open source days '23

#ASWF

/* ACADEMY SOFTWARE FOUNDATION */
The OpenPBR Surface Shading Model – Guido Quaroni, Adrien Herubel, et al

MaterialX in USD and Hydra – Karen Lucknavalai, Pixar USD

LookdevX in Maya – Nikola Milosevic, Autodesk

The Stråla MaterialX Editor – Magnus Pettersson, IKEA

Updates on MaterialX and MDL – Kai Rohmer, NVIDIA

RenderMan and MaterialX – Mark Manca, Pixar RenderMan

The QuiltiX Graph Editor – Manuel Köster, Richard Frangenberg
Virtual Town Hall Series

OpenPBR

August 2nd 2023
• Artists leverage creative applications from different vendors
• Look differences across apps is a major complaint
• "Innovate" and "Simplify" to put 3D artists forward
• Subproject of MaterialX within the ASWF
Announcing OpenPBR

A new material specification created by the teams at Adobe and Autodesk

This physically based shading model is being developed to offer creatives a more artist-friendly bridge between different software applications.

The new shading model will be a subproject of MaterialX within the Academy Software Foundation.
The following side-by-side renders are from a scene we affectionately called “Shader Playground.”

They are a work in progress.

Every 3D application has its differences but both art & tech teams are already seeing the benefits of OpenPBR working similarly inside Autodesk Arnold and the Adobe proprietary renderer.
OpenPBR inside Maya + Arnold

OpenPBR inside Adobe proprietary renderer
OpenPBR inside Maya + Arnold

OpenPBR inside Adobe proprietary renderer
OpenPBR – Technical Overview

- A physically based surface model based on **Autodesk Standard Surface** and **Adobe Standard Material**
- An evolution of two production-proven models
Principled and physically based layering specification

Formal structure (implementation agnostic)

Particular implementations

\[
\begin{align*}
\text{f} & = \text{lerp} \left( f_{\text{translucent}}, f_{\text{surface}} \right) \\
\text{f}_{\text{surface}} &= \text{f} \times \text{lerp} \left( 1, 1 - E(f_{\text{translucent}}), \text{f} \right) f_{\text{control-base}} \\
\text{f}_{\text{control-base}} &= \text{C} \times \text{lerp} \left( 1, T_{\text{control}} \left( 1 - E(f_{\text{translucent}}) \right), \text{C} \right) f_{\text{trans-substrate}} \\
\text{E} &= \text{lerp} \left( 1, T_{\text{control}} \left( 1 - E(f_{\text{translucent}}) \right), \text{E} \right) \\
\text{f}_{\text{trans-substrate}} &= \text{lerp} \left( f_{\text{dielectric-base}}, f_{\text{substrate}} \right) \\
\text{f}_{\text{dielectric-base}} &= f_{\text{specular}} + (1 - E(f_{\text{specular}})) f_{\text{dielectric-base}} \\
\text{f}_{\text{specular}} &= \text{lerp} \left( \text{lerp}(\text{f}_{\text{diffuse}}, f_{\text{specular}}, S), f_{\text{specular}}, T \right)
\end{align*}
\]
OpenPBR – Technical Overview

Commonalities and differences
Dedicated SSS component
Inspired by Standard Surface

[Diagrams showing lit and shadowed regions with different g values]
OpenPBR – Technical Overview

Commonalities and differences
F82-tint metal reflectivity
Inspired by Adobe Standard Material
Commonalities and differences
Sheen/Fuzz on top of coat
Different from both original models
OpenPBR - Conclusion

- Great collaboration and good spirit
- An open standard under MaterialX
- Early reviewing by third parties
- Integration in products
- Next steps & future initiatives
- Reach out and discuss at SIGGRAPH

Autodesk Booth Wed 9th & Thu 10th at 11am
OpenPBR - Contributors

- Zap Andersson
- Paul Edmondson
- Julien Guertault
- Adrien Herubel
- Alan King
- Peter Kutz
- Andréa Machizaud
- Jamie Portsmouth
- Frédéric Servant
Virtual Town Hall Series

MaterialX in USD/Hydra

Karen Lucknavalai, Pixar
Aug 2, 2023
MaterialX in USD/Hydra - Updates

- MaterialX enabled by default in USD
- Support up to MaterialX version 1.38.7
MaterialX in USD/Hydra - Updates
MaterialX in USD/Hydra - Updates
MaterialX in USD/Hydra - Updates

- More flexibility in how materials can be defined
  - Node placement wrt nodegraphs
  - Nodegraph input and interface connections
  - Material names
- Wider support for Custom Nodes
- Other Improvements
  - Can include relative paths
  - Handling string array processing
  - More metadata is parsed into the SDR properties
MaterialX in USD/Hydra - Updates

- Aligning MaterialX implementation of USDPreviewSurface Materials
- Opensourced and extended the imaging tests
MaterialX in USD/Hydra - Updates

- Aligning MaterialX implementation of USDPreviewSurface Materials
- Opensourced and extended the imaging tests
MaterialX in USD/Hydra - Updates

- Distant light support
- `geompromp/primvar` values in Storm
- Initial material caching added to Storm
Support for Displacement
Fixes for:
• Texture coordinate names
• Normal Maps
Future Work:
- Normal Map issues in Storm
- MaterialX Colorspace handling in Hydra (Goal: 23.11)
Thank you

More information at the USD, Hydra BOF at SIGGRAPH:

Tuesday Aug 8, 2pm - 4pm
LookdevX | Released in Maya 2024

- Agnostic Material Authoring tool
  - Native USD material authoring
    - Resolving USD Material Story
  - Native MaterialX authoring (in progress)
  - Arnold Integrated
  - UFE layer used for DCC connection
    - Enabling DCC portability
Release 2024 | LookdevX

Delivered Key Features

- Material manipulation from LookdevX, Outliner, VP and Attribute Editor
- Direct Material Assignment
- Arnold support
- MaterialX Viewport support
- Color Managed LookdevX Graph
- Material Authoring workflow
- USD Material Scope control
- Multi-selection actions
- Node & Graph Duplication
- Undo - Redo
- Enums support AE/PE
- And more......
Considering | LookdevX

Key Features

- Publishing workflows
- Presets workflow
- Icon Shelf
- Searching Nodes and/or Attributes
- Dimming and coloring Nodes and Noodles
- Expose Custom Attribute control on Node
- Material Swatches
- Create node from Port
- Deep Attribute Promotion
- Ramp node
- Automation tools
- Hydra implementation support
  - Material graphs can be accurately represented in Storm and Arnold delegate

- MaterialX Maya native support

- 3ds Max Implementation

- Procedural Material Binding – USD
  - Late material binding enabled by Biforst-USD
Virtual Town Hall Series

Stråla MaterialX Editor

2023-08-02
Agenda

- MaterialX at IKEA
- Technical Design
Technical Design

- C++
- **Dear ImGui** for interface
- **ImGui Node Editor** for displaying networks
- Uses MaterialX OSL backend for code generation
- OIIO for texture handling
- Modified OSL testrender for previews
Node editing
Swatches and texture handling

Material from AMD material library
MaterialX to gltf
NVIDIA Updates on MaterialX and MDL

August 2nd, 2023

Kai Rohmer
Senior Software Engineer
NVIDIA
MaterialX and ShaderGen

MaterialX

An open standard for network-based CG object looks originally developed by Lucasfilm

https://www.materialx.org
https://github.com/AcademySoftwareFoundation/MaterialX

MaterialX Physically-Based Shading Nodes

https://www.materialx.org/assets/MaterialX.v1.38.PBRSpec.pdf

ShaderGen

Transforms the MaterialX descriptions into executable code
Contribution by Autodesk

NVIDIA Material Definition Language (MDL)
- Domain-specific language to define PBR materials
- Declarative components to compose material graphs
- Procedural texturing functions to drive material inputs

SDK / Compiler / Backends
- Translation to HLSL, GLSL, PTX, x86, ARM, LLVM-IR
- Interface to sample and evaluate materials
- Independent of the lighting and rendering algorithms

Open Source (BSD 3-clause license)
https://github.com/NVIDIA/MDL-SDK
MaterialXGenMdl
Library for MDL Code Generation

Official part since MaterialX 1.38.0

https://github.com/AcademySoftwareFoundation/MaterialX/tree/main/source/MaterialXGenMdl

Joint development from

Autodesk & NVIDIA

Used for example in

Autodesk VRED & NVIDIA Omniverse
MaterialXGenMdl released with 1.38.7

Better integration in the MaterialX Test Suite

MDL DXR Example Renderer matches the MaterialXTest scene

added 3-way comparison to the tests_to_html.py script
MaterialXGenMdl released with 1.38.7

Full Sheen BSDF Support
Sheen can be layered over arbitrary nodes

<table>
<thead>
<tr>
<th>GLSL</th>
<th>OSL</th>
<th>MDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>(approximation only)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MaterialXGenMdl released with 1.38.7

Proper Support for the Thin-Film BSDF

over dielectrics, conductors, and the generalized Schlick

GLSL  OSL (work in progress)  MDL
MaterialXGenMdl released with 1.38.7

New Generalized Schlick EDF

Coordinated specification between MDL and MaterialX workgroup
MaterialXGenMdl released with 1.38.7

Volume Emission

Full Support for Add BSDF, EDF, and VDF Nodes

- Implemented by new MDL `unbounded_mix`
- Note, can violate energy conservation laws!

Minor Updates and Fixes

- Support for swizzles on custom types (used in UsdPreviewSurface)
- Handle structures that have Surface Shader fields
- Resource resolution on application side for more flexibility
- Improved generalized Schlick BSDF
MaterialXGenMdl  In Progress

Displacements

Already in the github main branch

Versioning

Add MDL ShaderGen option to select the target MDL version 1.6, 1.7, or 1.8

Rendered in NVIDIA Iray
MaterialX in Omniverse In Progress
MaterialX editing based on USD Shade Graphs
Thank you

nVIDIA + MATERIALX

MDL

Rendered in NVIDIA Omniverse
Virtual Town Hall Series

RenderMan and MaterialX

Mark Manca

August 2, 2023
The path to create once, use anywhere

- Final renders for Film/Streaming
- Previews in GL for Animators and others in different DCCs
- LED walls/game engine driven env
- Theme parks
- Digital backlot/Legacy content
- Interchange w/other studios
MaterialX and Future RenderMan

• Build support for the full expressivity of MaterialX
  • Down to the OSL primitive closure level
  • Support arbitrary composition of these “lobe primitives”
• Continue innovating with ILM on MaterialX Lama
• Building it all in XPU
Virtual Town Hall Series

Manuel Köster & Richard Frangenberg

2nd of August, 2023

#ASWF
What is **QUILTIX**

- Material Node Editor based on **MATERIALX**
- Export and Import Materials
- Live preview of your shaded Assets based on USD‘s **Hydra**
Why did we create **QUILTIX**

- Artist friendly
- Hydra renderer integration for production ready look development
- Easily integrate-able into DCCs and Pipelines
- ♥ Open Source ♥
Nodegraphs

- Organize nodes in subgraphs
- Expose relevant parameters
- Reusable nodegraphs
- Edit node definitions
Nodegraph Demo (Video)
Pipeline Integration

- Python/Pyside allows easy pipeline integration
- Replaceable viewer/scenegraph widgets
- Custom publish processes
Pipeline Friendly

- Highly decoupled and extensible
- Customizable through environment variables
- BYO MaterialX/USD (if you like)
- Open source (Apache)
Tech stack

github.com/jchanvfx/NodeGraphQt
Compatibility

- OpenUSD 22.08 - MaterialX 1.38.3 (+ Arnold)
- Experimental: OpenUSD 23.08 - MaterialX 1.38.7
- Karma: OpenUSD 22.05 (Houdini 19.5)
- Other USD/MaterialX versions (untested)
How does **QUILTIX** work

Nodegraph & Properties

- QuiltX nodegraph
  - serializes to MaterialX data

MaterialX XML String

- imports

Shading Layer

- Binds material to prims

Stage Tree / Scenograph

- UsdStage
- sublayers

Viewport

- Hydra
- Usdview's Stageview
- sublayers

USD prims

- Render Delegate
Next steps

- CI/CD
  - Testing & Linting
- Community collaboration
- Deepen NodeGraphQt collab
- Features
  - Renderer settings
  - Support more MaterialX features (like parameter folders)
  - Colorspace support
- And hopefully much more
Thanks to

open Source days'23

[Logos and Images]

#ASWF

/* ACADEMY SOFTWARE FOUNDATION */
Try out today!

github.com/PrismPipeline/QuiltiX
pypi.org/project/QuiltiX
Thank You!
Questions?