open shading language

TAC Project Review

October 30, 2024

#ASWF

What is Open Shading Language?

- Shading language for modern production renderers
- A compiler and efficient runtime for the language
- Leverages LLVM for execution on CPU and GPU
- Project just turned 16 years old (first commit Sept 3, 2008)
- ASWF member since 2020

What is Open Shading Language?

- Software supporting OSL:
  - 3ds Max, Arnold, Blender/Cycles, 3delight, Renderman, V-Ray, Octane, Redshift, …
- Studio renderers:
  - Animal Logic's Glimpse, Sony Imageworks' SPEAR/Arnold, Illumination Labs
- Ties to other ASWF projects:
  - Dependencies: OpenImageIO, OpenColorIO, OpenEXR, OpenVDB
  - Used by: MaterialX
  - Collaborate with: OpenImageIO, MaterialX, OpenColorIO
- At least 200 films (that we know of… please tell us!)

OSL Technical Steering Committee

- Meets every other week
  - TSC Chair: Chris Kulla
  - Architect: Larry Gritz
- 11 Total Voting members
  - Autodesk, Intel, Blender, Epic Games, DNEG, NVIDIA, Sony Imageworks, Apple, Animal Logic, Laika, Pixar
- Usually 5-10 additional attendees

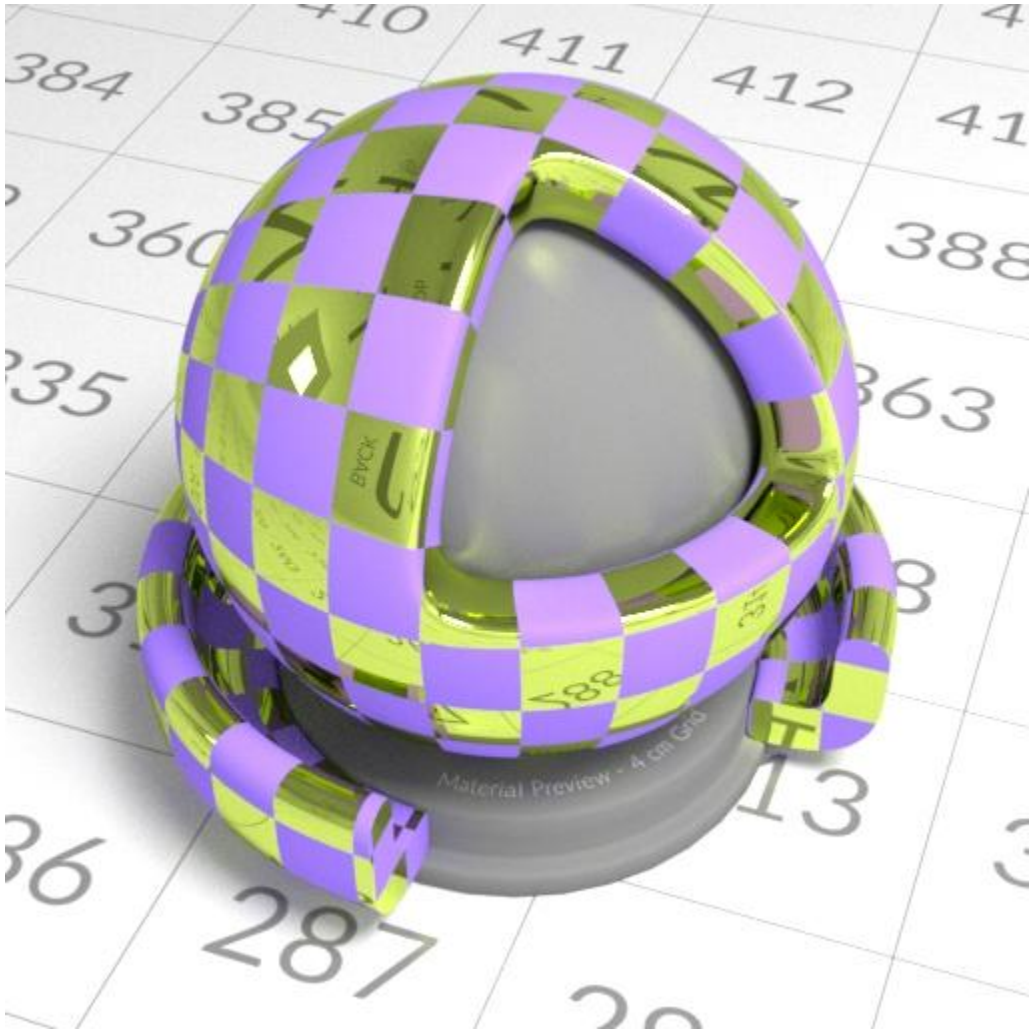
What's new in OSL this year?

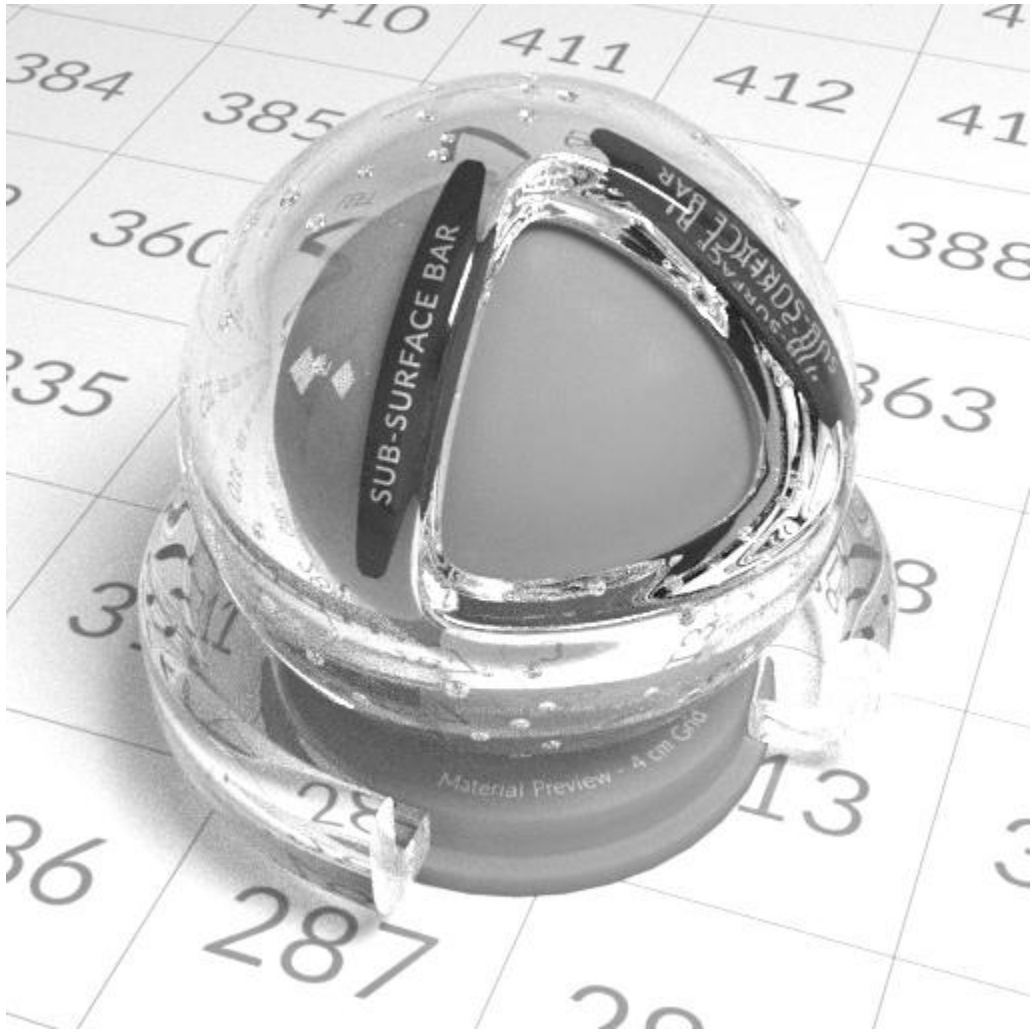
- Released OSL v1.13
  - 10 point releases since February
  - Bumped minimum versions across our dependencies
    - Optix 7, OpenImageIO 2.4, Imath 2.4
  - Initial support for modern LLVM (v18)
  - Improvements to NVPTX backend (Nvidia GPUs)
  - Texture call accepts colorspace keyword
  - API changes for renderer integration
  - `lockgeom` metadata is now `interactive` and `interpolated`
    - More accurately describes intent of shader parameters
    - Reduces runtime optimization costs for large networks
  - New journaling API for printing messages and errors from shaders
    - More GPU friendly
  - Many bug fixes

OSL 1.14 highlights – later this year

- `Testrender`
  - MaterialX/OpenPBR Oren-Nayar & Sheen BSDFS
  - Support for triangle meshes, displacement, GPU
- A variety of improvements to OptiX/CUDA back end
  - More feature parity, performance, bug fixes
  - Production use in several renderers now
- No more boost dependency!
- Support for LLVM 18 and 19
- Support for OpenImageIO 3.0
- Docs now on https://open-shading-language.readthedocs.io

OSL 1.14 highlights – later this year





OSL 1.14 highlights – later this year

- In progress
  - Raise dependency minimums (VFX Platform 2022):
    - C++17 / gcc 9.3 / Python 3.7 / Imath 3.x
    - Maybe also bumps for Cmake, LLVM
  - Dependency auto-build capabilities (similarly to OpenImageIO)
  - Continued GPU improvements
  - Improved guidelines on standard attributes and UI metadata
    - See Github Discussion page
    - Need to update documentation, and implement suggestions in testshade/testrender
    - Want to improve portability of OSL shaders across renderers
    - Help needed from DCC vendors for UI discussions

Roadmap – beyond 1.14

- Continued transition from RendererServices to “free functions” provided as LLVM bitcode
- SPIR-V back end (ongoing work by Intel)
- Lightweight **`oslcomp`** (new preprocessor, no more libclang)
- Evolutions to language syntax
  - New datatypes to more closely match MaterialX
  - Ability to re-evaluate connected nodes in a material graph
  - Shader-writer improvements: named function call arguments? templates?
- Infrastructure work
  - Need Windows and GPU CI (help wanted!)

Dev Days

- First time participation
- Unsure how many would join
- Got 3 participants
  - 1 patch accepted
  - 1 patch pending CLA
  - 1 contributor spent the full day figuring out the windows build (instructions may turn into a PR)
- Better than expected, will likely keep participating
- Hard to find “good first issues” for us

Current Status

- OpenSSF Best Practices
  - Passing 100%
  - Silver 84%
  - Gold 78%
- Most outstanding items are security related
  - Some low hanging fruit around docs, signing, etc …
  - Need help to add fuzzing to the project
  - OSL lets you execute arbitrary user code *by design*
  - Should not format your hard-drive or let you gain elevated privileges, but could loop forever
  - How OSL behaves is highly dependent on how it is integrated

How to Get Involved

- TSC meetings are every other Thursday at 2pm Pacific Time
  - https://calendar.openshadinglanguage.org/
- Slack channel
  - https://academysoftwarefdn.slack.com/
  - #openshadinglanguage
- Github
  - https://openshadinglanguage.org