Safe Harbor Statement

Autodesk and Adobe may make statements regarding future events and/or statements regarding planned or future development efforts for our existing or new products and services. We wish to caution you that such statements reflect our current expectations, estimates and assumptions based on factors currently known to us and that actual events or results could differ materially. Also, these statements are not intended to be a promise or guarantee of future delivery of products, services or features but merely reflect our current plans, which may change. Purchasing decisions should not be made based upon reliance on these statements. The statements made in this presentation are being made as of the time and date of its live presentation. Autodesk and Adobe do not assume any obligation to update any statements made to reflect events that occur or circumstances that exist after the date of this presentation.

All images copyright of their respective owners.
MaterialX: Origins at Lucasfilm
Open-Source Data Formats

ALEMBIC

OpenEXR

OpenColorIO
MaterialX Initiative

- Rich material description, node-based and color space-aware
MaterialX Initiative

- Rich material description, node-based and color space-aware
- First significant usage on *Star Wars: The Force Awakens* in 2015
MaterialX Initiative

- Rich material description, node-based and color space-aware
- First significant usage on *Star Wars: The Force Awakens* in 2015
- Open standard published in 2016
MaterialX Initiative

- Rich material description, node-based and color space-aware
- First significant usage on *Star Wars: The Force Awakens* in 2015
- Open standard published in 2016
- Open-source codebase released in 2017
Early Interest at Autodesk

- Consistent looks across Autodesk DCCs
  - multiple renderers in a single DCC
  - physics as ground truth
- Abstract Material Graph (AMG)
- abcMaterial, MaterialX
- Worked together with Lucasfilm to help build an open industry standard
ShaderX Collaboration

- A partnership between Lucasfilm and Autodesk beginning in June of 2016
- Inheriting the best of both MaterialX and Abstract Material Graphs
- Autodesk begins developing two key extensions to MaterialX
Physically-Based Shading Nodes

- The first new feature is a standard set of physically-based shading nodes
- In addition to patterns, the underlying physically-based shaders can now be portably captured
- MaterialX ships with shading graphs for Standard Surface and USD Preview Surface
The second new feature is a framework for shader code generation

Automatic conversion of a MaterialX document to domain-specific shading code for rendering

MaterialX ships with support for OSL and GLSL, with additional languages planned
MaterialX Viewer

- Leverages MaterialX shader generation in combination with the open NanoGUI framework
- Provides a ground truth reference for renders of MaterialX content
- Provides a reference for integration of MaterialX shader code generation into other applications
Open @ Autodesk

- Autodesk is a strong supporter of Open Source software
- Our customers’ pipelines are heterogeneous
- Founding member of the ASWF
  - providing funding and technical expertise
  - helping drive strategic direction
Open @ Autodesk

- Actively contributing to many existing projects
  - OCIO v2
  - MaterialX
  - USD
- Have open sourced our own projects:
  - AnimX
  - ShaderX
  - sitoa
  - Standard Surface
Standard Surface v1.0.1

An open spec by Autodesk
What is Standard Surface?

- Open uber-shader specification
- Artist-friendly parameters
- Production proven
- Supported in Autodesk products
Goals

- Compactly represent most materials
  - Modern set of scattering lobes
- Easy to use
  - Minimal set of intuitive parameters
- Simplification for
  - partial representations
  - real-time applications
Whitepaper

- Available now [autodesk.github.io/standard-surface](http://autodesk.github.io/standard-surface)
- Open source (Apache License 2.0) [github.com/autodesk/standard-surface](https://github.com/autodesk/standard-surface)
- Reference implementation
  - MaterialX
  - OSL
Physically Based Shader Design in Arnold

by Anders Langlands

Introduction

The Comprehensive PBR Guide by Allegorithmic - vol. 1

Physically-Based Shading at Disney

by Brent Burley, Walt Disney Animation Studios

[Review Aug. 30, 2013. Correct normalization factor in Equation 1.]

1 Introduction

Following our success with physically-based hair shading on Exodus [27], we began considering physically-based shading models for a broader range of materials. With the physically-based hair model, we were able to achieve a great degree of visual realism while maintaining artistic control. However, it proved challenging to integrate the lighting of the hair with the rest of the scene which had still used traditional "ill-bee" shading models and projection lights. For some shots we wanted to increase the realism of our materials while making lighting responses more consistent between materials and environments and also wanted to improve artist productivity through the use of simplified controls.

When we began our investigation it wasn’t obvious which model to use or even how physically-based we wanted to be. Should we be perfectly energy conserving? Should we favor physical parameters like index of refraction?

For diffuse, Lambertian seemed to be the accepted norm, while specular seemed to get most of the attention in the literature. Some models such as Anichkov et al. [5] aimed to be irremovable with BRDFs of materials such as wood, while others such as Torrance-Sparrow [28] aimed to be matte with BRDFs of materials such as metal.
Layered Mixture Model

- Transparency
- Emission/additive/
- Specular reflection (metal)
- Specular reflection (coating)
- Specular reflection
- Specular retro-reflection (sheen)*
- Diffuse reflection
- Diffuse transmission*
- Subsurface scattering*
Closure representation

1 - opacity + opacity

transparency +

close +

1 - metalness + metalness

emission * emission_color +

specular * specular_color +

1 - specular * specular_color * reflectance(spec_brdf)

spec_brdf +

1 - transmission +

transmission * (transmission_depth == 0 ?

transmission_color : 1)

spec_btdf +

1 - sheen * reflectance(sheen_brdf)

sheen_brdf +

subsurface * subsurface_color

(1 - subsurface) * base * base_color

diffuse_brdf +

thin_walled +

1 - thin_walled

diffuse_btdf +

subsurface
Transparency
Coat
Emission
**Metal**

![Three images of metallic objects with different textures]

<table>
<thead>
<tr>
<th>Transparency</th>
<th>Specular reflection (coating)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emission (positive)</td>
<td>Specular reflection (metal)</td>
</tr>
<tr>
<td>Specular transmission*</td>
<td>Specular retro-reflection (sheen)*</td>
</tr>
<tr>
<td>Diffuse reflection</td>
<td>Diffuse transmission*</td>
</tr>
<tr>
<td>Subsurface scattering*</td>
<td></td>
</tr>
</tbody>
</table>
Thin film
Specular reflection & transmission
Sheen
Diffuse reflection & transmission
Subsurface scattering
Future work

- Reciprocity
- Improved layering model
Contribute!

Join the conversation!

github.com/autodesk/standard-surface
Why we need a new standard Uber Shader
Substance Materials in a nutshell

- Procedural packages, from a comp graph
- Typically bake **specific textures** before render
  - e.g. albedo, roughness, height, etc...

![Diagram](Compositing graph \rightarrow Bake to textures)
Maximize Portability

- Rely on standard Uber-Shaders
- Know how to approximate for speed
Maximize Portability

- Conversion to other shading models is known
- Easy to adopt and port (even when lossy)
Two production strategies for Surfaces

- **"Playmobil": Uber Shaders**
  often good for 90% assets. More common and makes exchange easier

- **Lego: Lobe combining**
  more powerful, covers final 10% of LookDev challenges
Can work with either

- Thanks to the ShaderX addition, MaterialX now supports the Lego approach

- With packaging and subgraphs, it can support well Playmobil approaches too
We like the portability of UberShaders

- Our baseline shading models don't cover cases that are now common
  - Too many semi-documented extended variants for advanced lobes

- We were considering drafting our own updated standard
How Standards Proliferate:
(See: A/C chargers, character encodings, instant messaging, etc)

Situation: There are 14 competing standards.

14?! Ridiculous! We need to develop one universal standard that covers everyone's use cases. Yeah!

Soon:

Situation: There are 15 competing standards.
Standards are hard

- UberShaders **need** strong standards to be successful for exchange purposes
- **Need** very good documentation
- One standard is better than none
Standards are hard

- UberShaders need strong standards to be successful for exchange purposes
- Need very good documentation
- One standard is better than none
- One standard is also better than too many
  - Must have very good reasons to create a new one
- Takes time to gather support from other vendors and studios
  - If only we could find someone to collaborate right from the start...
Let’s collaborate on a standard

- Last Siggraph, Autodesk included us on a draft whitepaper for a Standard Surface

- Perfect timing to start a collaboration
We like Standard Surface

- Good balance:
  - More complete feature set
  - Not overcomplicated
  - Considers simplifications for preview purposes

- Well documented, well thought through
  - Crucial to adopt it partially, or evolve towards it

- First serious effort to make a collaborative BXDF UberShader
  - Included in the discussion top experts in the field
We like Standard Surface

- We are interested in growing some of our existing materials towards it
  - Would like to fully embrace it natively
  - That is not the state currently

- We still must be able to export to most models
Implemented for Substance Designer
Implemented for Substance Designer
Contribute

- Please follow and participate to this project on GitHub!
MaterialX Prototype in Substance Designer

David Larsson
Making and sharing Materials
Shaders
MaterialX Prototype in Substance Designer
Demo

MaterialX Editor in Substance Designer
Demo
Droid material setup in Designer

© & ™ Lucasfilm Ltd. Used with Permission.
Demo

Droid material setup in Designer

© & ™ Lucasfilm Ltd. Used with Permission.
Demo

Droid washing in Substance Painter

© & ™ Lucasfilm Ltd. Used with Permission.
Maya / Arnold Demo
Questions?