VRay
  - Redshift 2025.2
  - Blender 4.5
- OpenPBR 1.1.1
  - Follow semantic versioning
  - Alternative implementations
  - Clarifications and fixes

OpenPBR 1.2

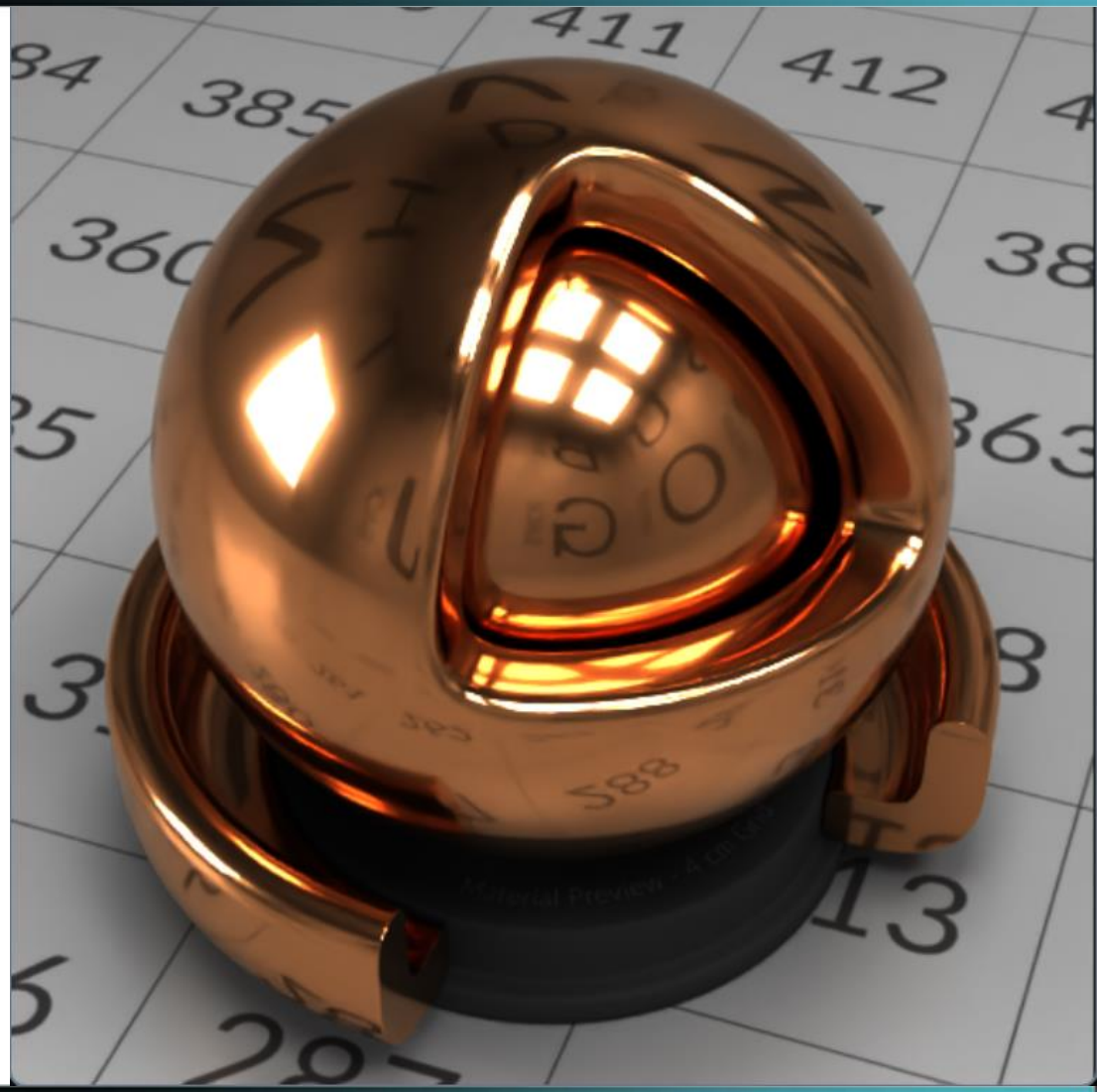
- Work in progress
- New features
- Improved correctness
- Addressing user feedback
- Versioning

- In scope for 1.2 :
  - Specular haze / tail controls
  - Retroreflection
  - Improved coat darkening model
  - New roughening model
  - Decoupled IOR
  - New *emission_weight* controls

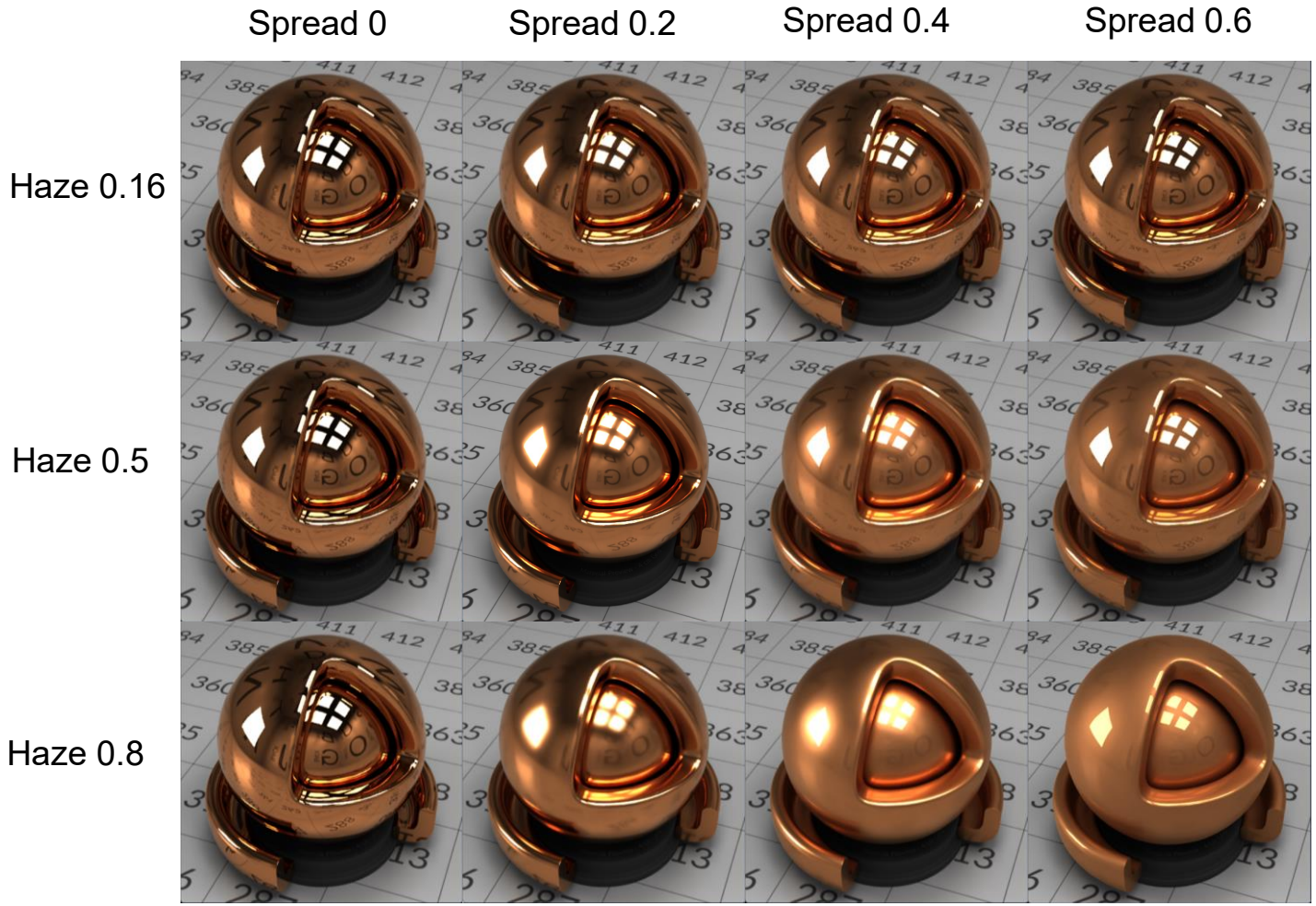


OpenPBR 1.2 : Specular Haze

- Represent smudged reflective surfaces
- New **specular haze** and **haze spread** parameters
- Mix two BSDFs
- PR#254

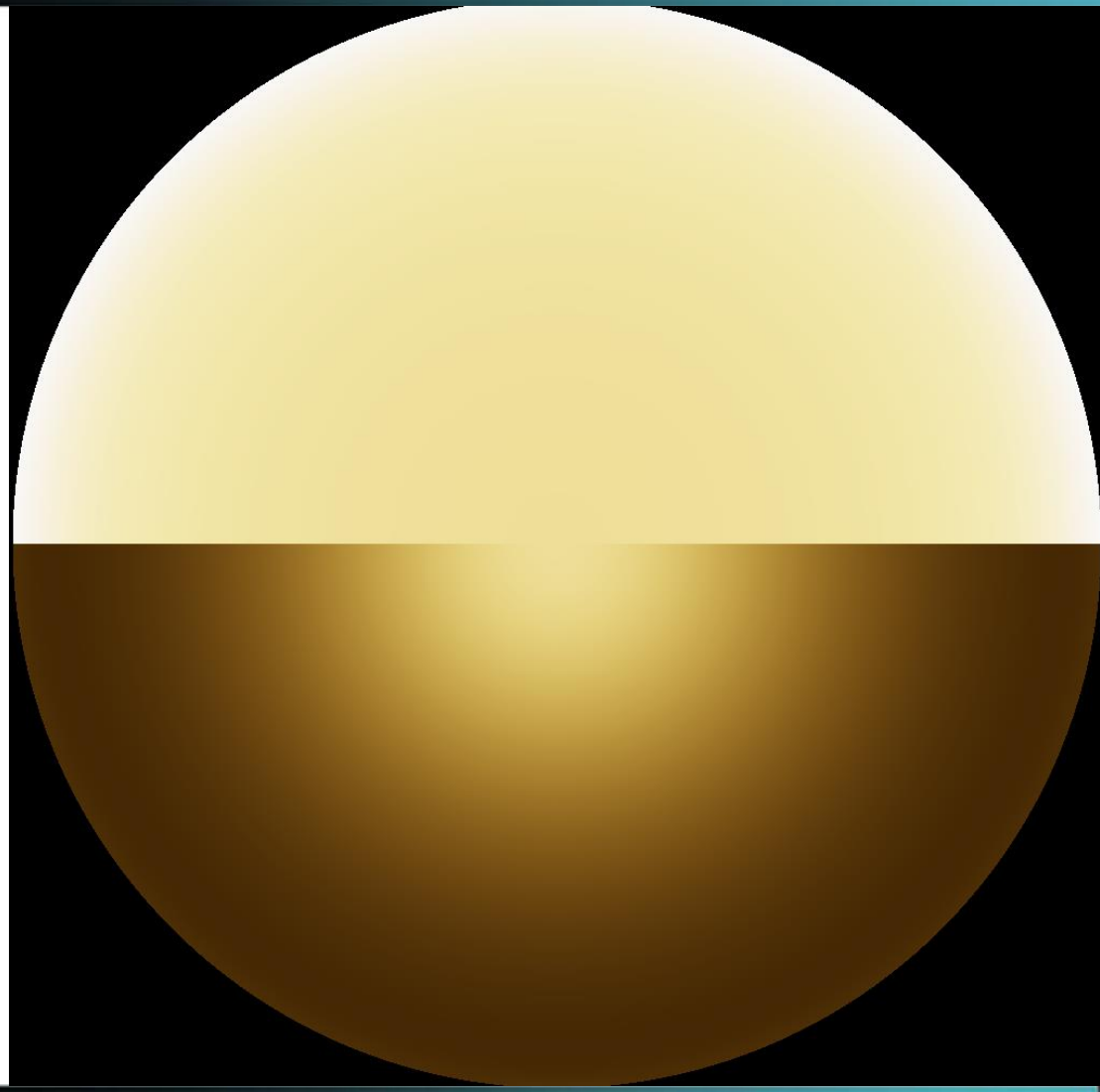


OpenPBR 1.2 : Specular Haze

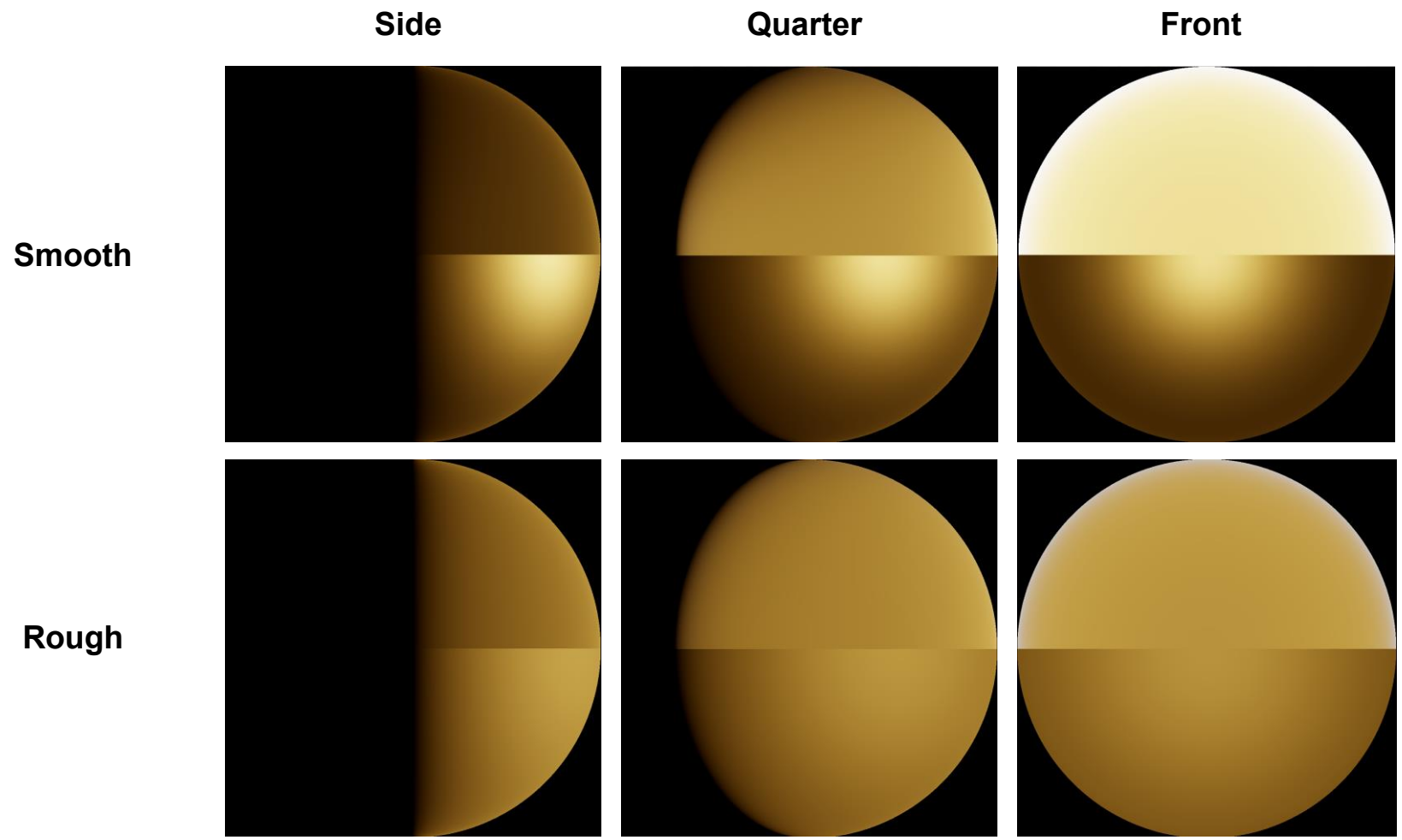


OpenPBR 1.2 : Retro-reflectivity

- Represent common retro reflective materials
- Modified microfacet BRDF
- PR#255



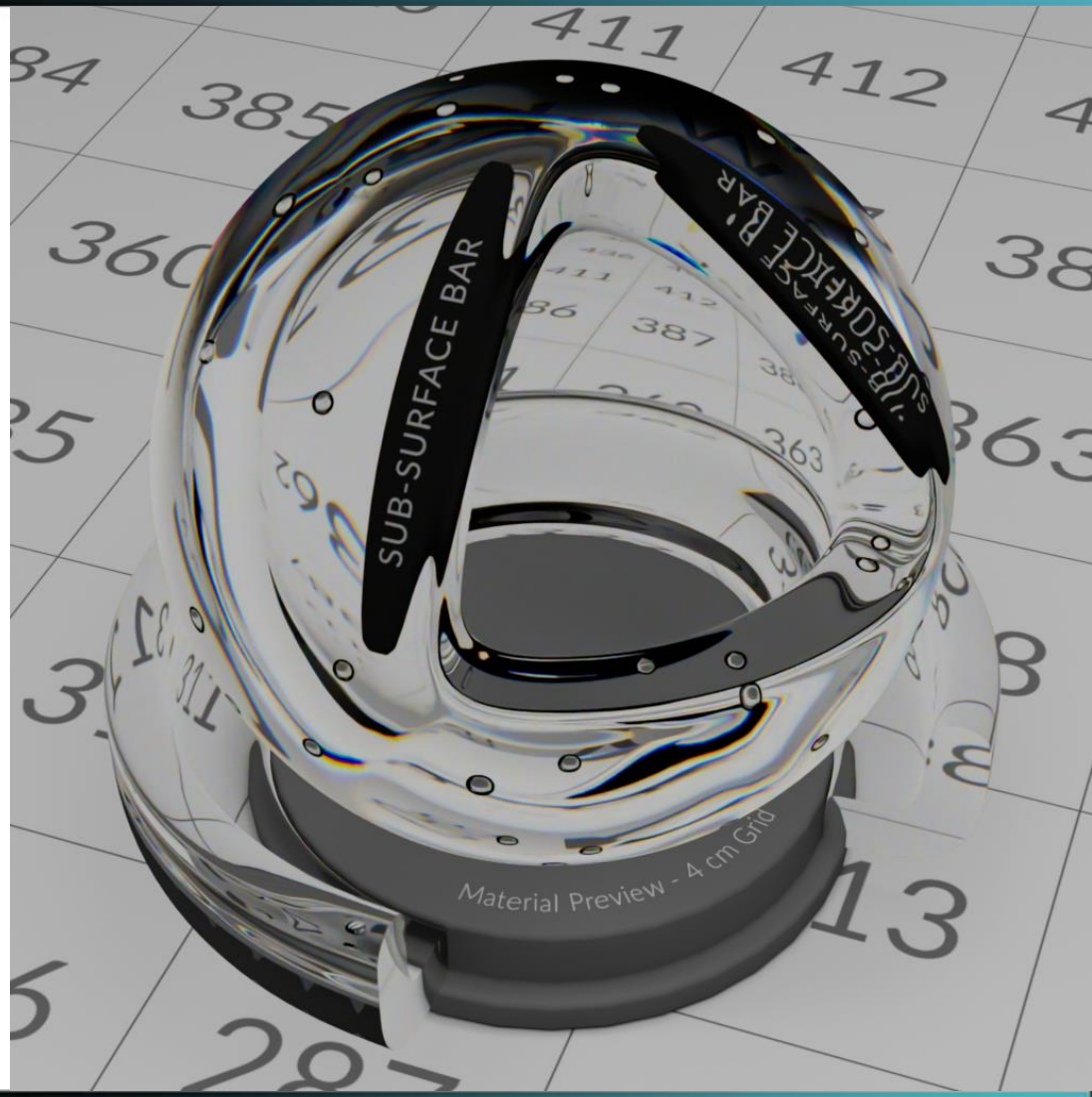
OpenPBR 1.2 : Retro-reflectivity



Smooth and rough retro reflectors with different light directions

OpenPBR 1.2 : Decoupled IOR

- Stop specular_weight from affecting refractions
- New specular_weight formula
- PR#247



OpenPBR 1.2 : Decoupled IOR

Specular Weight 1

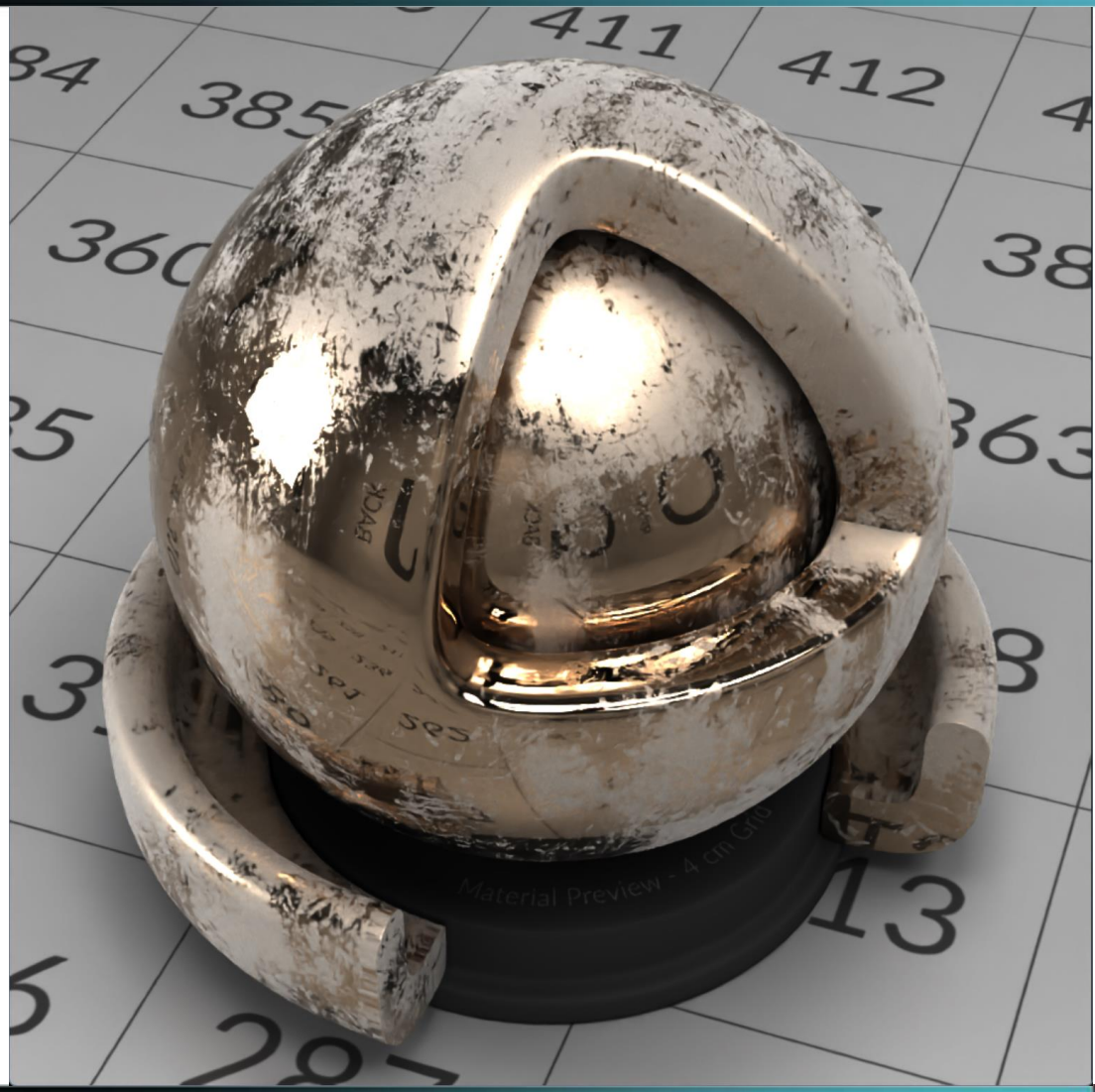
Specular Weight 0.01

OpenPBR 1.1

OpenPBR 1.2

OpenPBR 1.2 : Roughening

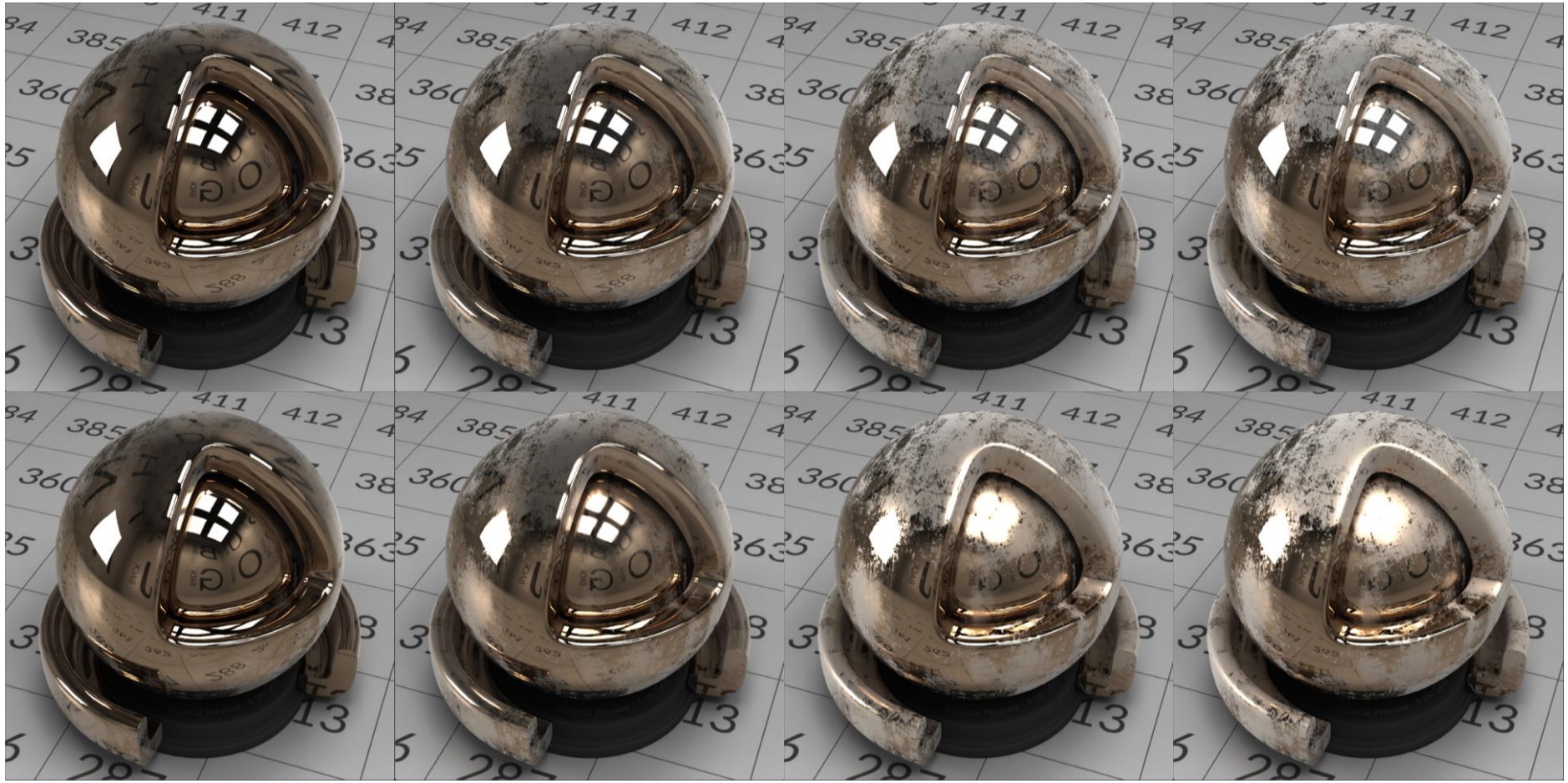
- Modify roughness of underlying layers
- Avoid unrealistic looks and manual steps
- PR#261



OpenPBR 1.2 : Roughening

No roughening

Roughening



Varying **fuzz roughness** on a smooth metal base

Announcing OpenPBR Volume

OpenPBR Volume

- An additional appearance model for volume assets
- Clouds, smoke, fire, explosions
- Collaboration between Autodesk and SideFX
- Combining Houdini Pyro and Arnold Standard Volume
- Work in progress, not public yet
- Same goals as OpenPBR Surface
  - Specification describing appearance and parametrization
  - MaterialX reference implementation
- Reach out to us if interested

AUTODESK

SideFX®

Future work

- OpenPBR 1.2
- Real-time implementations
- Beyond surfaces
  - OpenPBR Volume
  - OpenPBR Hair

Walt Disney Animation Studios Cloud Data Set

OpenPBR at SIGGRAPH

DigiPro 2025 | OpenPBR Surface: An Open Shading Model for Physically Based Materials
Saturday 9 August 2025 - 9:00am PST

SIGGRAPH Course | Physically Based Shading in Theory and Practice
Sunday 10 August 2025 - 9:00am PST

Open Source Days | OpenPBR Community Birds of a Feather
Monday 11 August 2025 - 1:00pm PST

Adobe Substance Days | OpenPBR Panel
Monday 11 August 2025 - 6:30pm PST

MATERIALX BSDF Nodes in RENDERMAN XPU

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

Fran González
Sr. Software Engineer

Katrin Bratland
Sr. Software Engineer

Akshay Shah
Sr. Software Engineer

Jonathan Stone
Lead Rendering Engineer
Materials and Shading

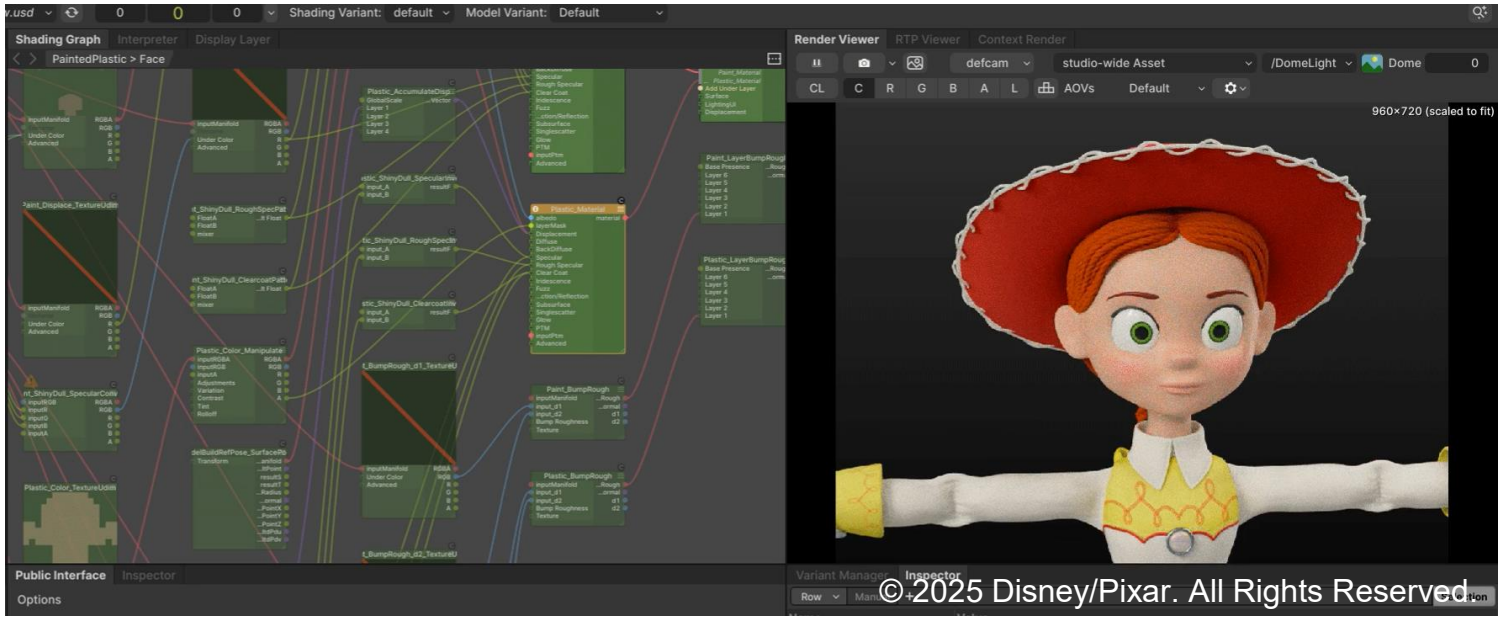
Mark Manca
Director, Product Mgmt

Content

Introduction

RENDERMAN XPU

- Pixar's next-gen hybrid CPU/GPU rendering architecture
- Scale & complexity of Pixar's feature animation projects.
- Interactive workflows & final frame rendering



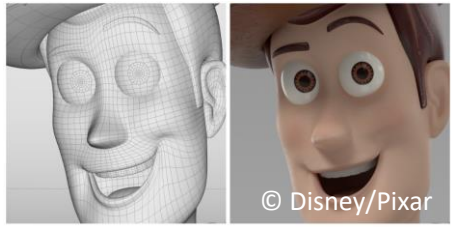
Elio © Disney/Pixar

Introduction

RENDERMAN XPU

- Materials
  - Pixar Materials: PxrSurface, PxrDisneyBsdf, PxrDiffuse

Elio's room rendered at 2K with RenderMan XPU with 256spp




RenderMan XPU: A Hybrid CPU+GPU Renderer for Interactive and Final-Frame Rendering

Introduction

RENDERMAN XPU

- Materials
  - Pixar Materials: PxrSurface, PxrDisneyBsdf, PxrDiffuse
  - MATERIALX Lama
    - Conductor, Dielectric, Diffuse, Emission, GeneralizedSchlick, HairChiang, Iridescence, Sheen, SSS, Surface, Translucent
    - Combiner Nodes: Add, Mix and Layer

Elio © Disney/Pixar

Introduction

Why MATERIALX ?

- Open standard description of rich materials
- Platform independent & exchange across renderers
- Aligment with ASWF to unify workflows across the industry
- We already support MATERIALX Lama

MATERIALX

The Open Chess rendered in RenderMan RIS with HdPrman
And shaderGen translation to PxrSurface

Collaboration with Lucasfilm

Pixar + Lucasfilm Collaboration

- Shared goals on shading standardization and adoption across productions at Disney
- Joint efforts in implementing MATERIALX BSDF Nodes

PIXAR

MATERIALX

LUCASFILM Ltd

Jonathan Stone
Lead Rendering Engineer, Materials and Shading

MaterialX BSDF Nodes in XPU

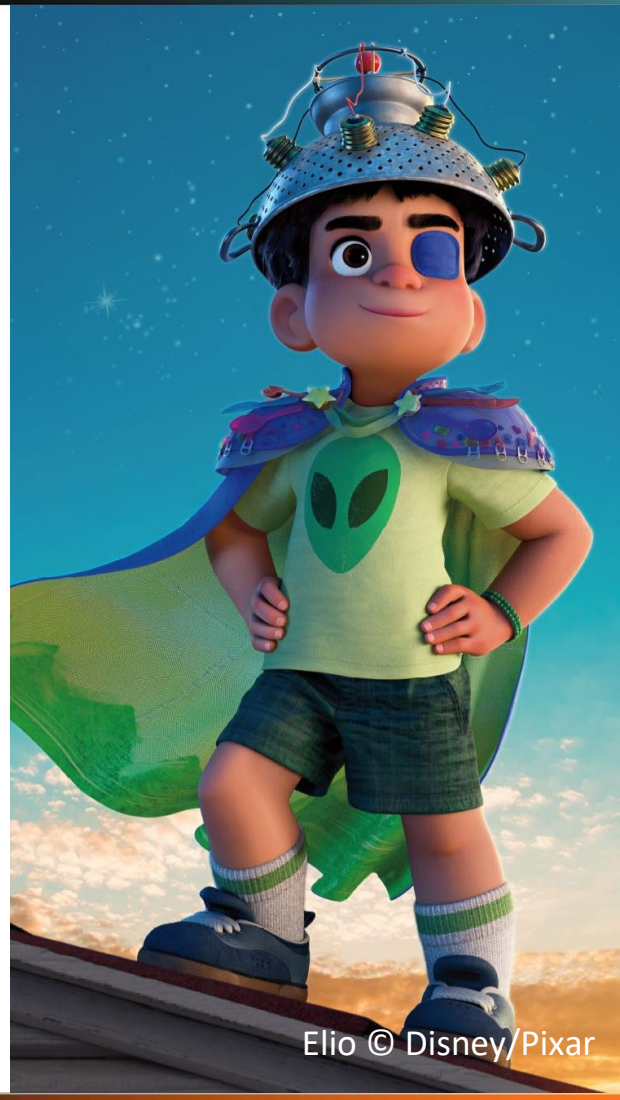
Goal

- **Native** MATERIALX BSDF support in XPU for artist-friendly, interoperable shading.
  - Supporting arbitrary BxDF combinations with physical correctness.
  - Maintaining performance on both CPU and GPU devices.

PIXAR

MATERIALX

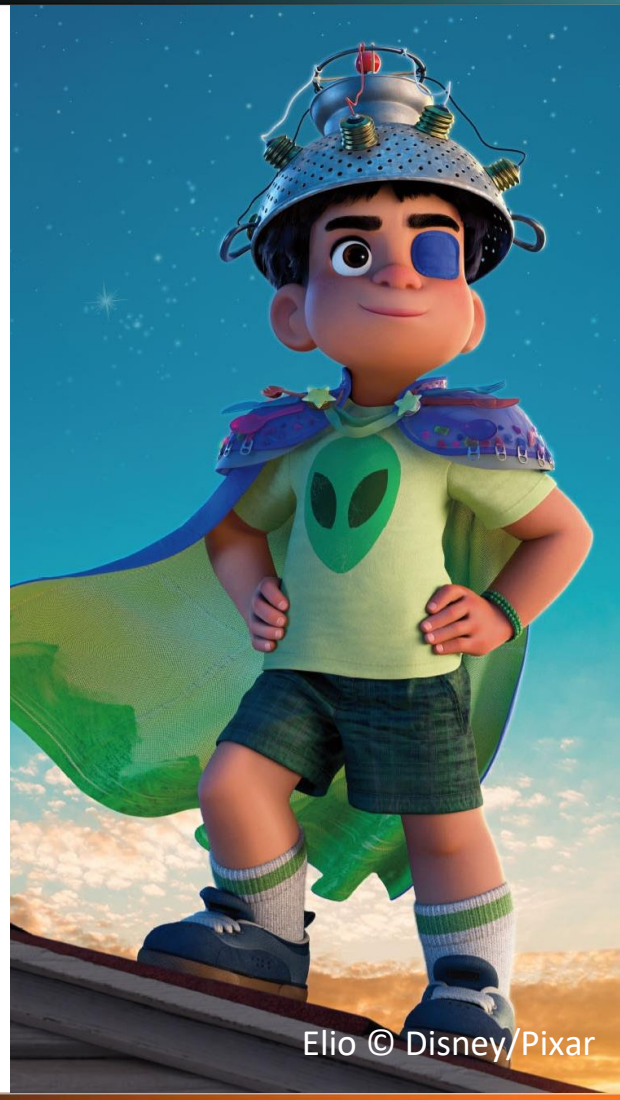
LUCASFILM Ltd



MaterialX BSDF Nodes in XPU

Materials in RenderMan XPU

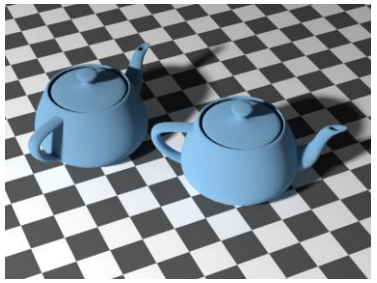
- Bxdfs are implemented as plugins
  - Easily expand XPU's capabilities
  - Kernels executed in both CPU and GPU devices sharing most of the code
  - Usage of custom OSL material closures representing Bxdfs



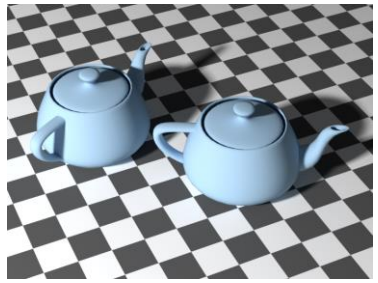
MaterialX BSDF Nodes in XPU

Current Status

- **Native** support for MATERIALX BSDF nodes in XPU
  - Oren Nayar Diffuse
  - Burley Diffuse
  - Dielectric
  - Conductor
  - Generalized Schlick
  - Translucent
  - Subsurface (WIP)
  - Sheen
  - Chiang Hair (TBD)

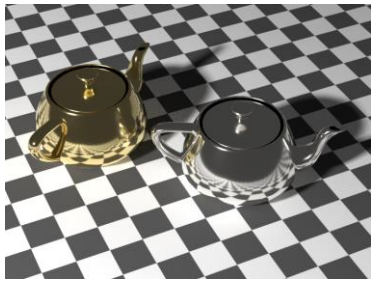


Oren Nayar Diffuse

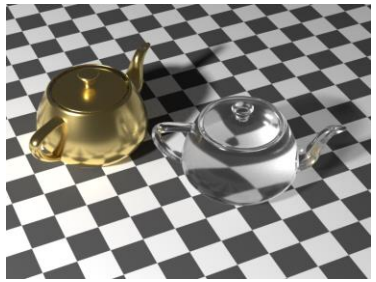


Burley Diffuse

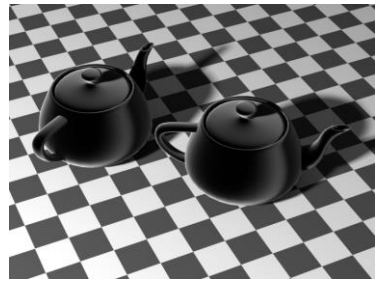
Dielectric



Conductor



Generalized Schlick

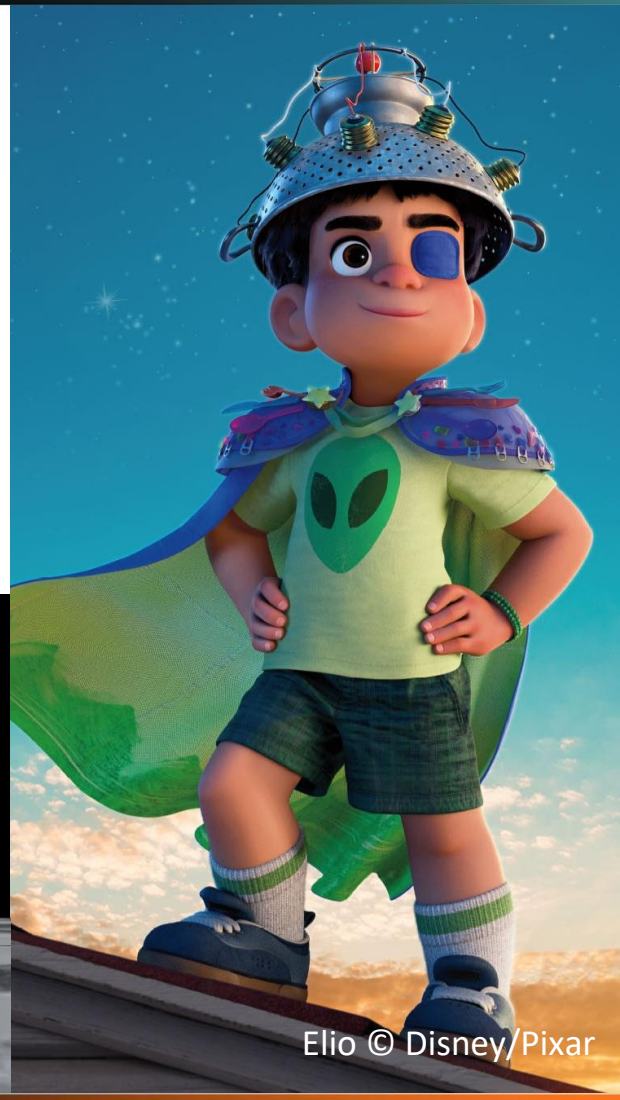


Sheen

MaterialX BSDF Nodes in XPU

Current Status

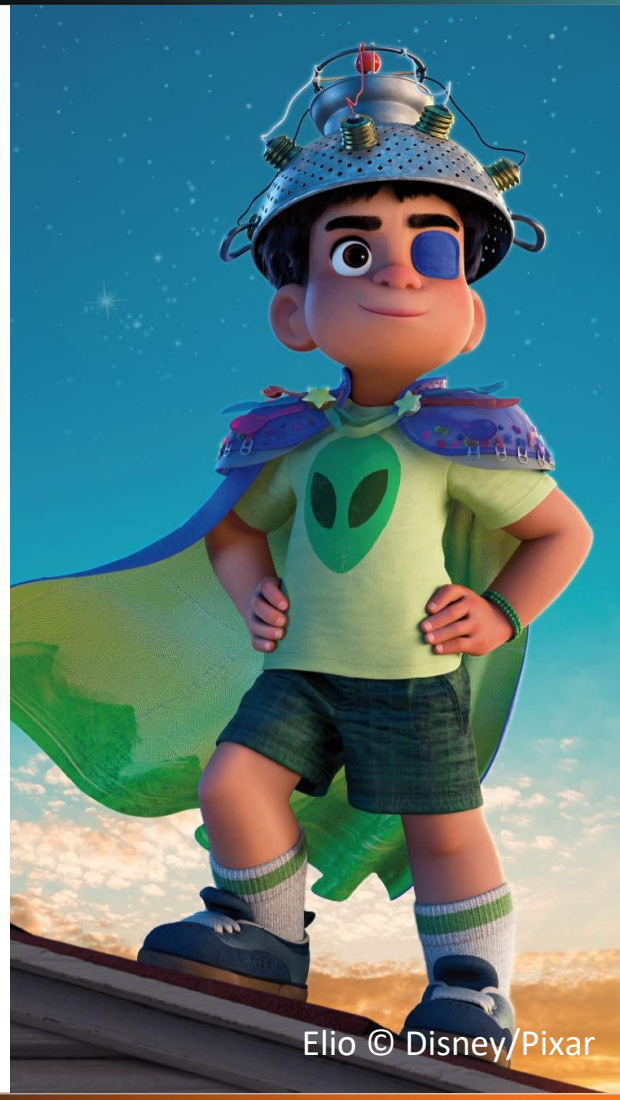
- **Native** support for MATERIALX BSDF nodes in XPU
  - First citizen Bxdf plugin in XPU
  - Rely on OSL Material closures
  - Shared code between MaterialX BSDFs and MaterialX Lama



MaterialX BSDF Nodes in XPU

Current Status

- **Native** support for MATERIALX BSDF nodes in XPU
  - First citizen Bxdf plugin in XPU
  - Rely on OSL Material closures
  - Shared code between MATERIALX BSDFs and MATERIALX Lama
- Combiner / utility nodes
  - **Add** and **mix** nodes fully implemented
  - **Layer** is work in progress

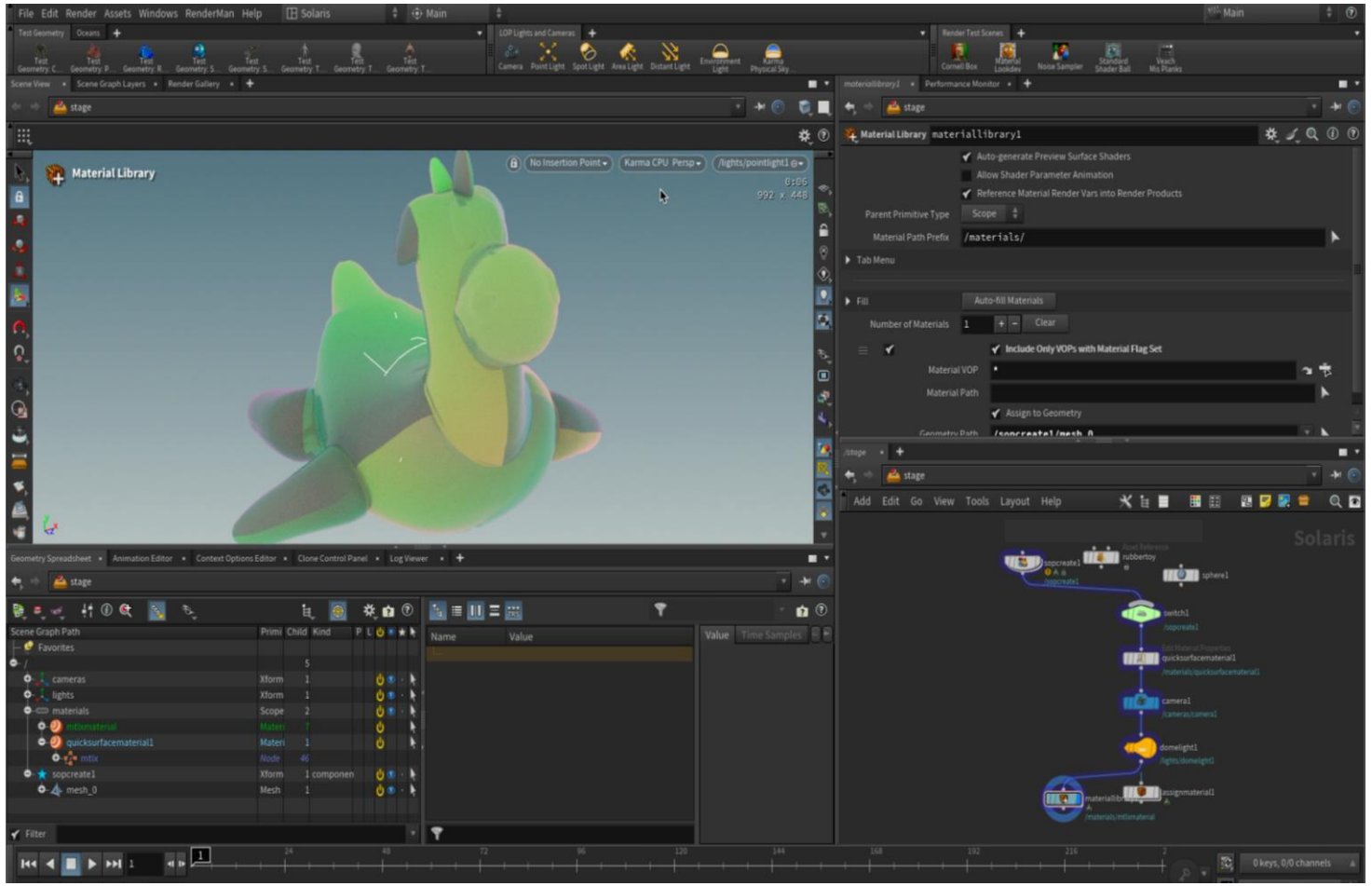


MaterialX BSDF Nodes in XPU

MaterialX BSDF Nodes in XPU

Real world use & Integration

- Solaris + HdPrman
  - Early Integration of RenderMan XPU supporting MATERIALX in Solaris
  - Initial work done to support MATERIALX StandardSurface
- LookdevX in Maya



MaterialX BSDF Nodes in XPU

What's Next?

- Finish support for layering
- Support MaterialXStandardSurface in XPU
  - Native MATERIALX BSDF nodes
  - Combiner nodes (Add, mix, layer, etc).
- OpenPBR adoption via MATERIALX

MaterialX BSDF Nodes in XPU

Conclusions

- **MATERIALX BSDFs are now first-class citizens in RenderMan XPU**
  - Native integration supports physically-correct, production-ready shading.
- **Built with collaboration and scalability in mind**
  - A joint effort between Pixar and Lucasfilm to unify shading across studios and platforms.
- **Optimized for both interactivity and final frame**
  - One shading system, seamlessly running on both CPU and GPU devices.

MaterialX BSDF Nodes in RENDERMAN XPU

Thanks for your attention!

RenderMan XPU: A Hybrid CPU+GPU Renderer for Interactive and Final-Frame Rendering

open source days '25

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

Optimizing Workflows at IKEA using Translation Graphs

2025-08-05

#ASWF

IKEA

MATERIALX

A bit of background

20,000 articles at IKEA website

BILLY was introduced 1979,
is still sold today

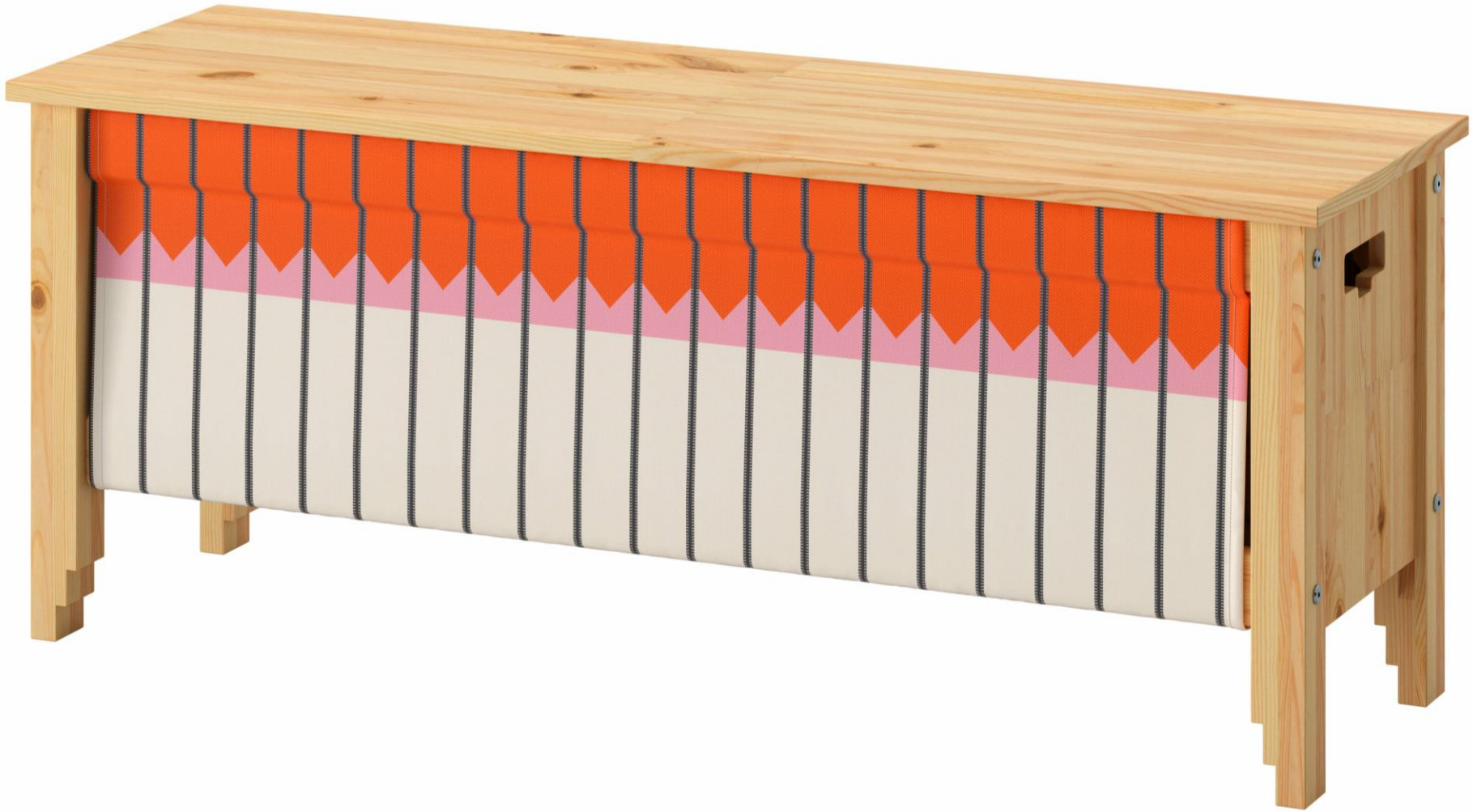
What we deliver

Material Library

New

OMMJÄNGE

Bench with storage



What is a Translation Graph?

MaterialX and Translation graphs

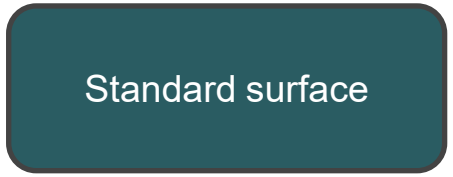
Translation Graph

Standard surface

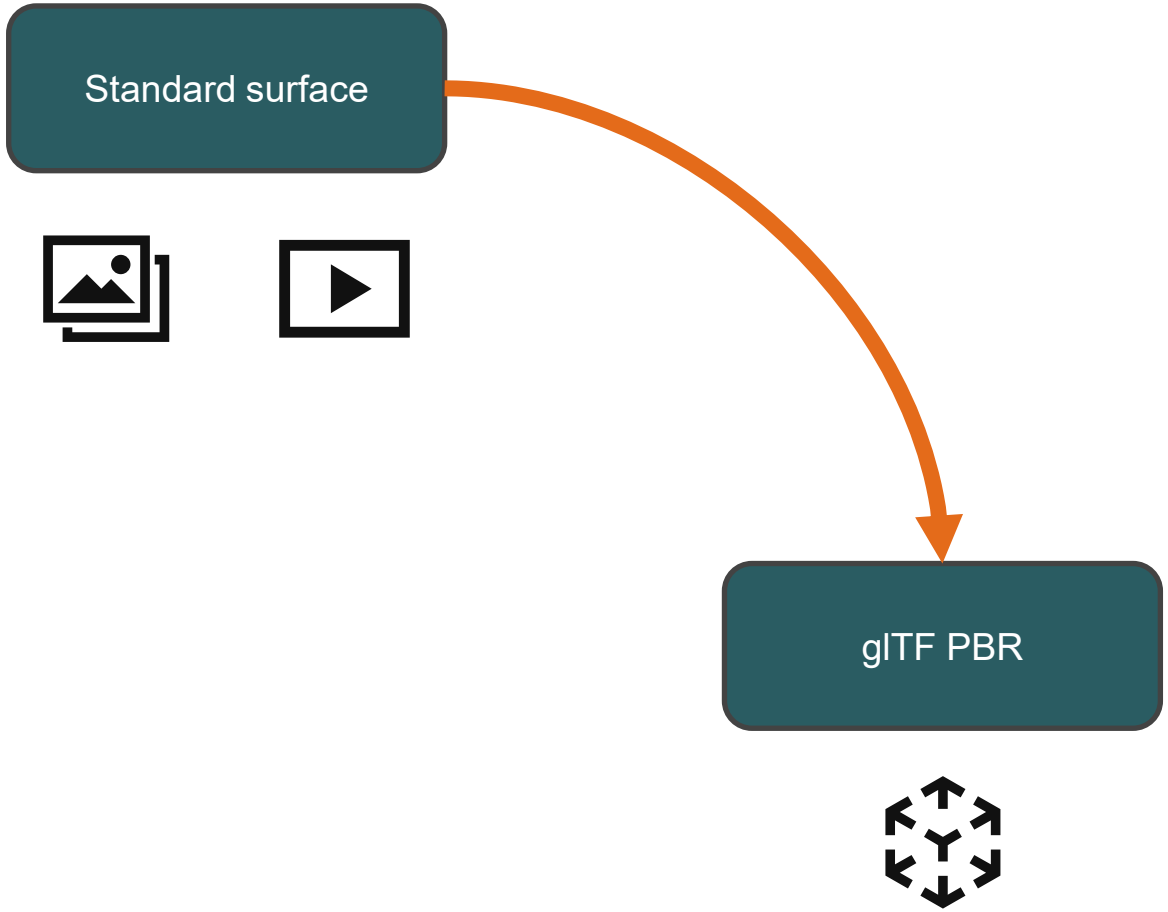
glTF PBR

Reduce work using translation graphs

Creating a material



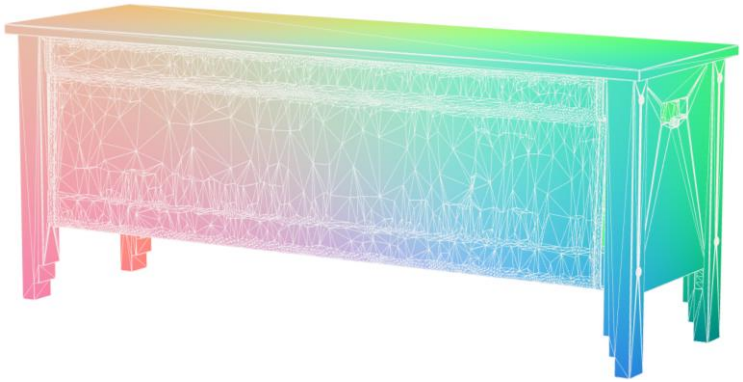
Generate Real-time Materials



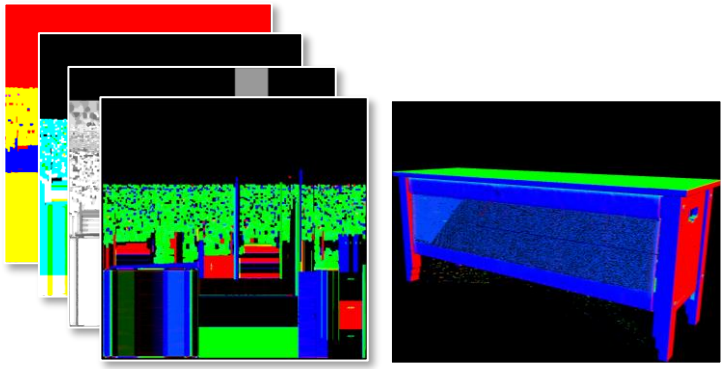
To glTF

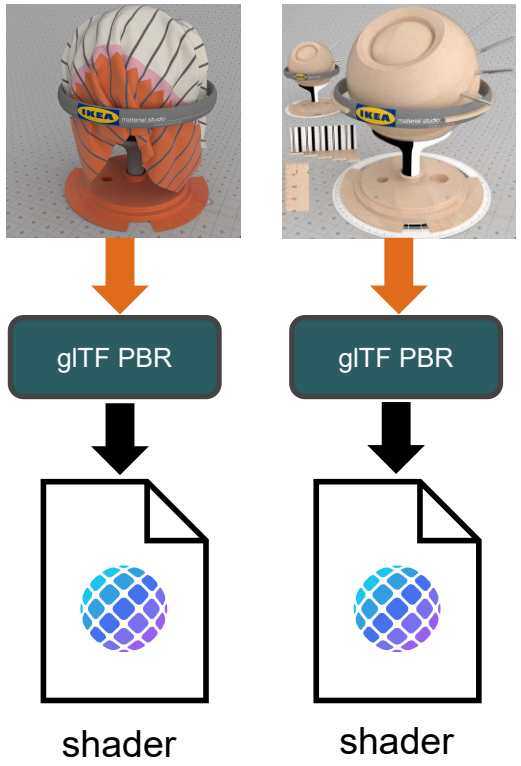
Bake to model

Geometry



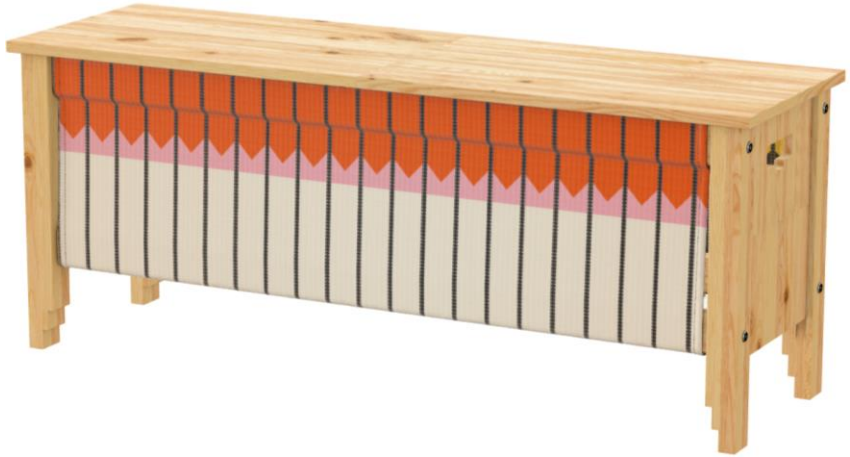
Mesh attribute textures



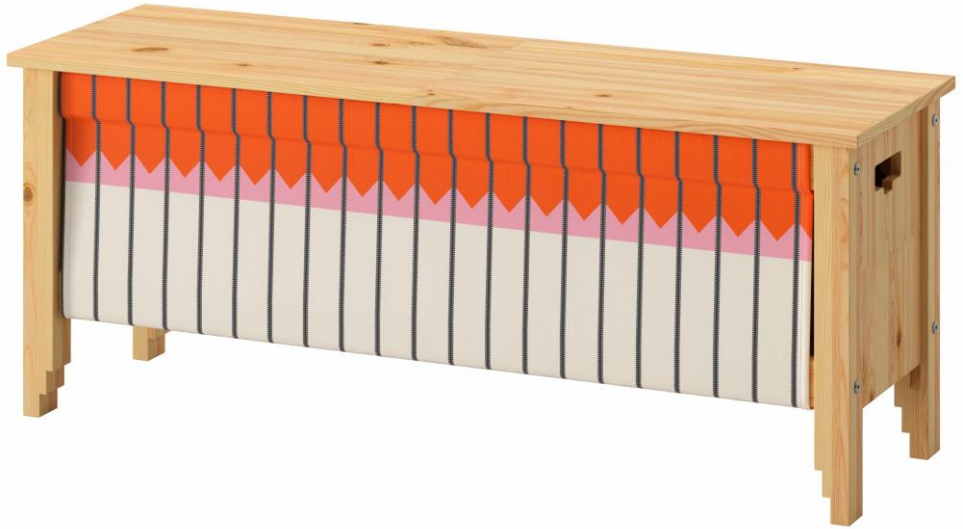


Packed glTF textures

Final glTF model



Offline render vs Real time model



Article offline rendered using standard surface materials

Article in Real-time with baked gltf_pbr material

glTF Extensions

Clearcoat

2020

Image: Zero roughness, half roughness, and half roughness plus clearcoat

Sheen

2020

Image: Before, and after adding KHR_materials_sheen

Specular

2021

Image: Before, and after adding KHR_materials_specular

Anisotropy

2023

Image: Before, and after adding KHR_materials_anisotropy

Transmission

2020

Image: Alpha blending, versus KHR_materials_transmission

Volume

2021

Image: Transmission alone, versus transmission with volume

Iridescence

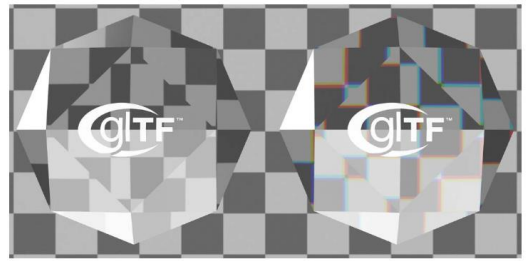
2022

Image: Before, and after adding KHR_materials_iridescence

Dispersion

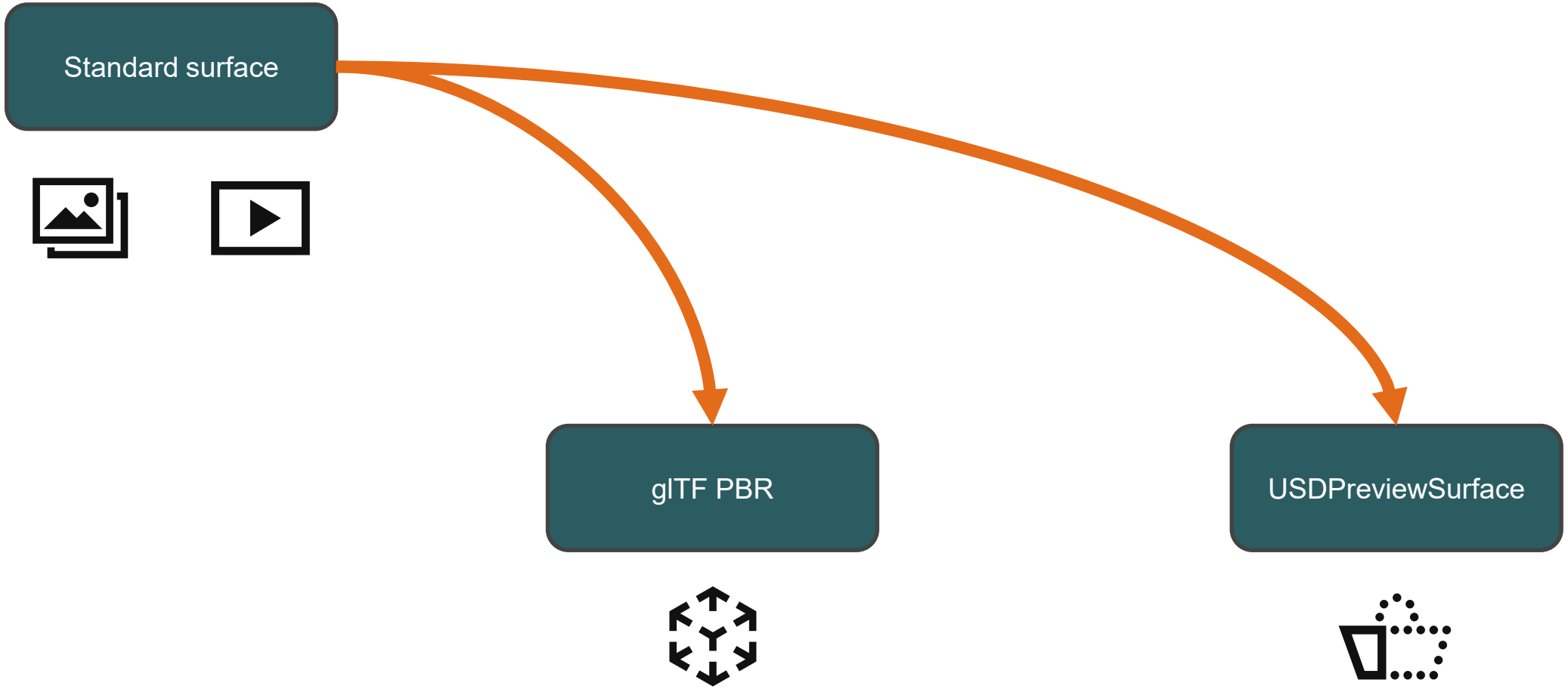
2024

Image: Before, and after adding KHR_materials_dispersion

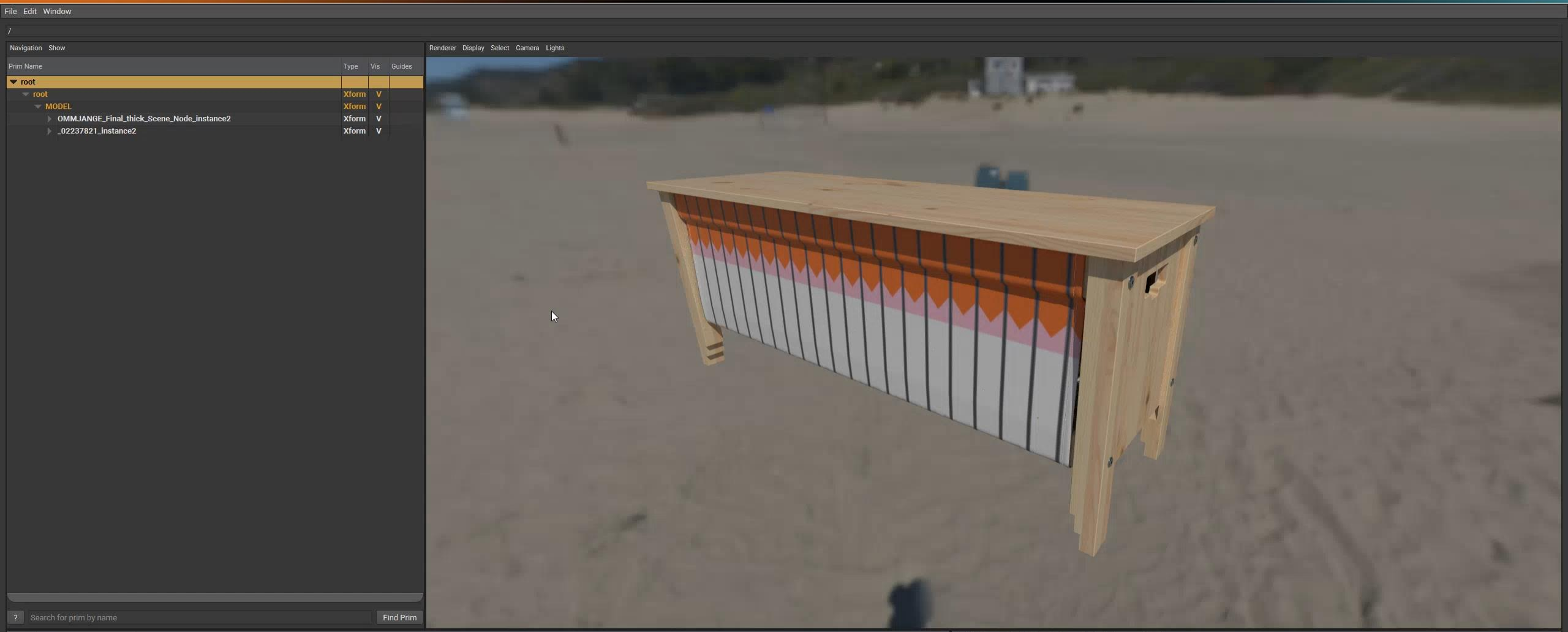


Speed up our workflow using translation graphs

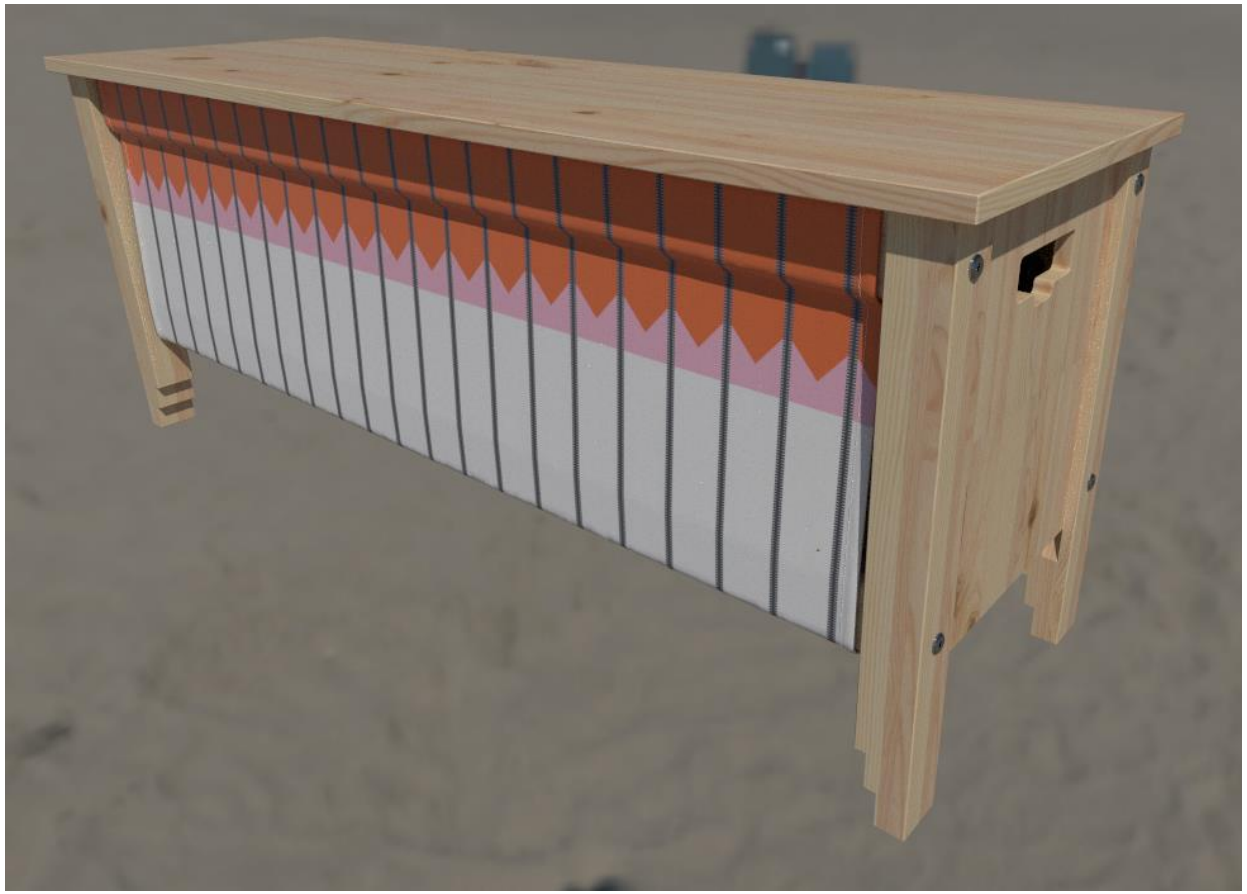
Proxy Materials



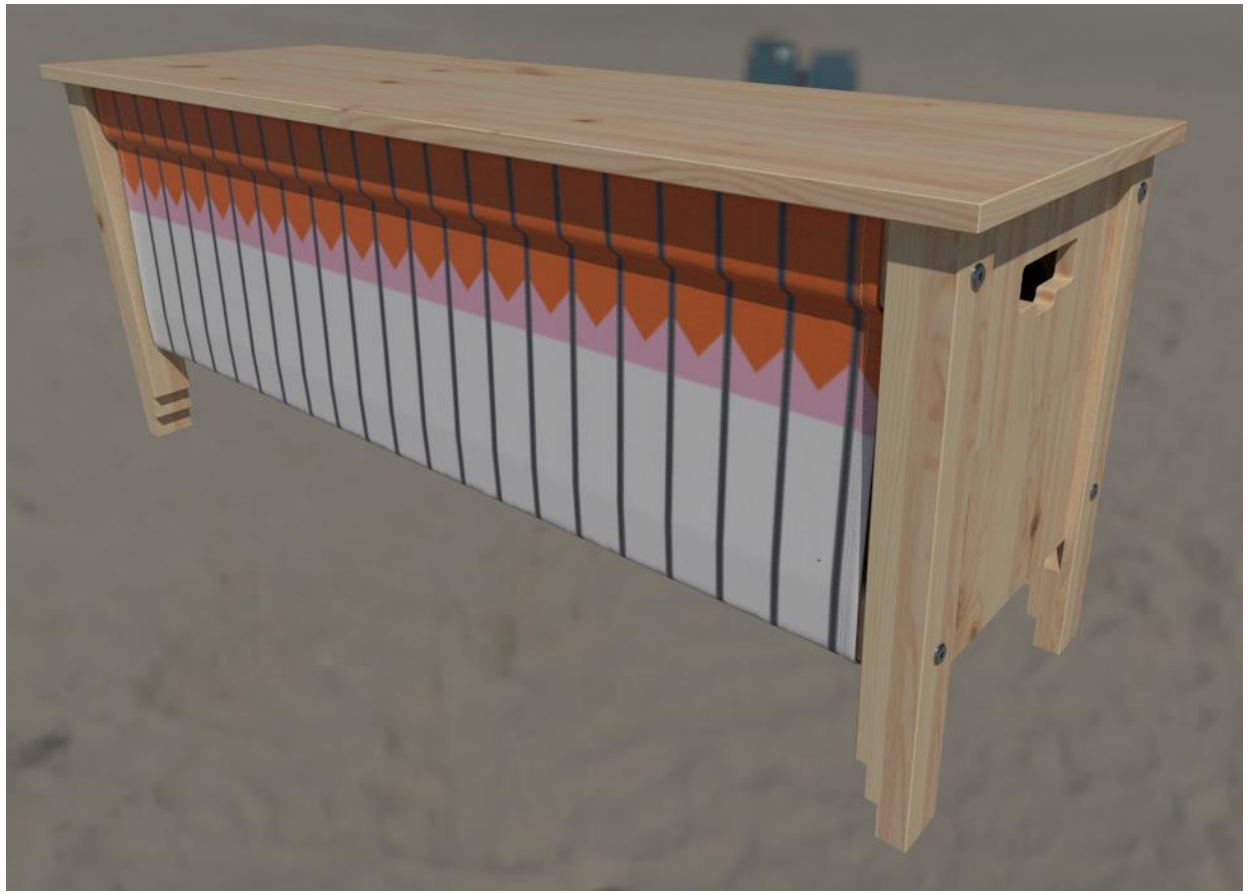
OpenUSD model with proxy materials



Full material vs Proxy material



MaterialX - Standard surface materials



USDPreviewsurface

Load time benchmark

USDPreviewSurface	MaterialX (.tx)	MaterialX (.png)
		
10.2 s	13.5 s	42.1 s

Summary

We ♡ Translation graphs

open source days '25

MATERIALX

openpbr

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

MaterialX and OpenPBR in OctaneRender® 2026

Arnon Marcus, OTOY

octanerender

otoy

#ASWF

About OctaneRender®

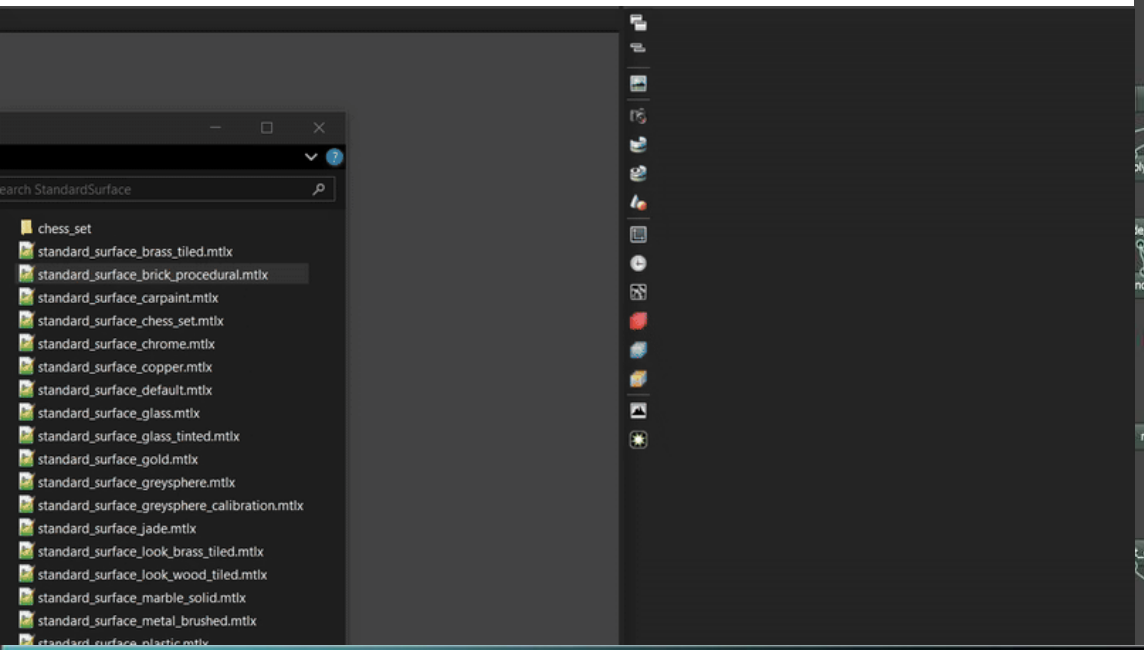
octanerender™

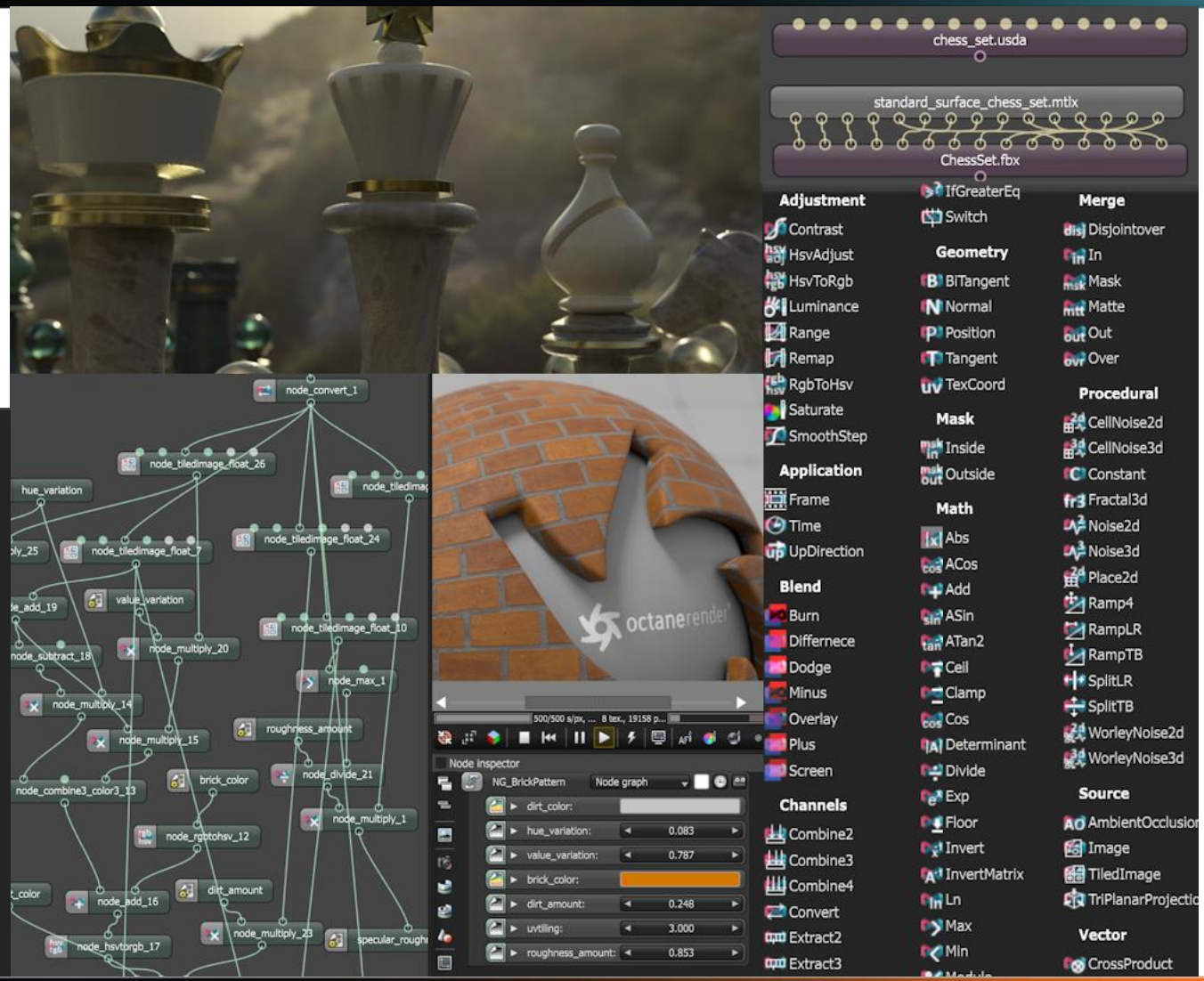
Spectral Production Renderer
Runs on the GPU
Broad industry support
Plugins for many DCCs
Has a Standalone Application

MaterialX in OctaneRender® 2026.1

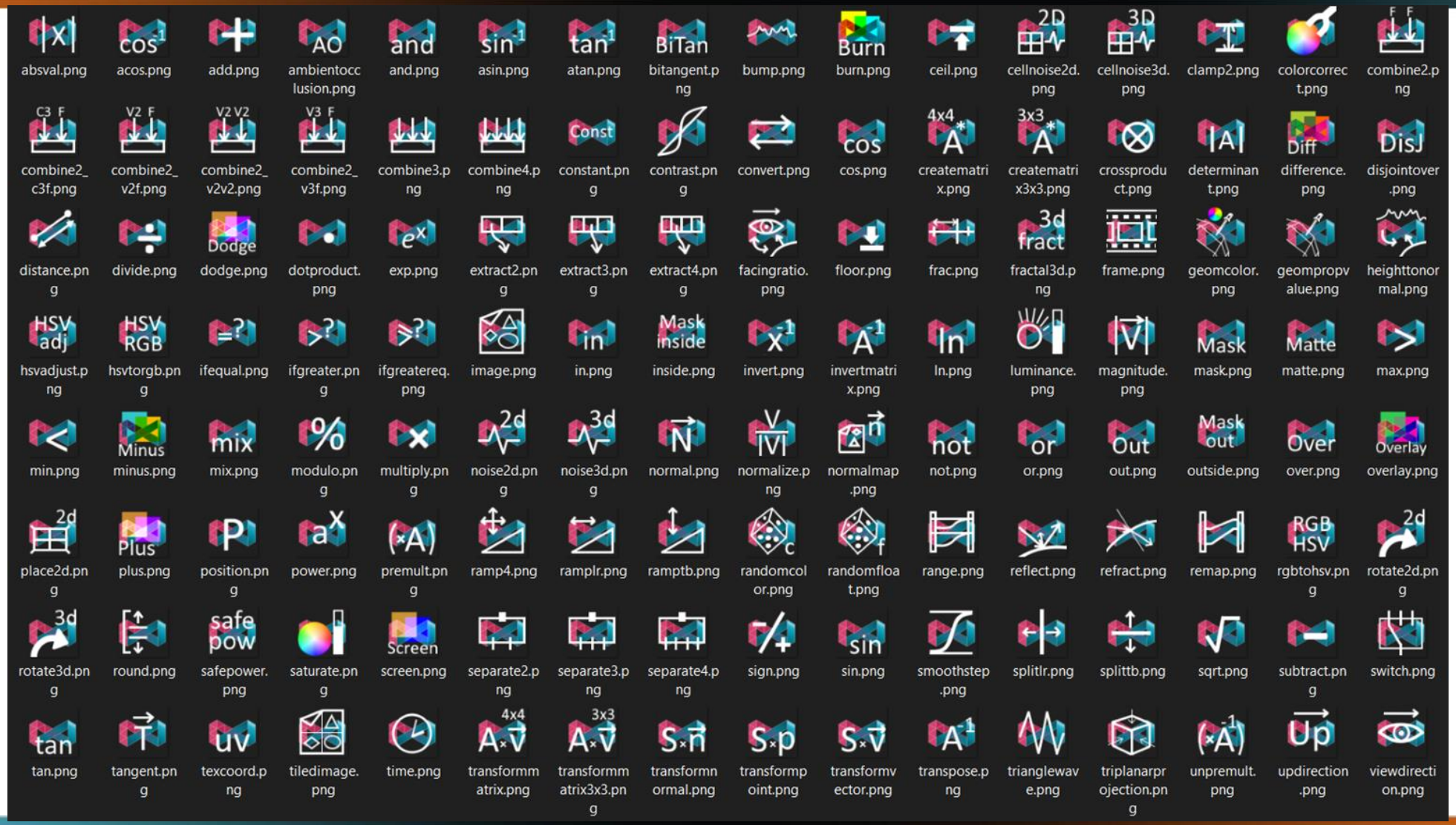
Native Support for MaterialX 1.39.2

MaterialX Native Nodes
Importable from .mtlx
Importable from .usd/c/a



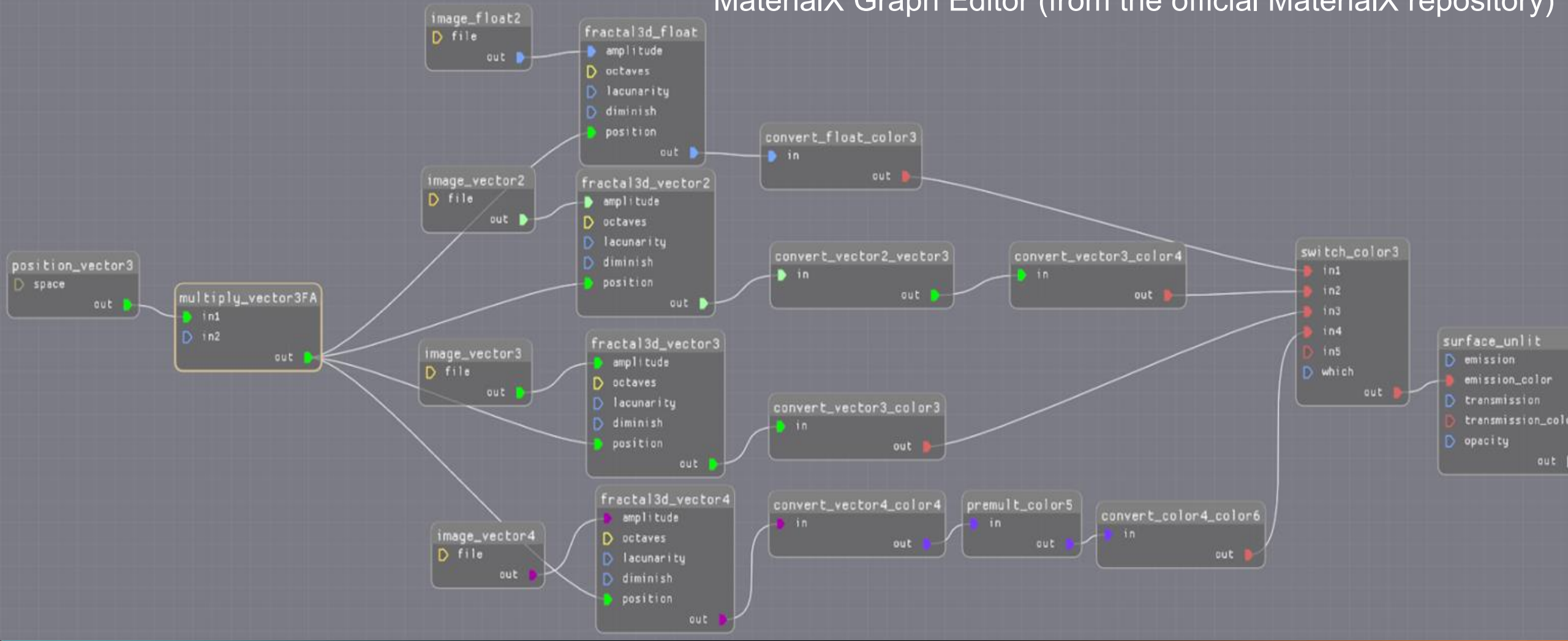


MaterialX Native Nodes (~130)



Typed Shading Nodes (MaterialX)

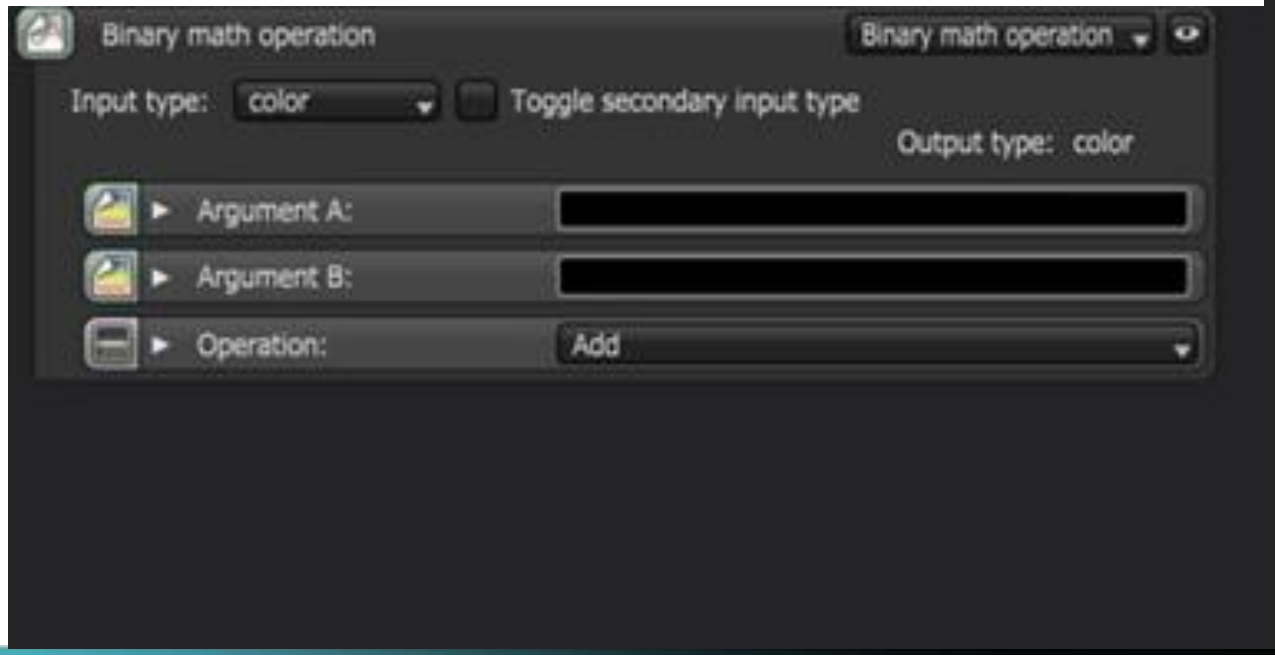
MaterialX Graph Editor (from the official MaterialX repository)

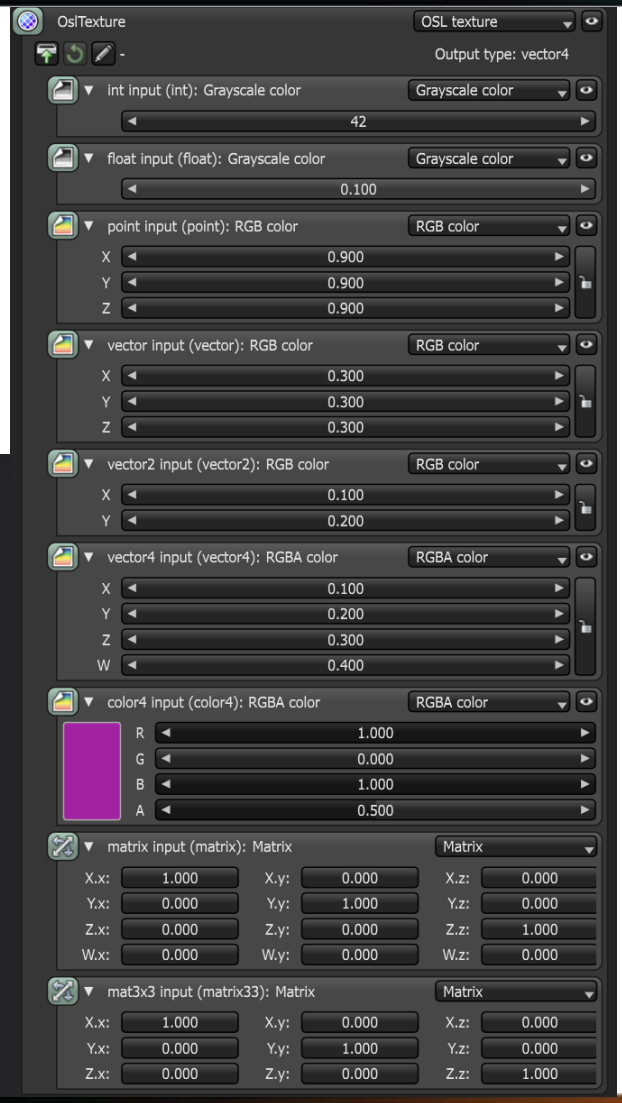


Typed Shading Nodes (OctaneRender®)

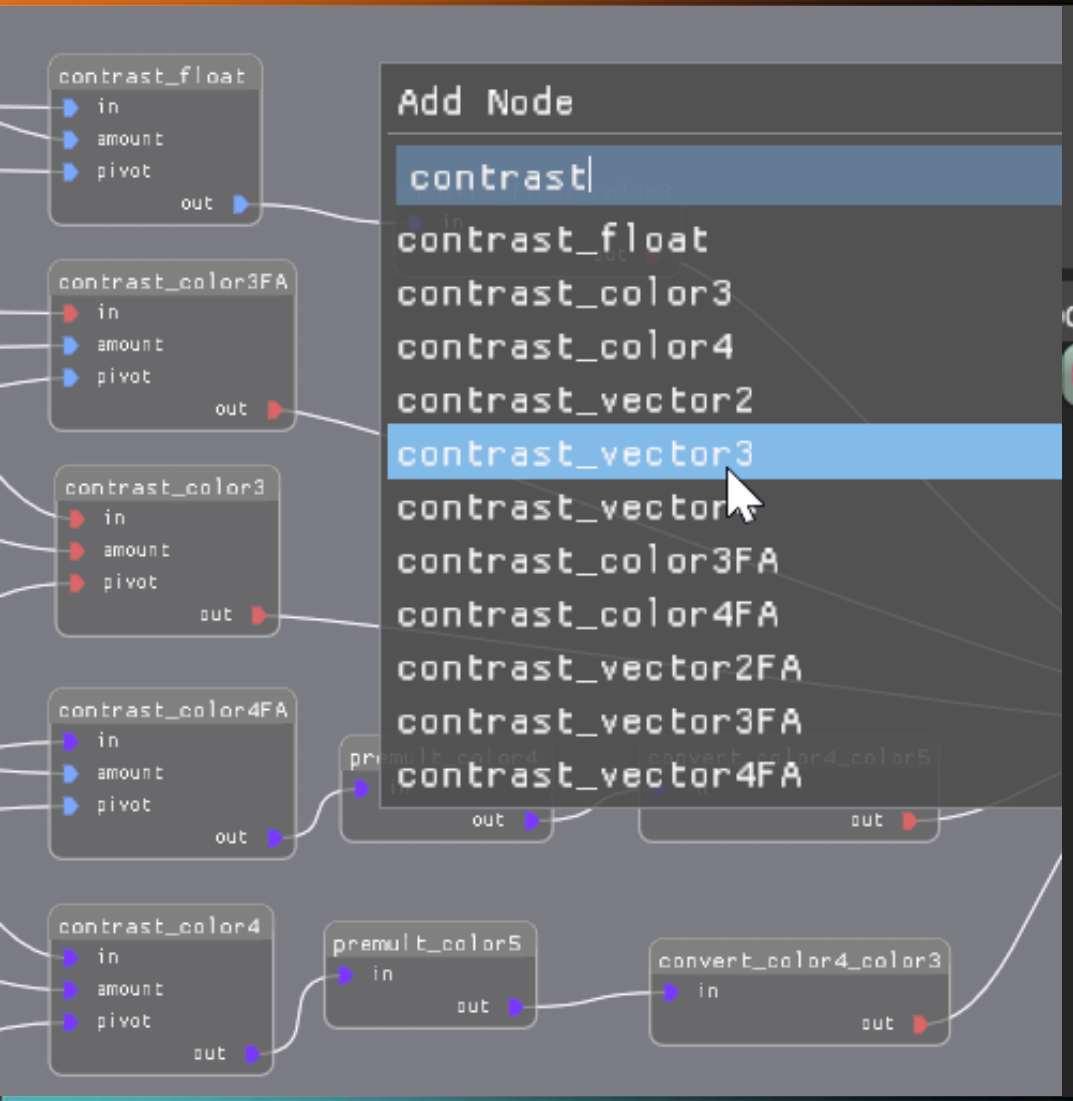
Shading nodes now support all types:

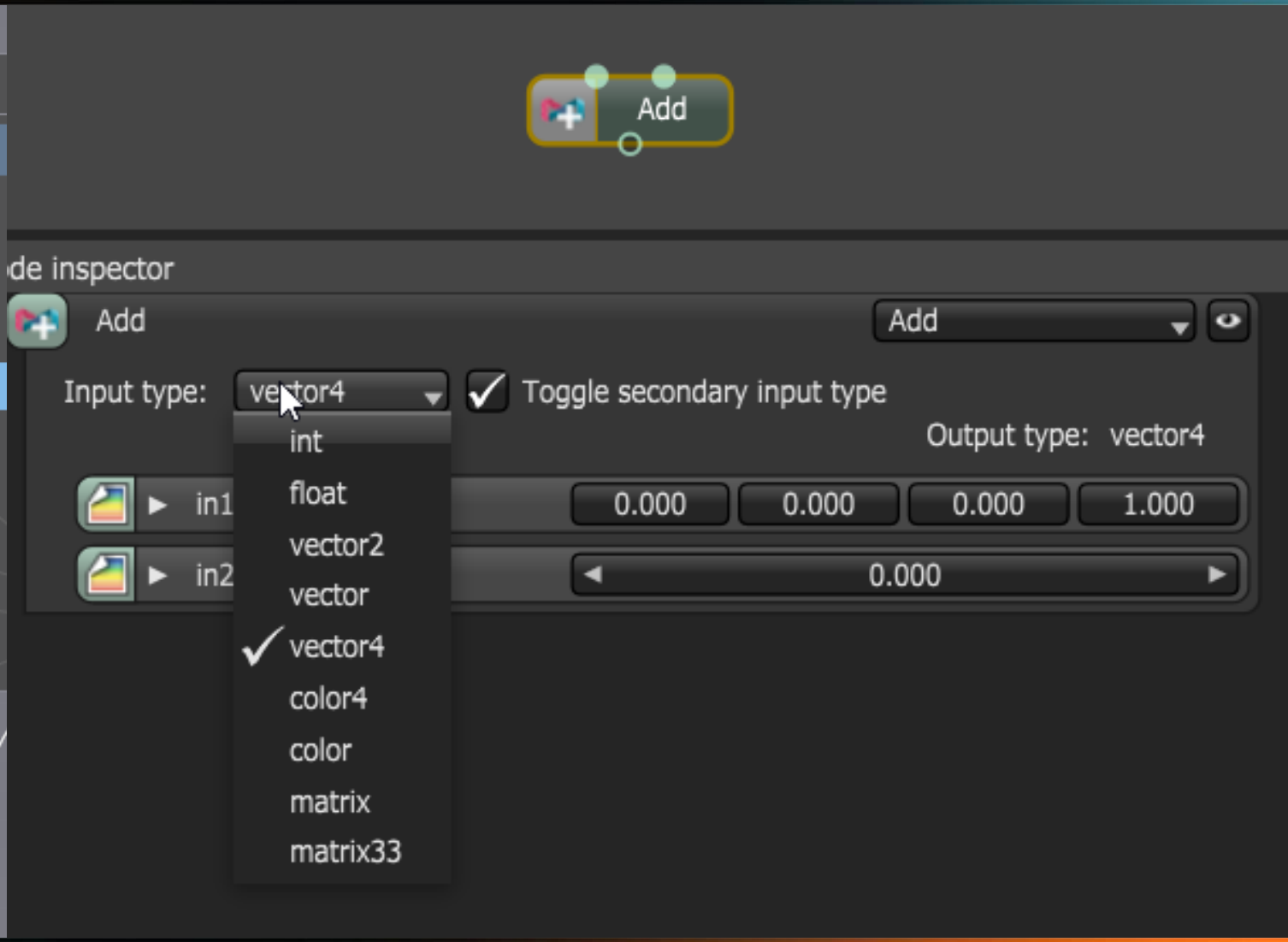
- Scalar: Integer, float, boolean
- Vector: vector2/3/4, color3/4
- Matrix: matrix33, matrix44





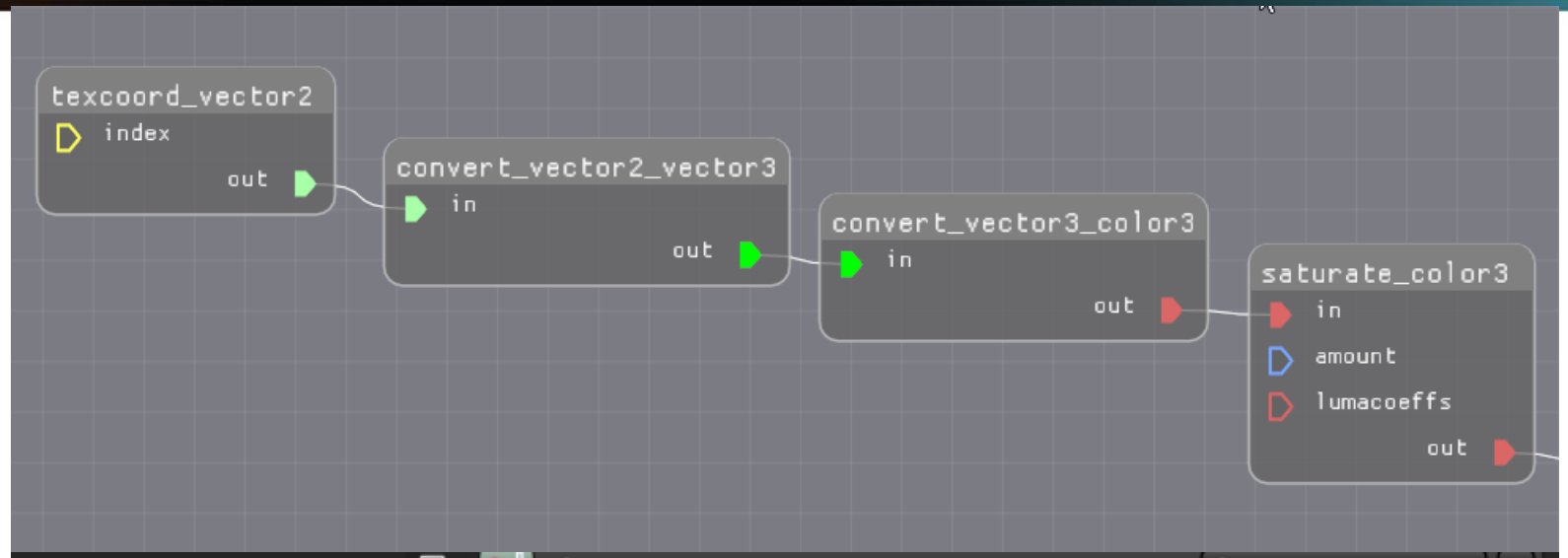
MaterialX Node Type Consolidations



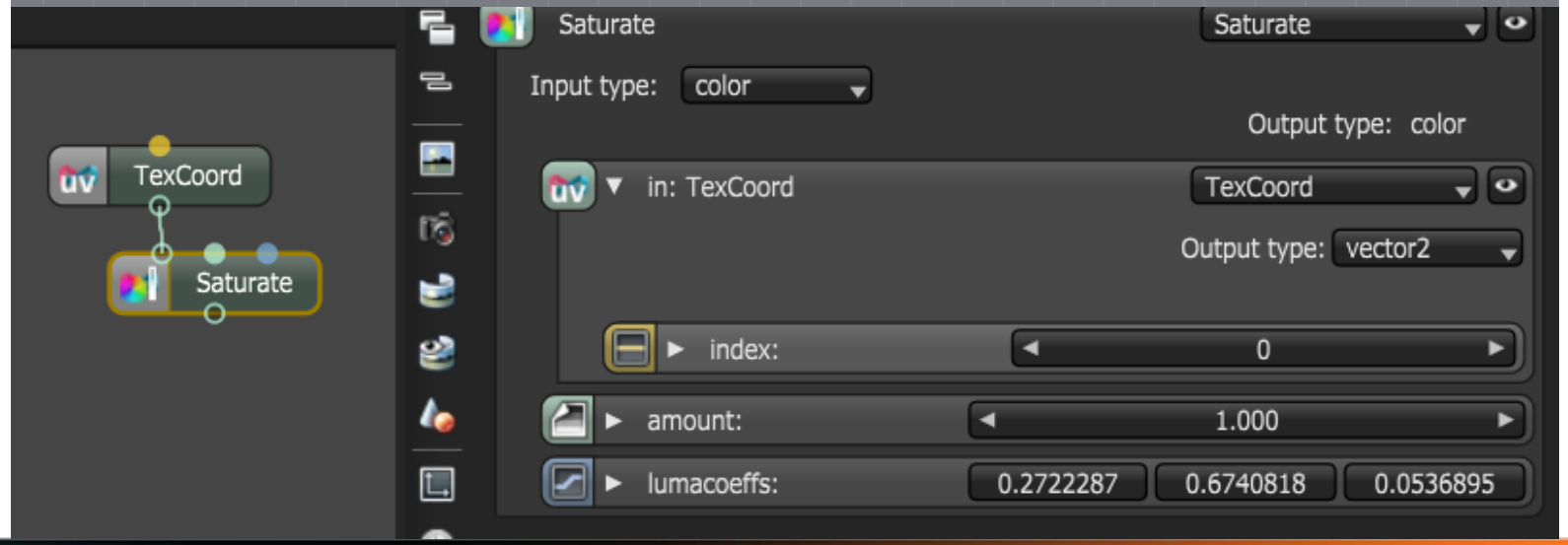


Implicit Type Conversions

Explicit Type conversion:

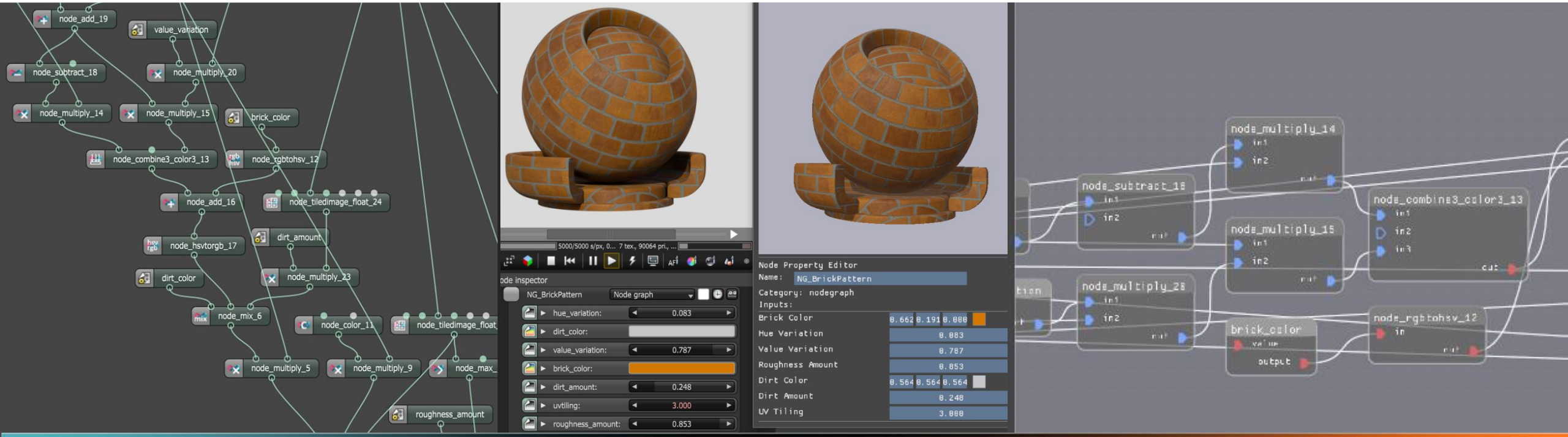


Implicit Type Conversion:



MaterialX Importing (.mtlx)

- Custom nodes defined as node-graphs at arbitrary nesting levels
- Tracking of color-spaces of color inputs and image nodes
- Relative file paths and UDIM patterns for image/tiled-image nodes

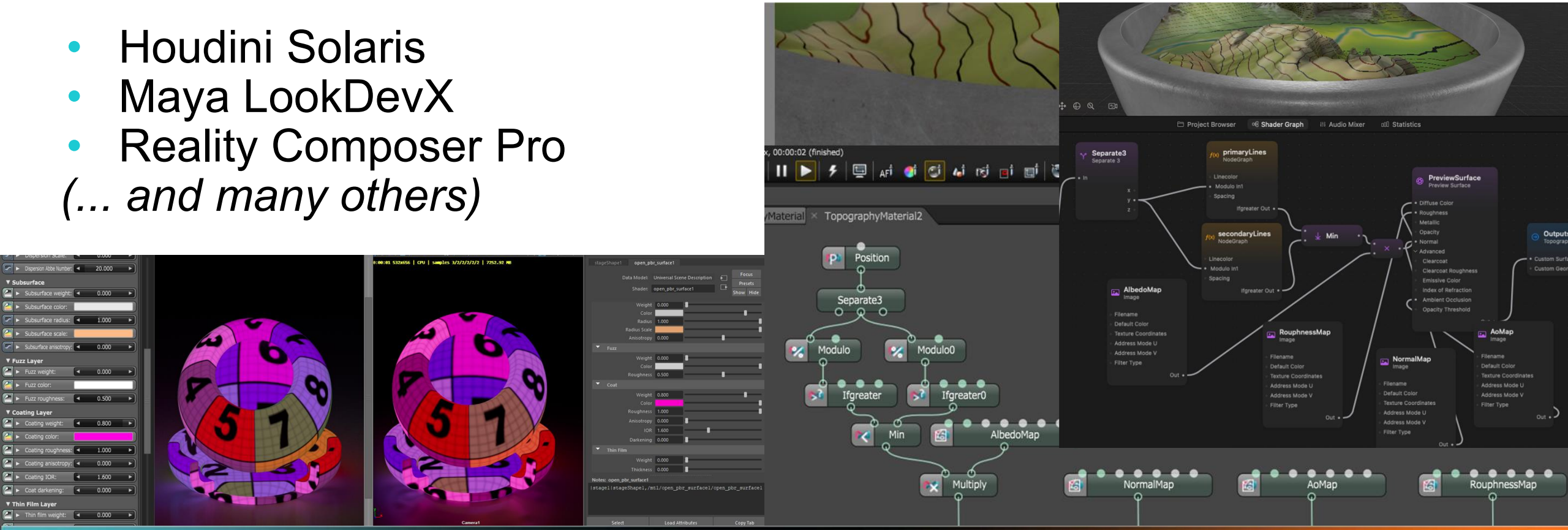


MaterialX Importing (.usd/c/a/z)

USD Importer supports MaterialX in USD from many sources

- Houdini Solaris
- Maya LookDevX
- Reality Composer Pro

(... and many others)

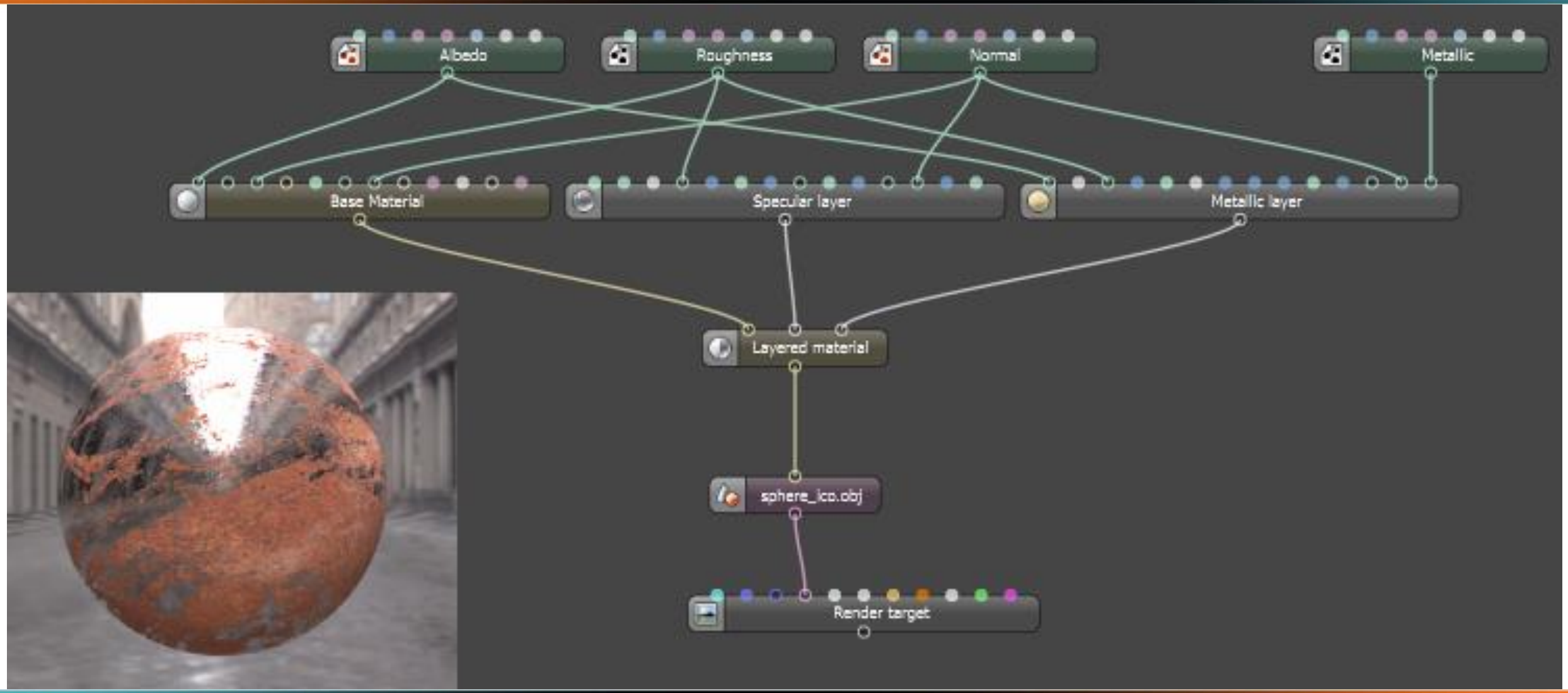


OpenPBR in OctaneRender® 2026.1

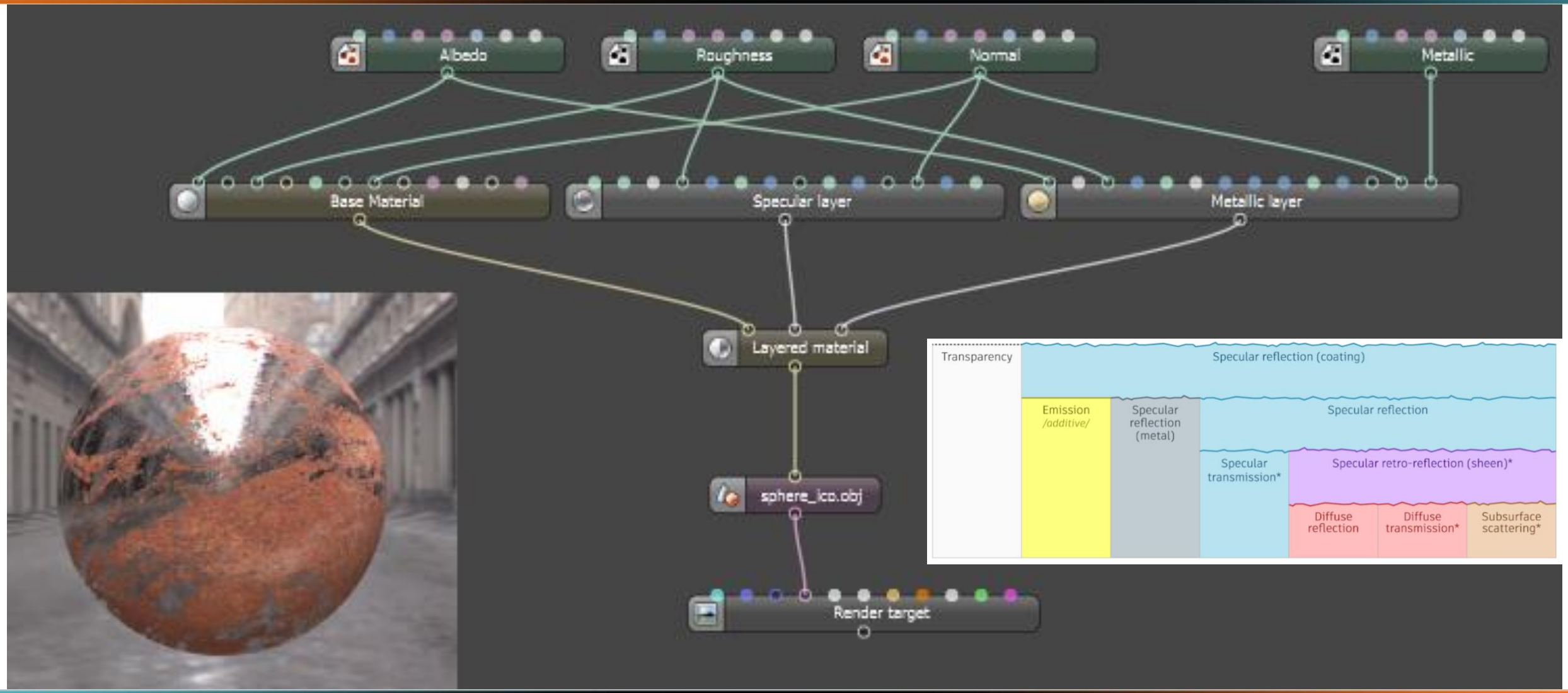
Material Node: OpenPBR Surface (1.1)

- Material Layering & Mixing
- Physically Spectral Effects
- Probabilistic Sampling
- Importable from MaterialX
- Importable from USD

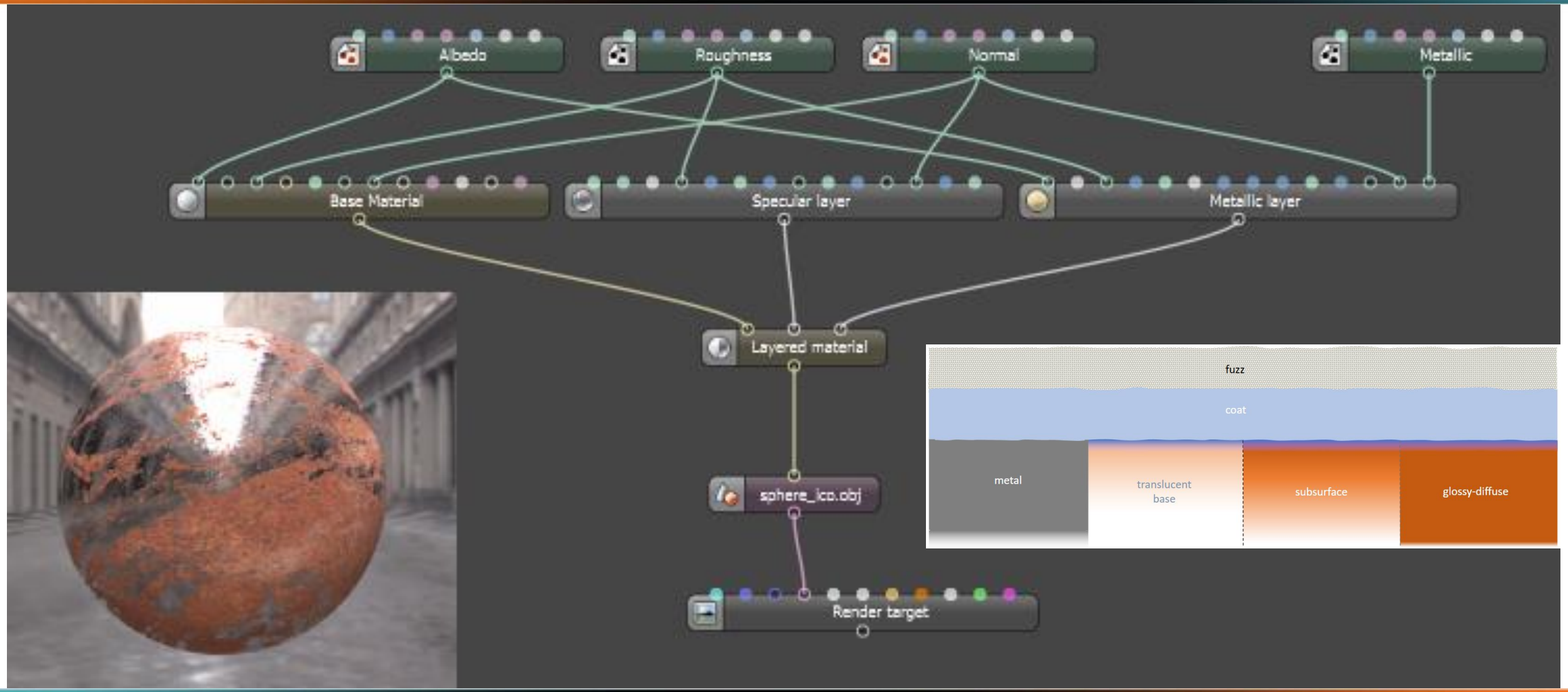
Material Layering & Mixing



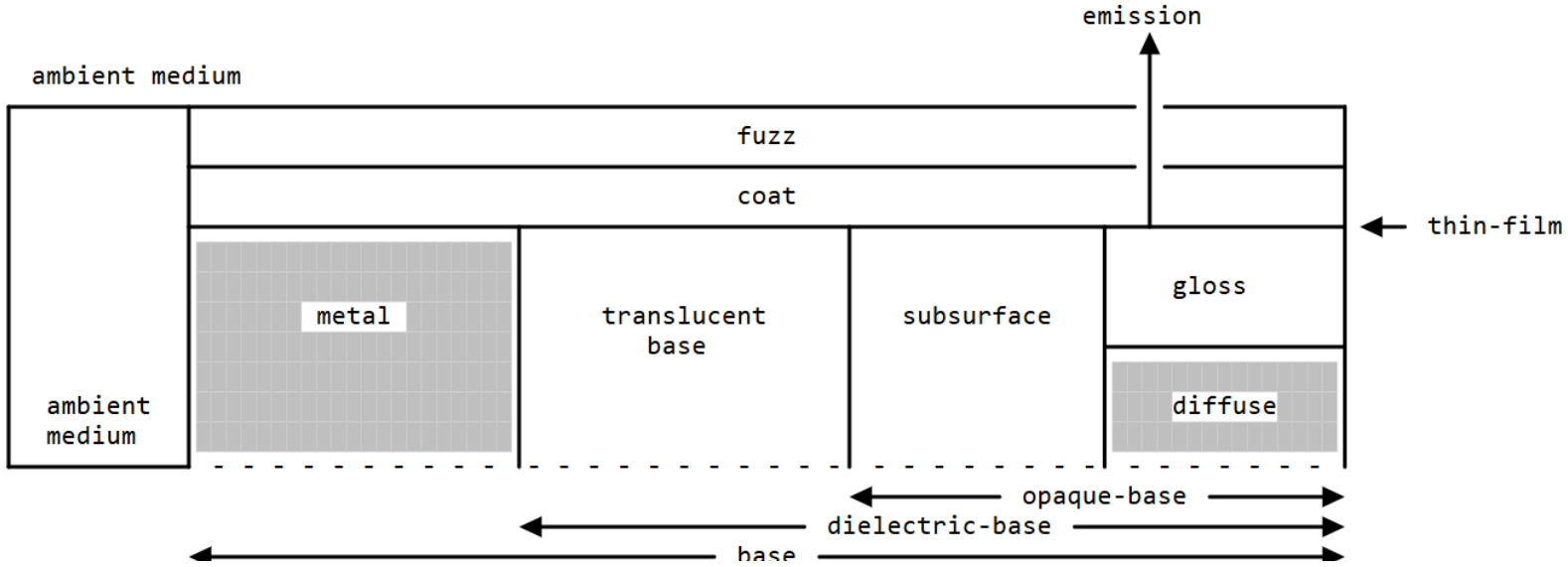
Material Layering & Mixing

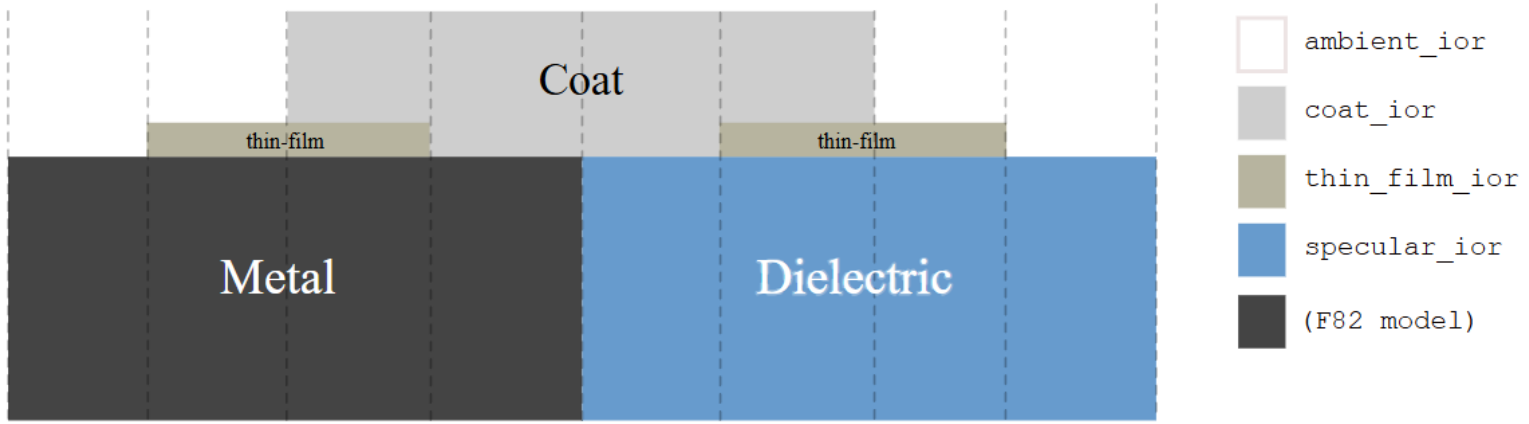


Material Layering & Mixing



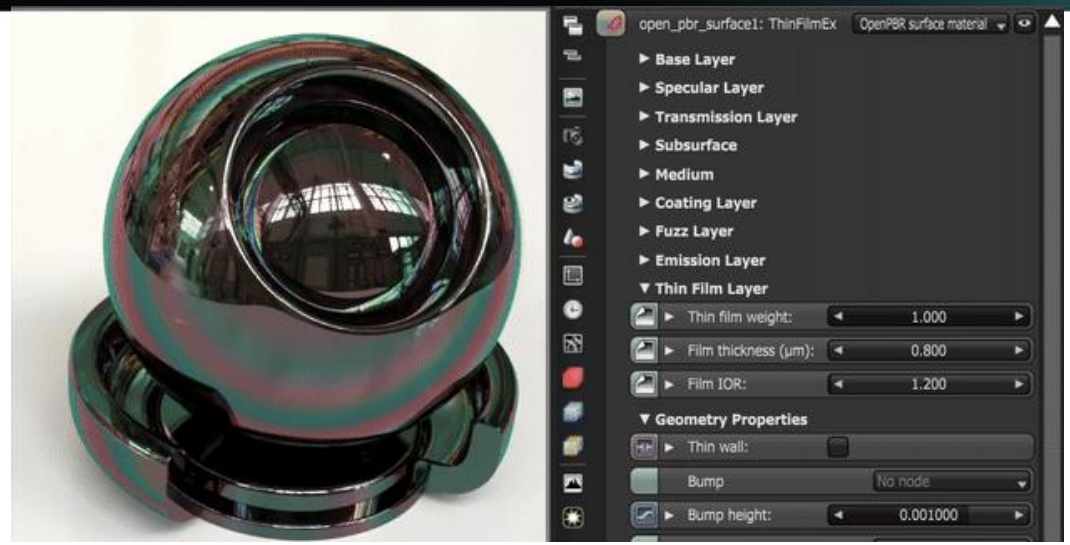
Physical Structure & Implementation



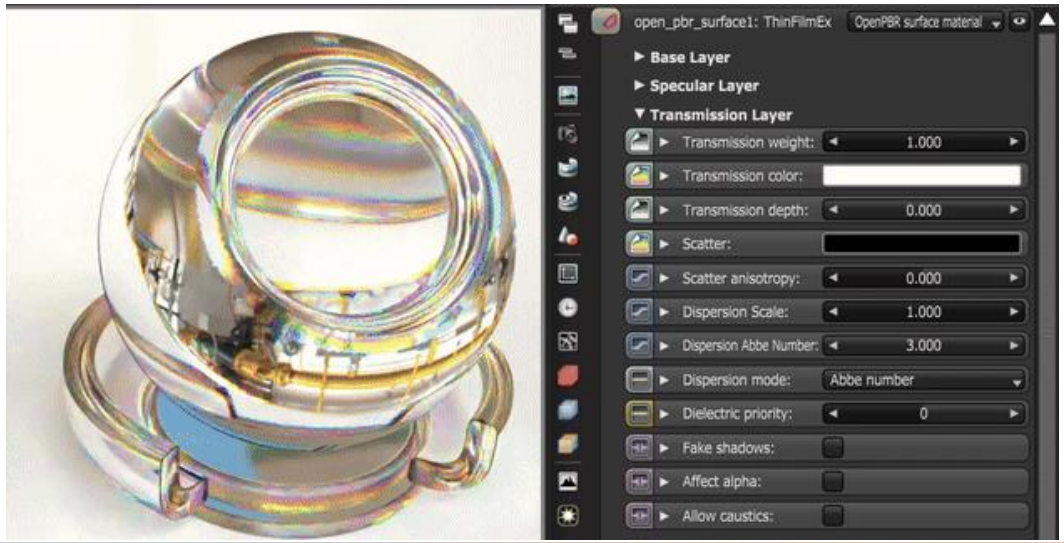


Spectral Effects

Thin Film Weight



Dispersion Scale



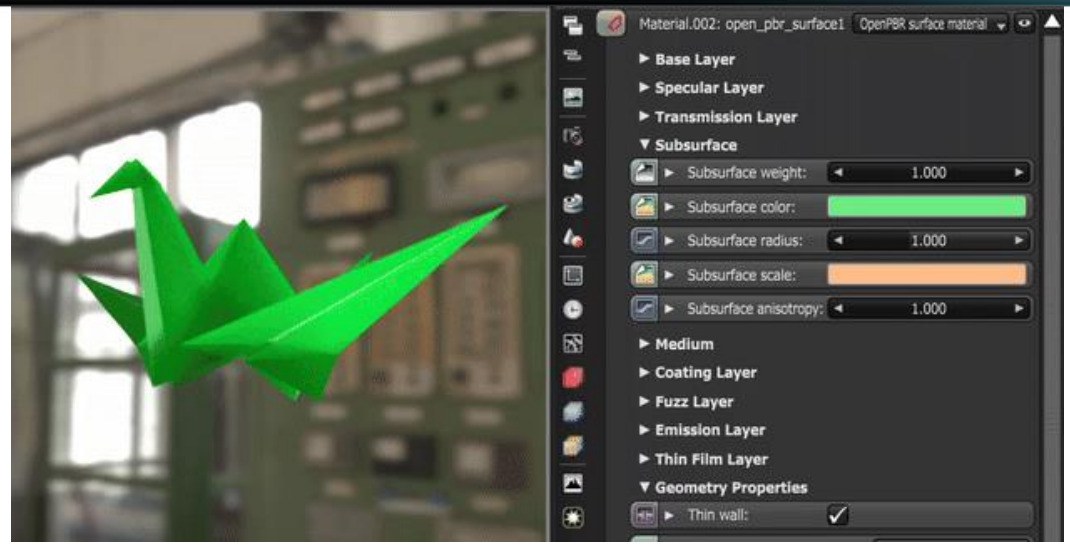
Probabilistic Sampling

Thin Walled SSS:

Diffuse Transmission

vs.

Diffuse Reflection



Emission through Coat:

Tinted Emission Color

vs.

Untinted Emission Color

JuanG3D

CC Attribution

Future Work

OpenPBR (finalization for 2026.1):

- Oren Nayar -> Energy conserving (EON)
- Sheen -> Fuzz (microflake)

MaterialX:

- Gradient Ramp
- Hex Tiling
- PBR: BSDF nodes*

* Subject to changes in the roadmap

Credits

Arnon Marcus: MaterialX, OpenPBR, Shading System Typing

Roeland Schoukens: Shading System Types (back-end compiler)

Wallace Yuen: Standard Surface (which OpenPBR is derived from)

Vijay Thirukonda: USD Importer

open Source days '25

/* ACADEMY SOFTWARE FOUNDATION

Virtual Town Hall Series

MaterialX and OpenPBR in NVIDIA Omniverse

#ASWF

Hextiling

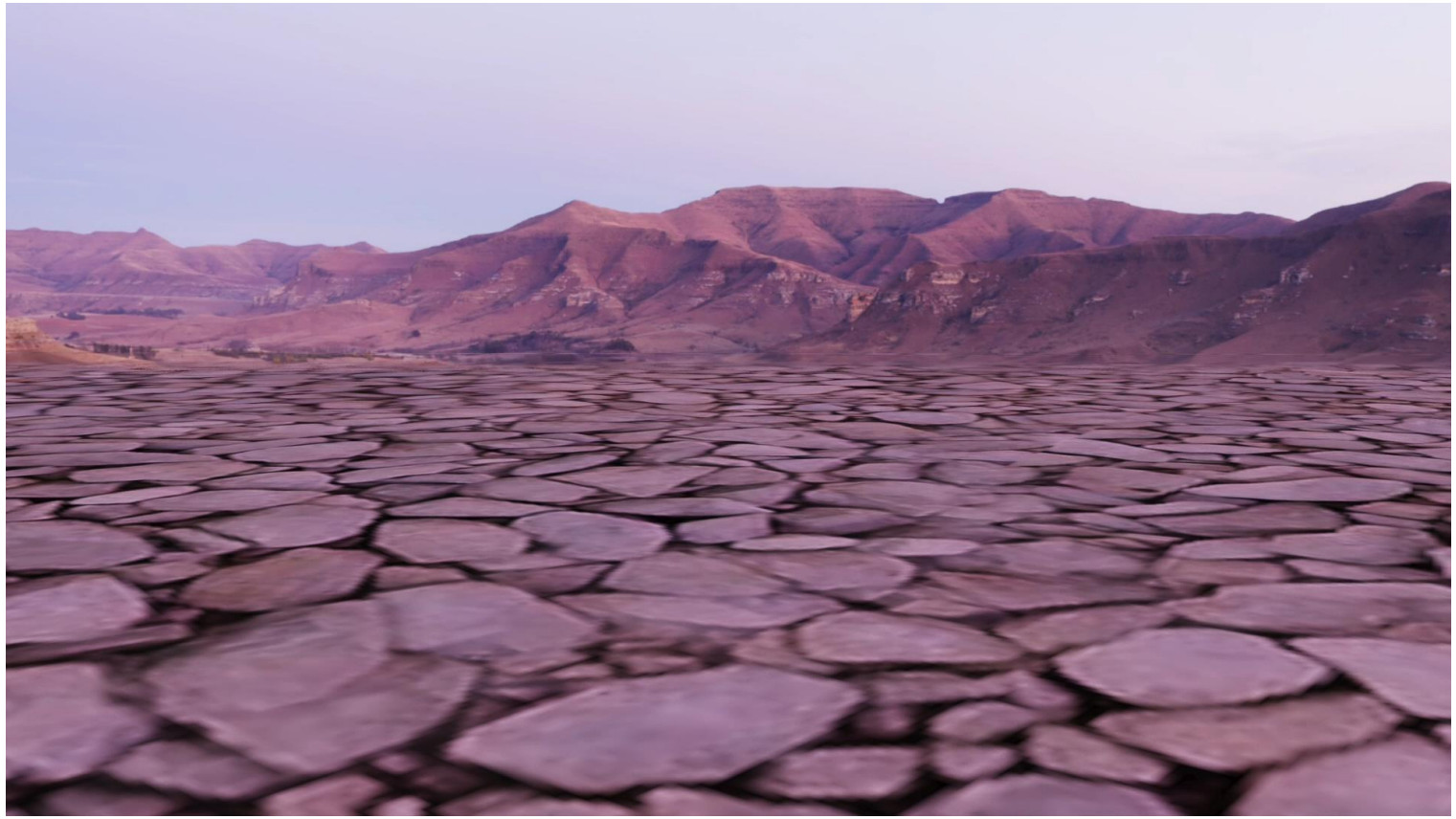
HEXTILING NODES IN MATERIALX 1.39.4

- Introduces 2 new nodes to enable texture tiling without visible repetitive patterns.
- Based on “Practical Real-Time Hex-Tiling” by Mikkelsen et. al 2022.
- Designed for Real-time graphics

Hextiling

Uniform grids with ND_tileimage

Hextiling



Random hexagonal grids with ND_hextileimage

Hextiling

HEXTILING NODES IN MATERIALX 1.39.4

- Introduces 2 new nodes to enable texture tiling without visible repetitive patterns.
- Based on “Practical Real-Time Hex-Tiling” by Mikkelsen et. al 2022.
- Designed for Real-time graphics

Hextiling

HEXTILING NODES IN MATERIALX 1.39.4

- <hextileimage>: for color textures
- <hextilenormalmap>: for normal maps

Hextiling

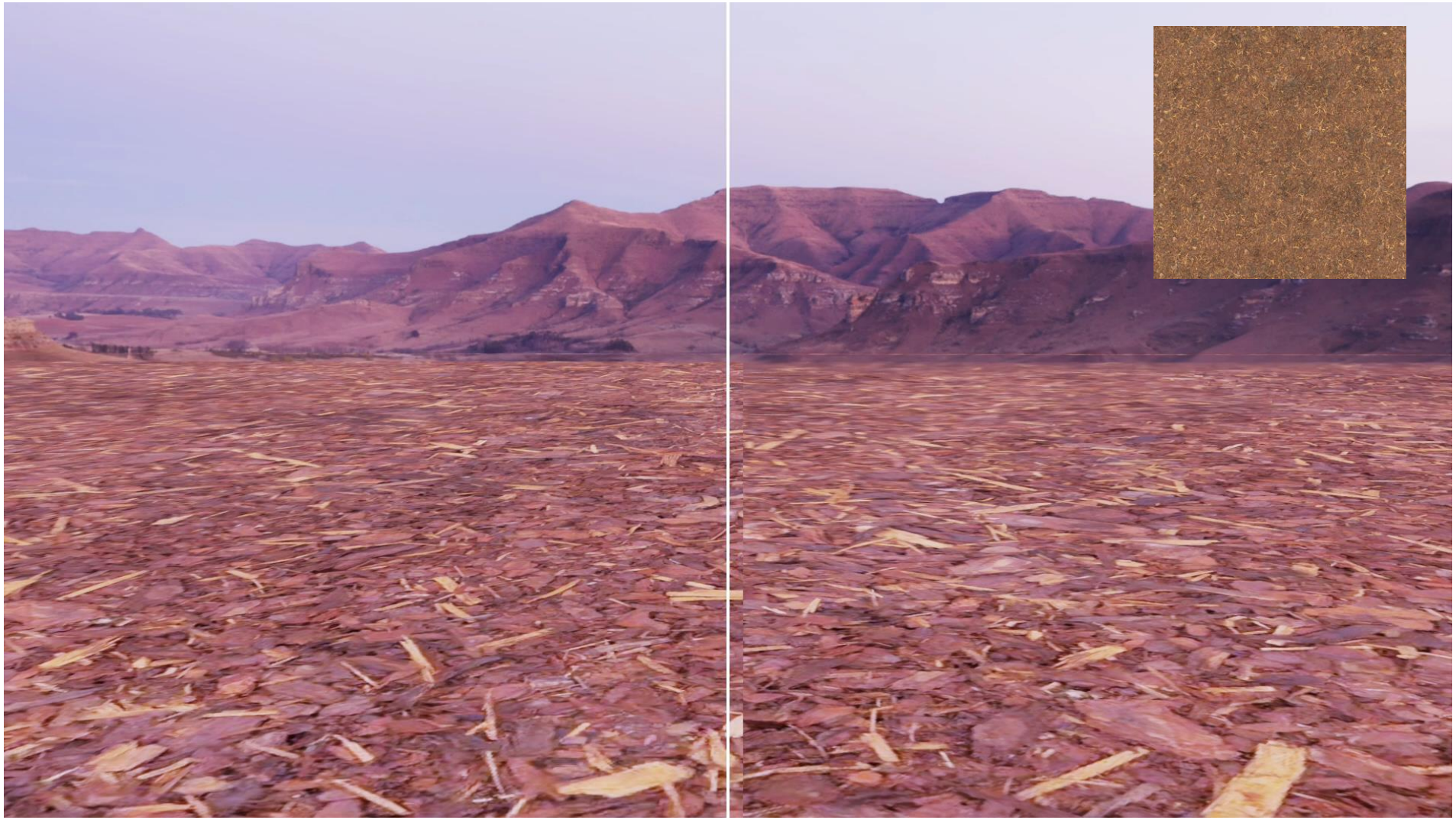
HEXTILING NODES IN MATERIALX 1.39.4

- Builds on a by-example noise algorithm, adapted to reduce implementation complexity
- Omits histogram preservation in favor of direct blending of source texture
- Uses a contrast function for color maps to blend weights to maintain visual detail
- Maintains surface derivatives for normal map blending

Hextiling

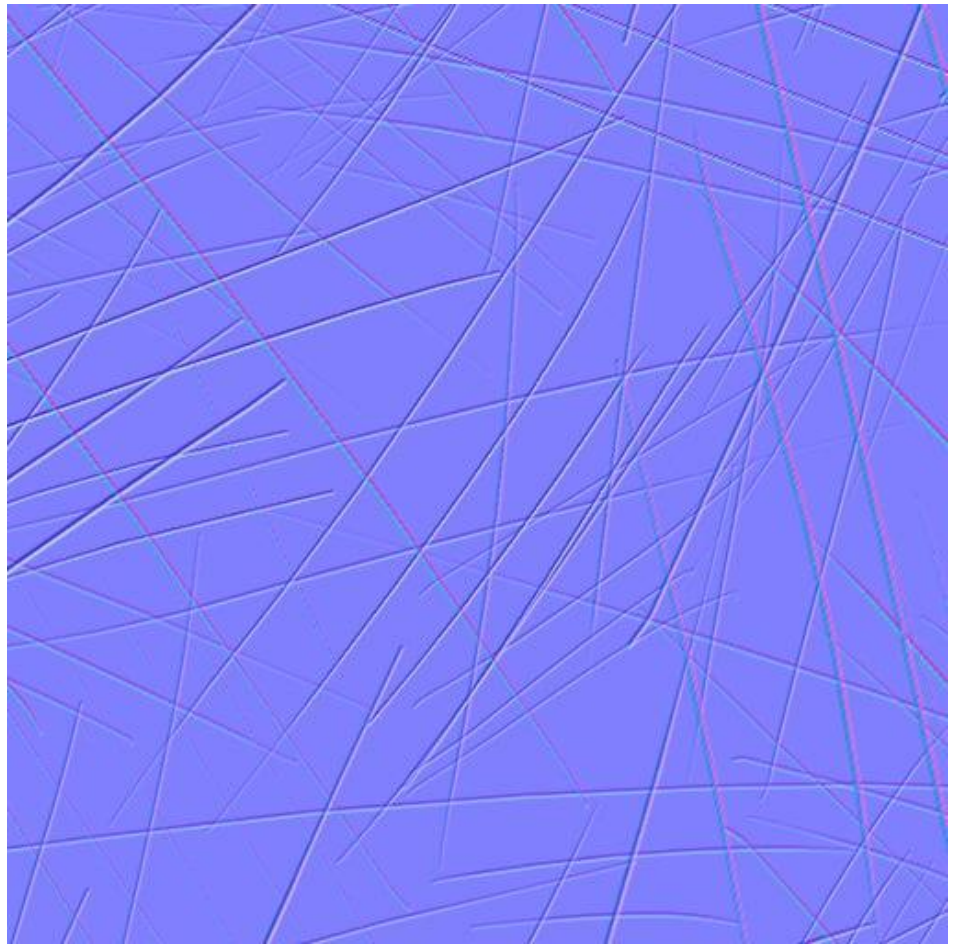
Parameter	Type
file	Filename
default	color3
texcoord	vector2
tiling	vector2
rotation	float
rotation range	Vector2
scale	float
scale range	Vector2
offset	float
offset range	Vector2
falloff	float
falloff contrast *[color only]*	float

Hextiling



Hextiling

Hextiling



PhysicalAI-SimReady-Materials

PhysicalAI-SimReady-Materials

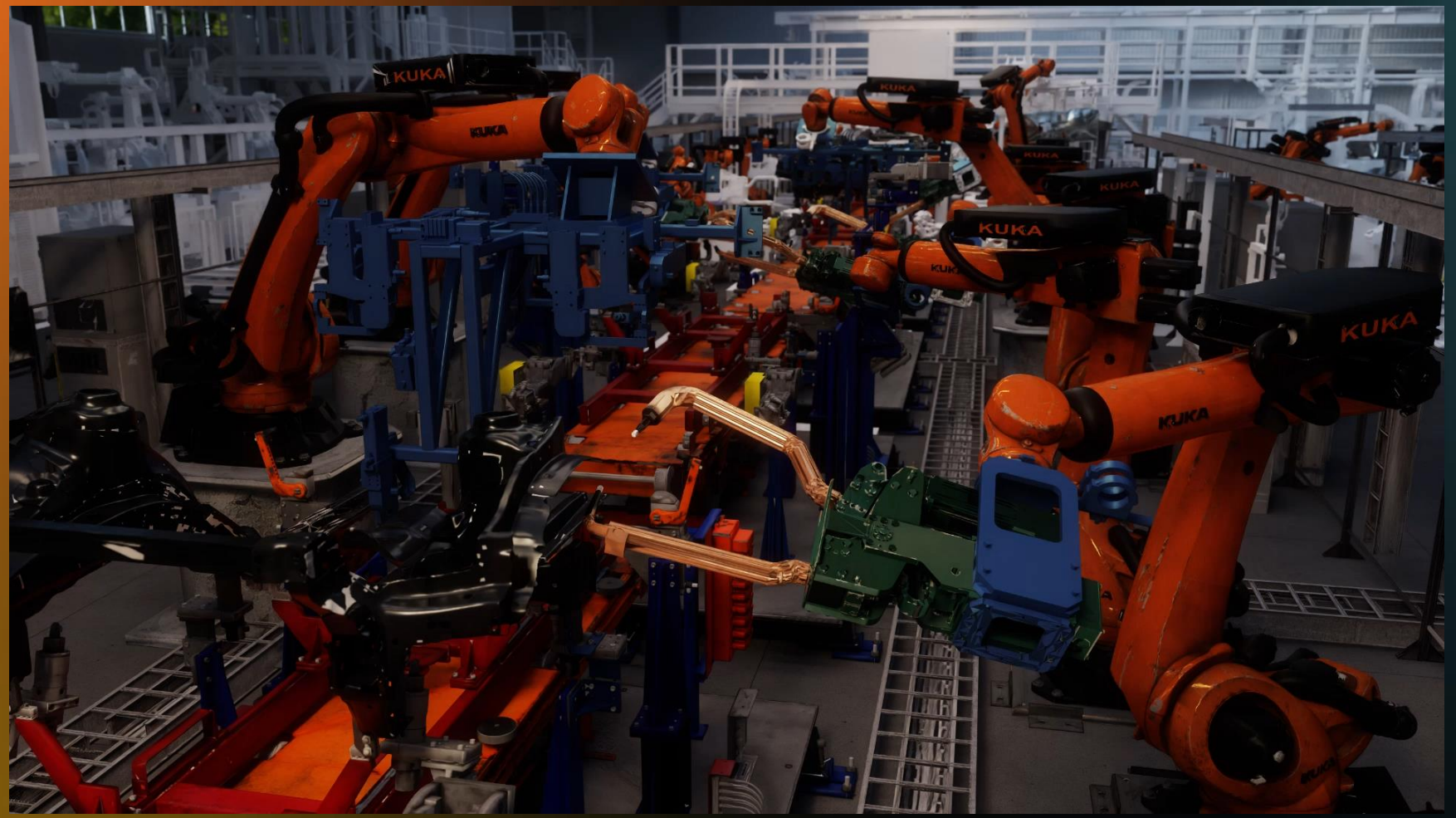
PhysicalAI-SimReady-Materials on Github

- MIT-0
- Combines MaterialX, OpenPBR and Non-Visual Sensor Attributes to enable high-quality physical simulation for visible and non-visible sensors.

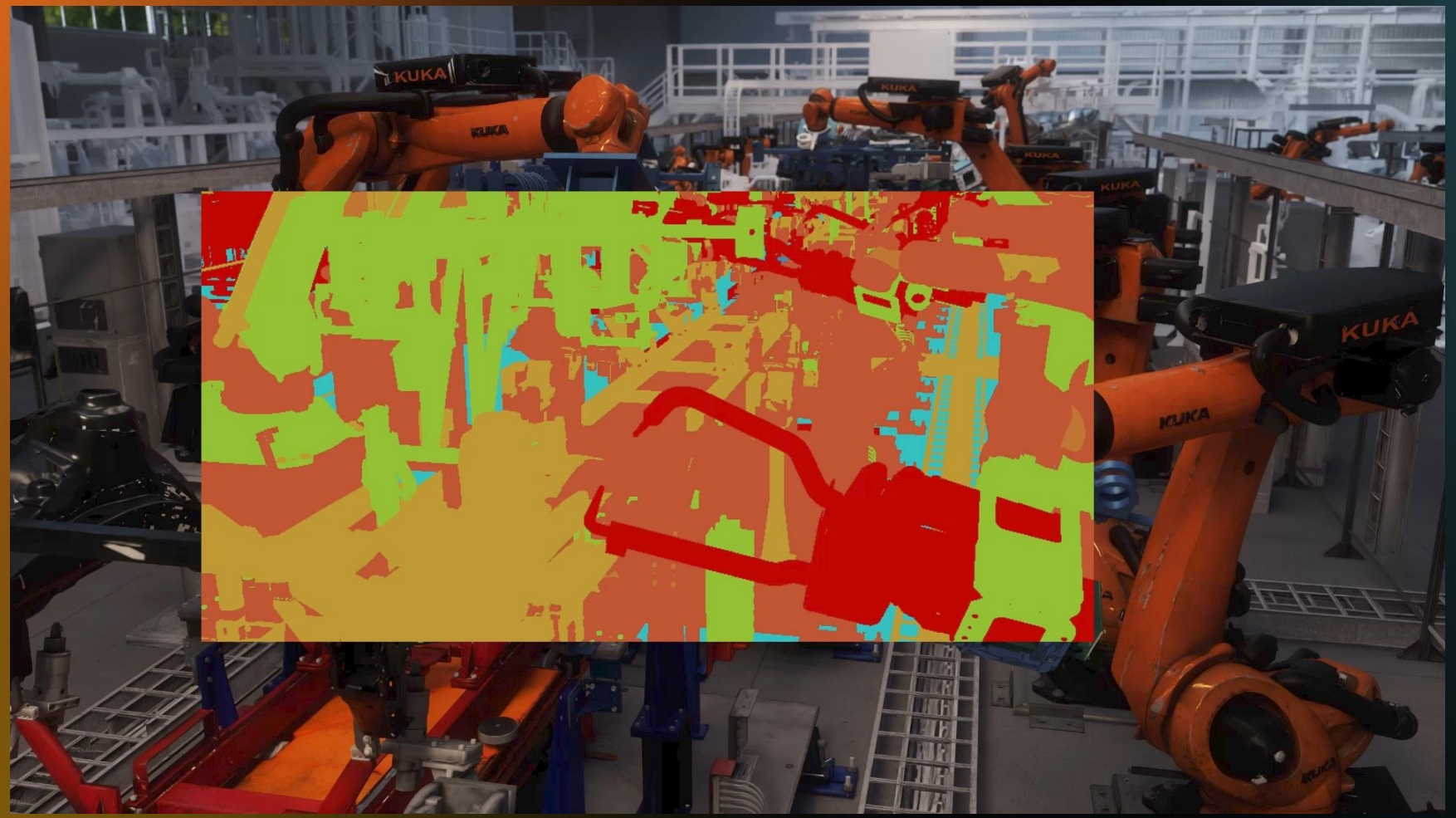
PhysicalAI-SimReady-Materials

NVIDIA Isaac Sim is a reference application to develop, train and test AI driven robots in physically based virtual environments.

NVIDIA Sensor RTX is a suite of microservices and APIs to enable physically accurate sensor simulation for things like autonomous vehicles, robots, manufacturing, and smart infrastructure.



open source days '25



▼ Debug View

Render Target: Non-Visual Material ID

Enable Pixel Debug

▼ Base Material Filter

- 0 None
- 1 Aluminum
- 2 Steel
- 3 Oxidized Steel
- 4 Iron
- 5 Oxidized Iron
- 6 Silver
- 7 Brass
- 8 Bronze
- 9 Oxidized Bronze Patina
- 10 Tin
- 11 Plastic
- 12 Fiberglass
- 13 Carbon Fiber
- 14 Vinyl
- 15 Plexiglass
- 16 PVC
- 17 Nylon
- 18 Polyester
- 19 Clear Glass
- 20 Frosted Glass
- 21 One Way Mirror
- 22 Mirror
- 23 Ceramic Glass
- 24 Asphalt
- 25 Concrete
- 26 Leaf Grass
- 27 Dead Leaf Grass
- 28 Rubber
- 29 Wood
- 30 Bark
- 31 Cardboard
- 32 Paper
- 33 Fabric
- 34 Skin
- 35 Fur Hair
- 36 Leather
- 37 Marble
- 38 Brick
- 39 Stone
- 40 Gravel
- 41 Dirt
- 42 Mud
- 43 Water
- 44 Salt Water
- 45 Snow
- 46 Ice
- 47 Calibration Lambertian

▼ Coating Filter

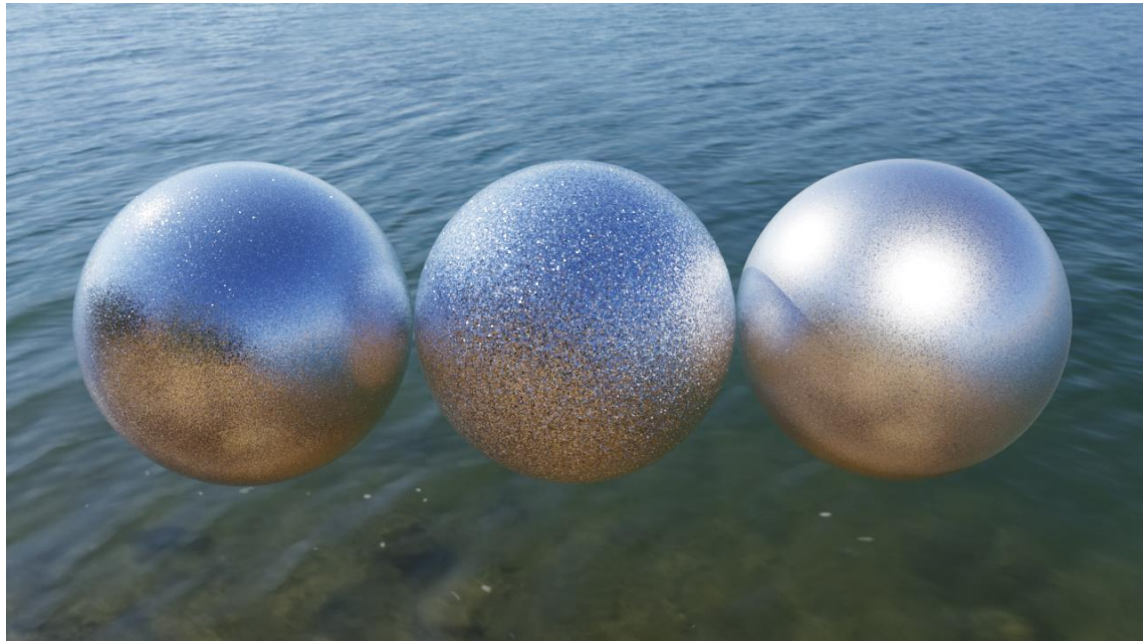
- None
- Paint
- Clearcoat

▼ Attribute Filter

- None
- Emissive
- Retroreflective
- Single Sided
- Visually Transparent

Future Work

Metallic Flakes



Huge Thanks

Charles Anderson
Derek Hase
Maik Herzberg
Jan Jordan
Anders Langlands
Matthias Raab
Ruediger Raab
Kai Rohmer
Masuo Suzuki

AUTODESK
Make Anything

MaterialX and OpenPBR at Autodesk

Ashwin Bhat,
Sr. Principal Engineer/Architect, Autodesk.

Safe Harbor Statement

This presentation may contain forward-looking statements about future results, performance or achievements, financial and otherwise, including statements regarding our guidance for our quarterly and financial year results.

These statements reflect management's current expectations, estimates and assumptions based on the information currently available to Autodesk. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause.

Autodesk's accrual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation. A discussion of the factors that may affect future results is contained in Autodesk's most recent SEC Form 10-K and Form 10-Q filings, including descriptions of the risk factors that may impact Autodesk and the forward-looking statements made in this presentation. If this presentation is reviewed after the time and date this presentation was first recorded, even if it subsequently is made available by Autodesk, on its web site or otherwise, this presentation may not contain current or accurate information.

Autodesk disclaims any obligation to update or revise any forward-looking statement based on new information, future events or otherwise.

AUTODESK Make Anything

"Autodesk makes software for people who design and make things"

ARCHITECTURE, ENGINEERING & CONSTRUCTION

R RVT | C C3D | N MAN | I IWX

PRODUCT DESIGN & MANUFACTURING

F 360 | I PRO | A CAD | V PRO

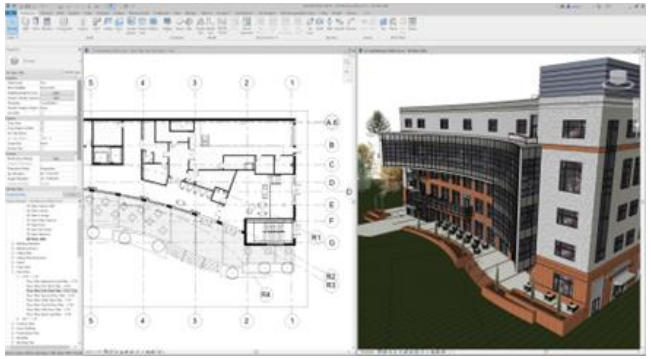
MEDIA & ENTERTAINMENT

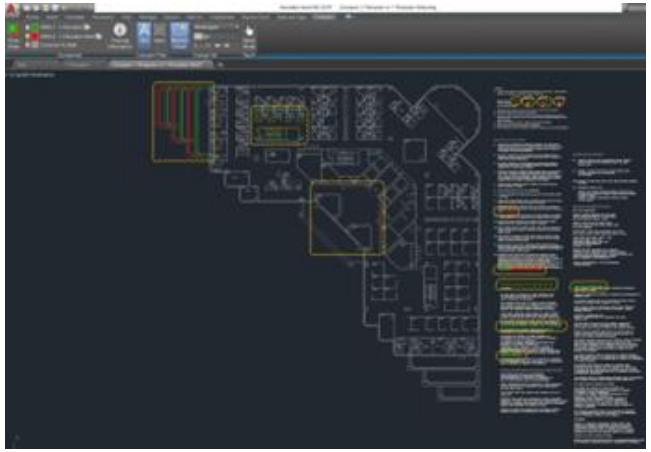
M AYA | 3 MAX | A ARN | F FLM

Credit: Henrik Edstrom

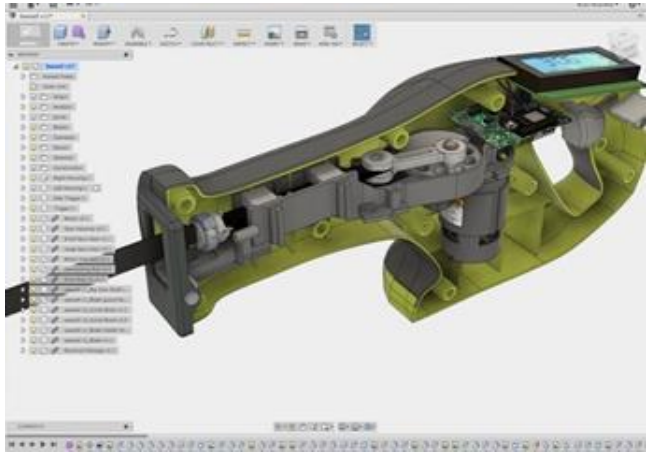
We need a wide range of Graphics Capabilities

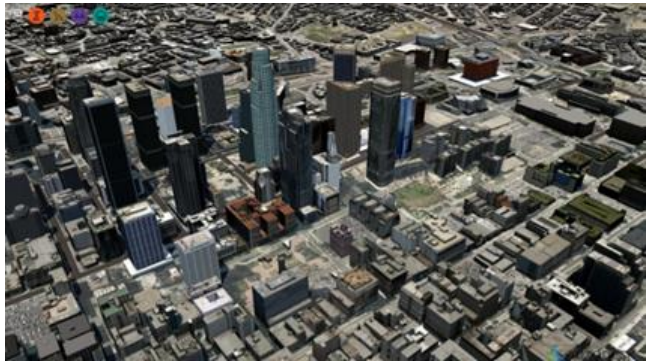
2D & Simple 3D





3D Modeling





Realistic Rendering

Autodesk's Graphics Objectives

Modern APIs

Vulkan

DirectX 12

WebGPU

Open Standards

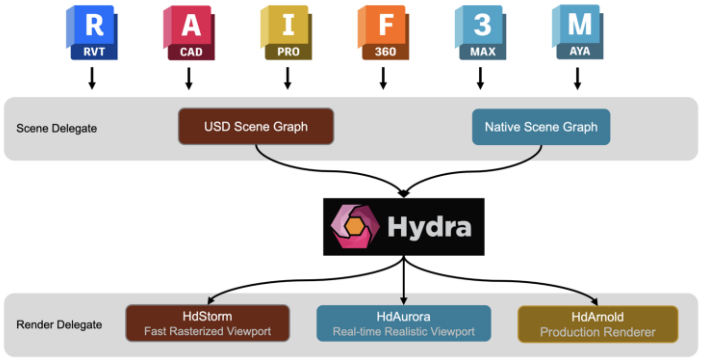
USD UNIVERSAL SCENE DESCRIPTION

MATERIALX

openpbr

OCIO

Decoupled Architecture



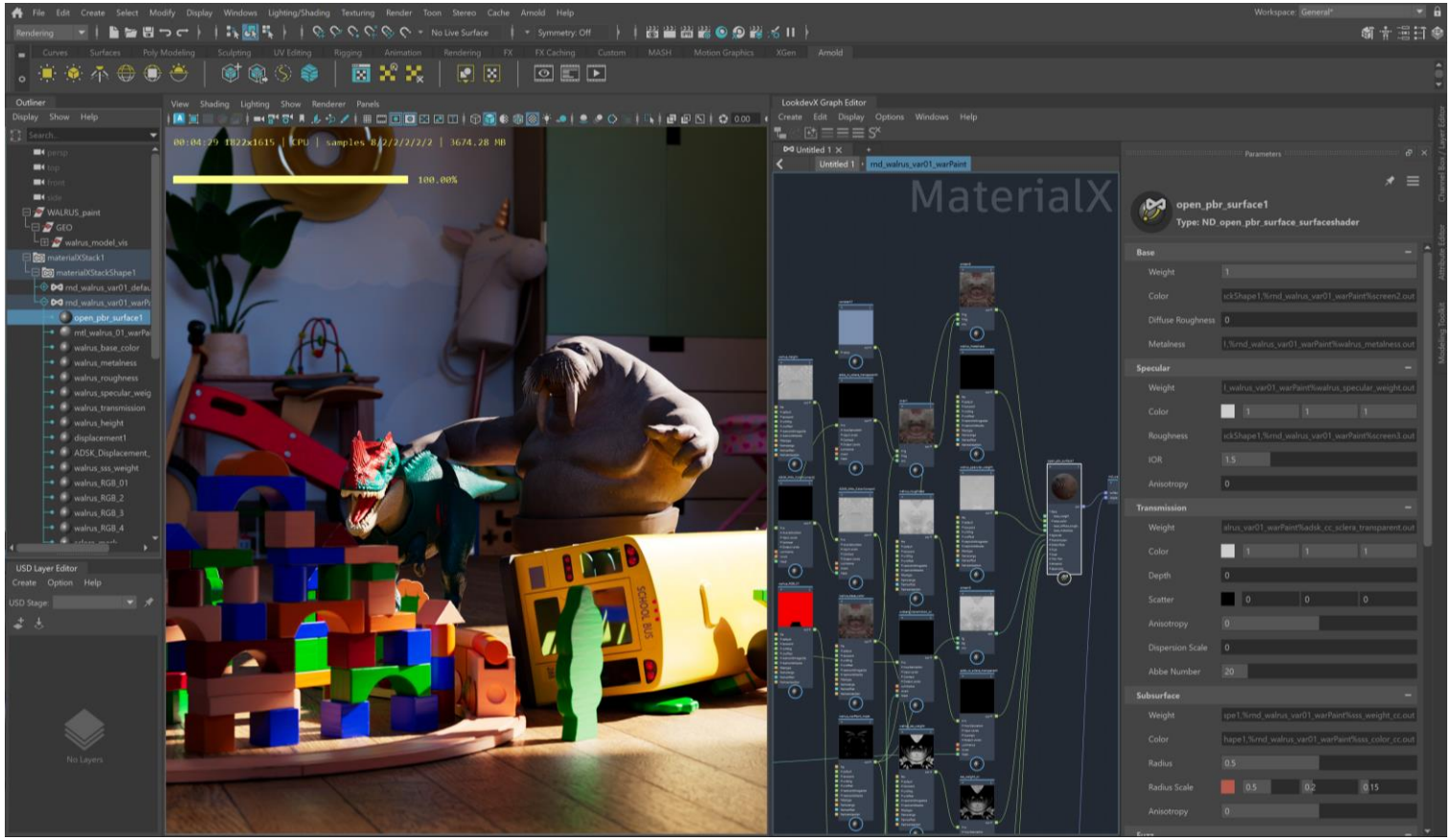
Available on Desktop, Mobile, and Web

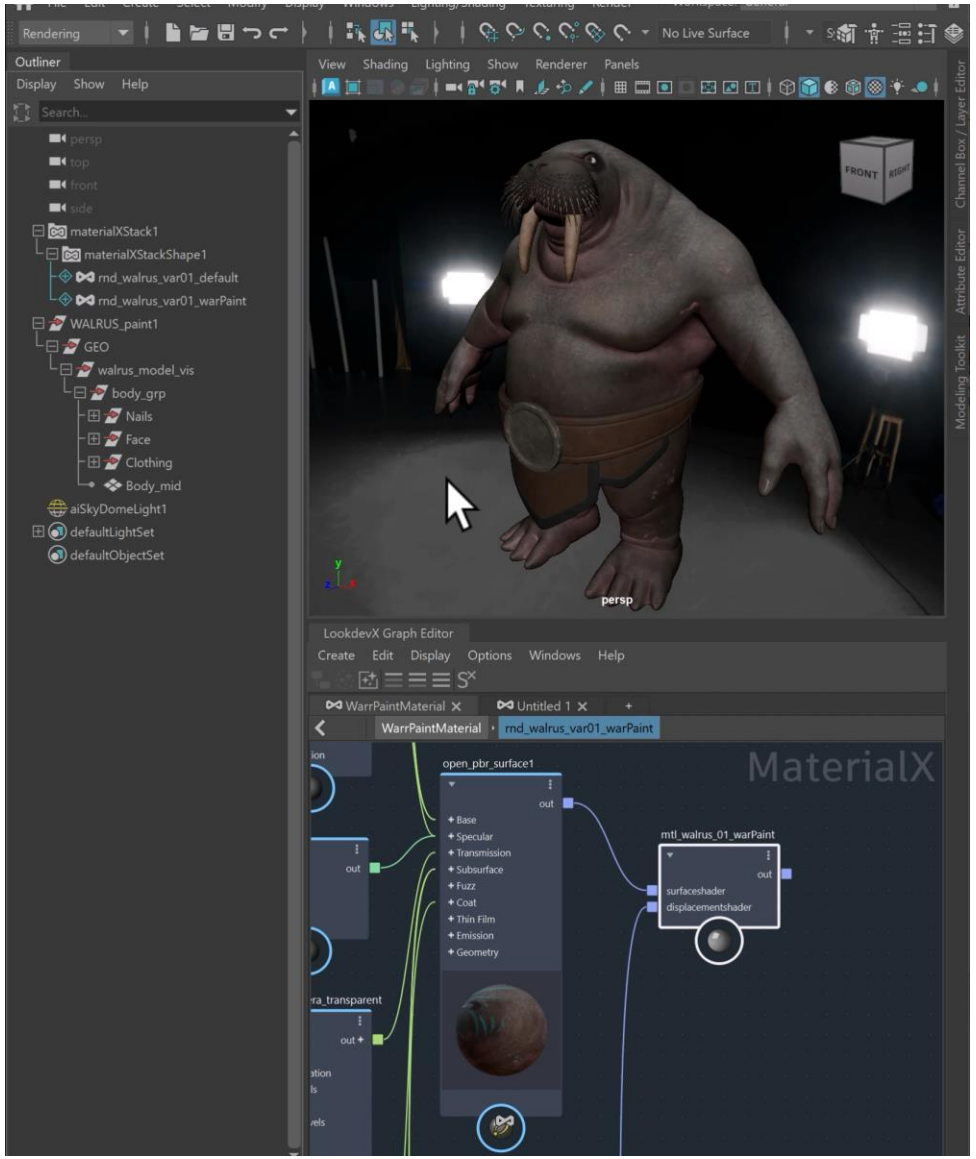
Maya, 3ds Max and Arnold

OpenPBR is the default.

Credit: Procedurals by Lee Griggs

LookDevX | Agnostic Material Editor



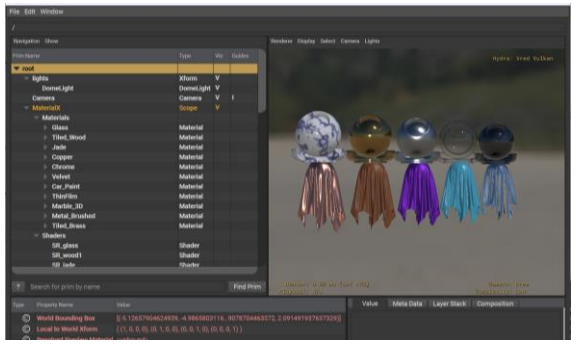


Credit: Nikola Milosevic

VRED

Visualization software for automotive and product design.

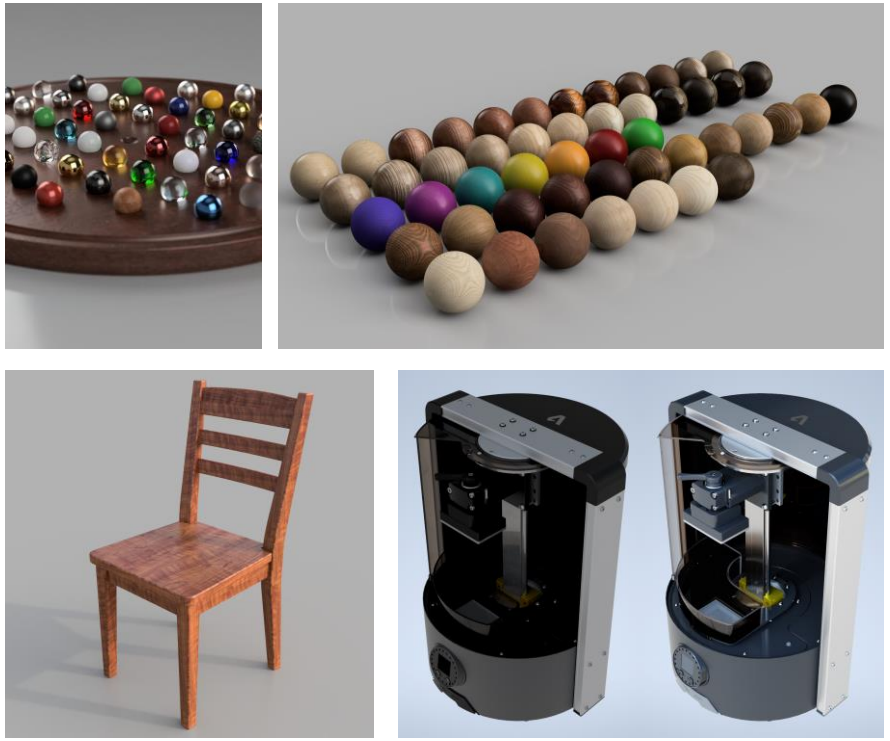
- MaterialX supported with OpenGL, Vulkan, CPU and GPU Ray tracer
- Hydra plugin via VRED Render API and Vulkan backend



Credit: Andreas Zieringer

Revit, Fusion and Inventor

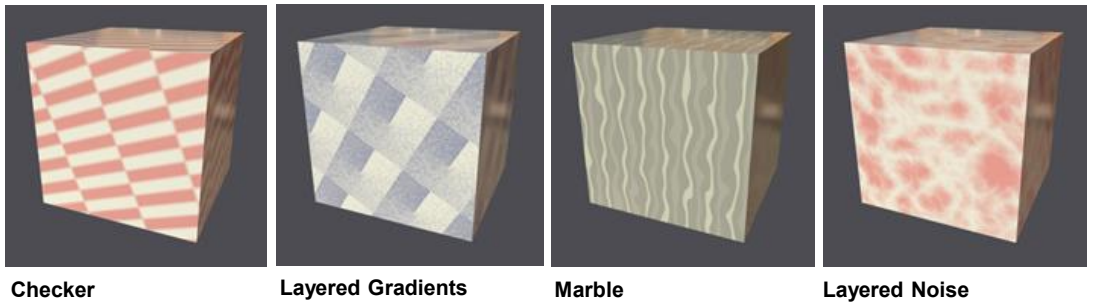
- MaterialX in various workflows for viewport, data import and export (e.g. USDZ)
  - Using MaterialX translation graphs.
- Supporting “legacy materials” for AEC and MFG using MaterialX.
  - Ensuring Design Intent is retained.

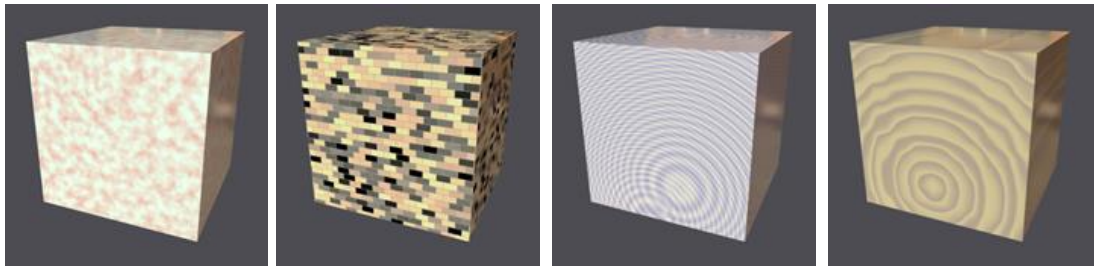


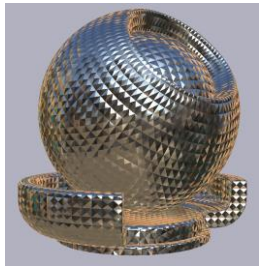
Credit: Roberto Ziche

Procedural nodes and pattern graphs

- Real-world units.
- Suitable for materials used in AEC & MFG industry.
- Authored using Maya LookDevX.
- Compatible with GraphEditor and Viewer.







Credit: Roberto Ziche

Procedural nodes

Knurl

Pattern of ridges or beads, typically found on the surface of metal parts.

Checker

With blur, stretching, and blur compensation (constant blur amount despite stretching)

Tiles

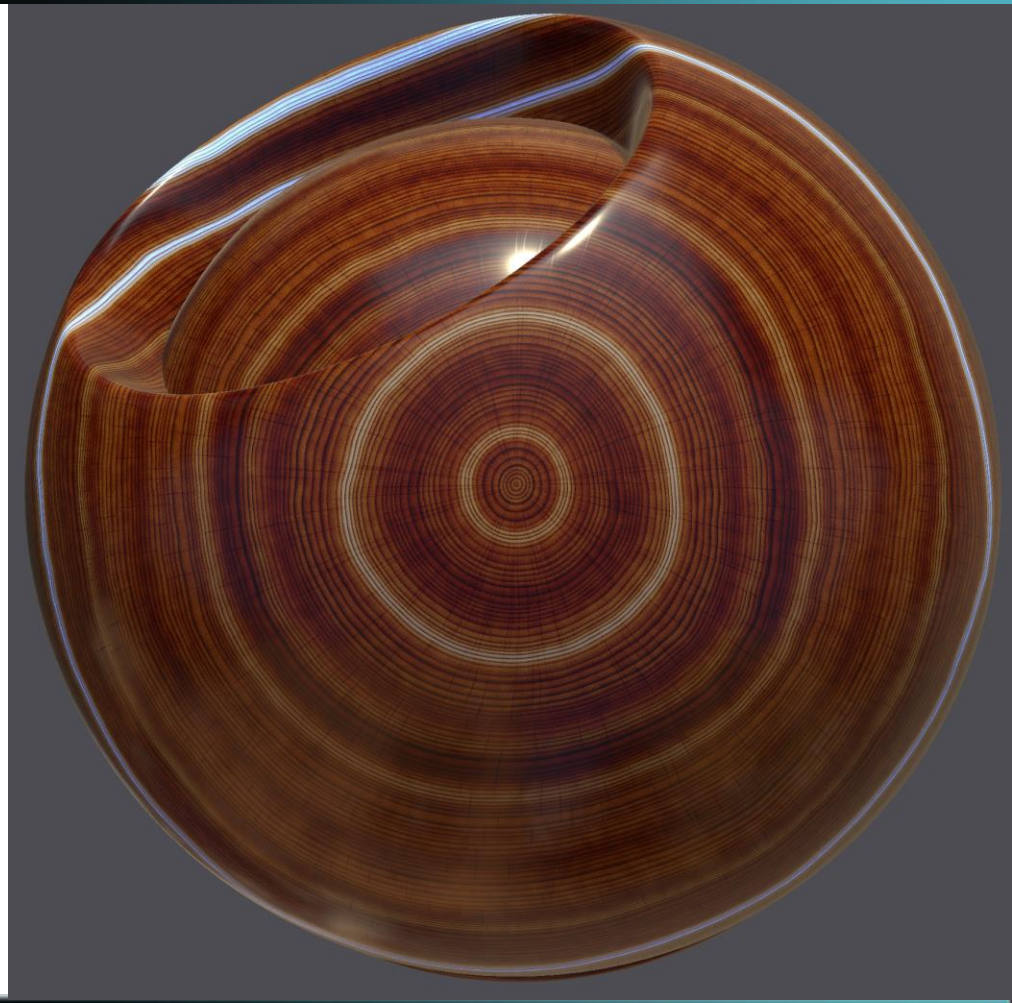
With grout noise, fade and color variances

Credit: Roberto Ziche

Procedural wood textures

Introduced in Fusion (~2015)

- Based on Procedural wood textures.
  - Albert J. Liu, Stephen R. Marschner, Victoria E. Dye
- OSL implementation 3dsMax:
  Zap Andersson (Autodesk)
- MaterialX implementation:
  Zilin Xu (UCSB, Autodesk Intern 2024), Roberto Ziche, Erika Harrison & Ashwin Bhat.



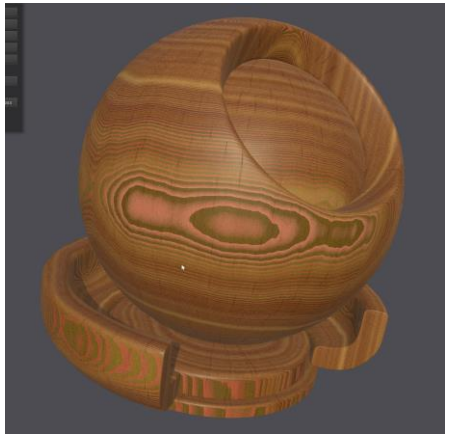
Procedural wood textures (species and variations)

Ash Glossy

Cherry Semigloss

Pine Stained

Walnut Semigloss



Ash Stained Semigloss

Maple Semigloss

Pine Glossy

Mahogany Glossy

Ash Painted

Koa Semigloss

Credit: Roberto Ziche

Performance Improvements

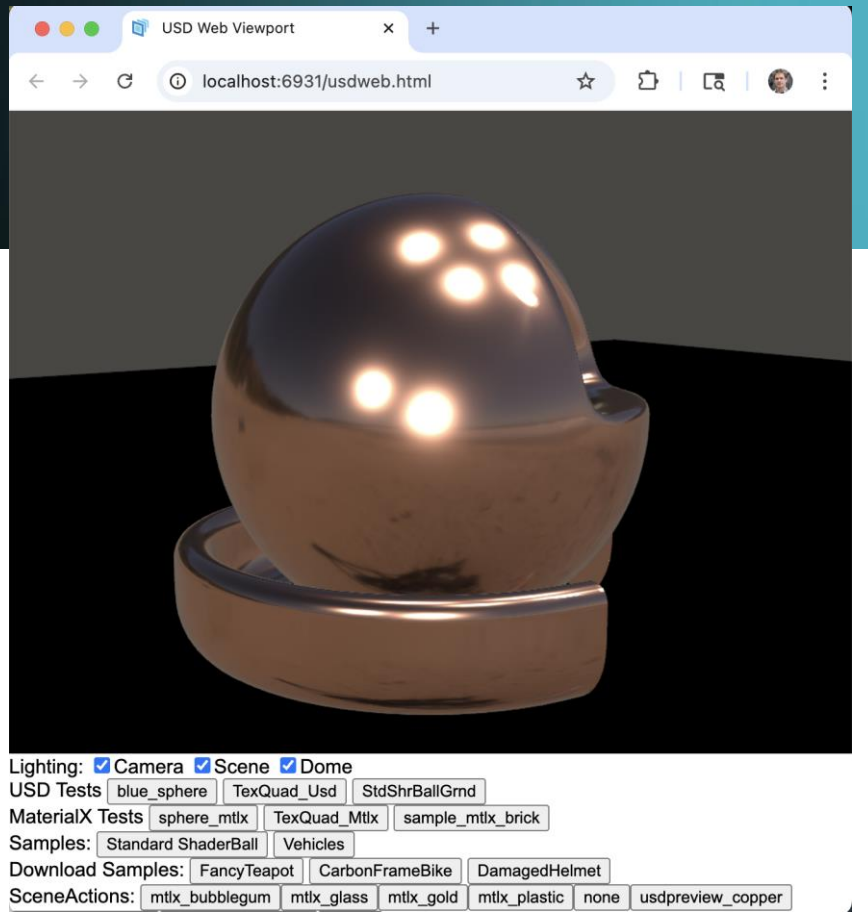
- MaterialX workflows:
  - Standalone
  - USD
  - Hydra 2.0 and Storm
- Simple aggregate test scene
  - Variations of surface shader BSDF and texture network topology
  - Does not use procedurals yet.
- Improvements
  - Data Library load time
  - Lobe Pruning
  - Parallel Shader Generation and compilation where supported
  - Caching at various levels i.e., MaterialX, Hydra Networks & .glslfx
  - APIs for early shader generation
  - Reduced time to first pixel in USDView from ~40s to ~8.5s *(aggregate test scene)*
- Next Steps
  - USD Vulkan MaterialX performance
  - Material editing

Thanks to Jerry Gamache and Pavlo Penenko for these improvements!

Aggregate test scene with ~600 materials

WebGPU

- MaterialX WebAssembly with OpenGL ES Shader generator available since 1.38.2.
- New! WebGPU Shading Language (WGSL) generator.
  - Generates GLSL that can be converted to WGSL using open toolchains e.g., dawn/tint.
- Validated in USDWeb.
- Thanks to Scot Brew for this contribution!

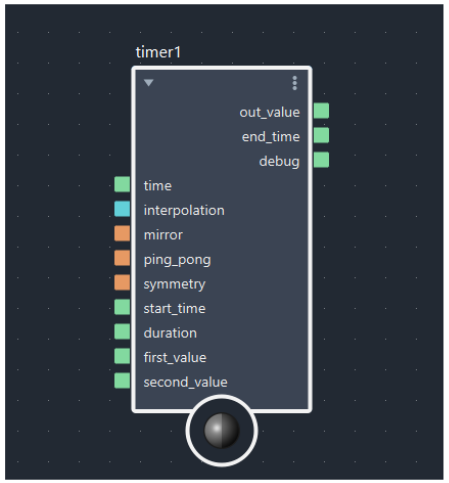


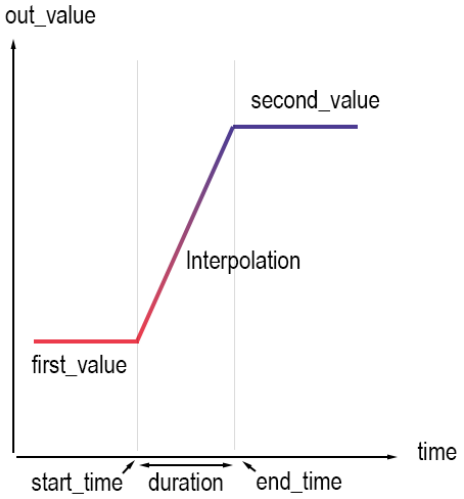
WA

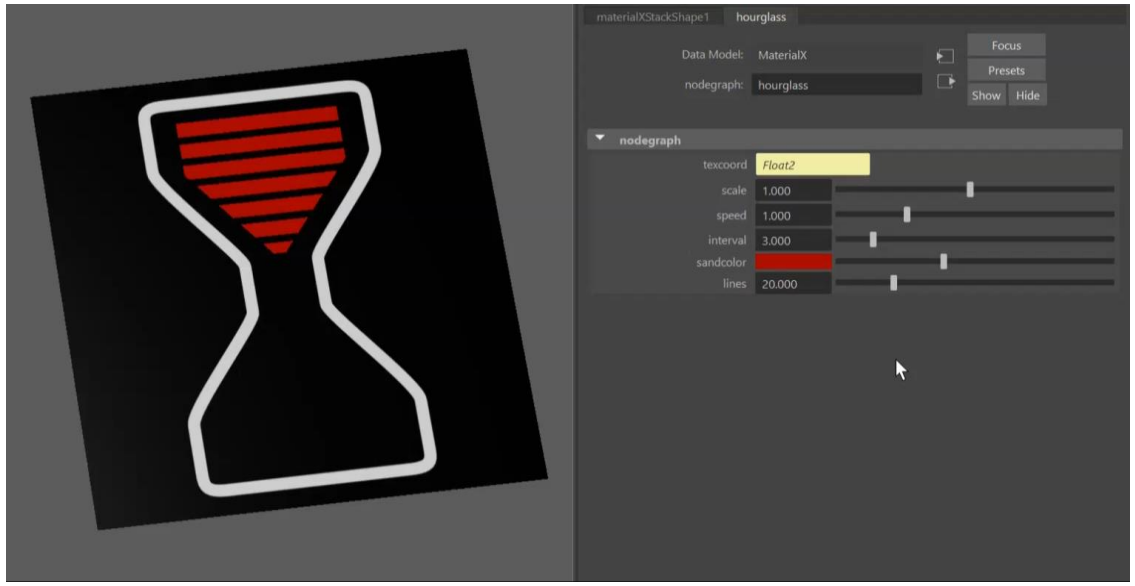
WebGPU

Animation Nodes

Experiments with animated procedurals, based on a few interpolation and utility nodes. Authored in Maya/LookdevX, and tested in MaterialX Viewer and OpenUSD Usdview.







AUTODESK
Make Anything

mx:flappy_bird
author: Roberto Ziche (Autodesk)


Thank You!
Questions?