About Myself

Sony Pictures Imageworks
OpenColorIO
Katana
IIF / ACES
What makes color management hard?

- Context-Driven Requirements
  (Animation / VFX / Film / TV)

- Many schools of thought

- Varied Client Requirements

- Complex Software Ecosystem
OpenColorIO Motivation

- Consistent Image Display
- Consistent Color Transforms
OpenColorIO Motivation

- Roots in visual effects
- But... potentially useful across full digital motion picture workflow
System Architecture

[Diagram showing the system architecture with OpenColorIO at the center, connected to four client apps, and user color configuration at the bottom.]
OpenColorIO: Batteries Included

- Core Library (C++)
- Example ‘real world’ color workflows (including IIF)
- Plugins for 3rd party apps
- Command-Line tools
- Multi-Platform: (Linux, OSX, Windows)
Workflow

**OCIO**: color transform building blocks.

**User**: color workflow configuration.
(Glue to link transforms together)

**3rd party apps**: matched results
History

Started in 2003 at Sony Imageworks

Supports user-defined color workflows: vfx, anim, iif

Already used on dozens of motion pictures:
OCIO Details
OCIO::Transform

- Common LUT formats / flavors
- Simple Math: (ASC-CDL, matrix, gamma, log, exponent, ...)
- ICC Export (LittleCMS)
- Truelight Compatibility (Optional at compile time)

Additional transforms will be added as needed.
• Provided at runtime, $OCIO$ environment variable

• Defines all color conversions client apps may use

• Provides named ColorSpace

• Defines output display devices
OCIO::Config

(free at opencolorio.org)

• VFX Config
• Animation Config
• IIF Config
• Nuke-default
OCIO Image Processing

• CPU / GPU Support

• Native f32 color processing

• Native HDR Support
  (explicit dynamic range allocation control)

• Per-Shot LUT / Look Support
OCIO 3rd Party Support

Native Integration

- Nuke 6.5
- Katana 1.0
- Mari 1.3
- Silhouette
- OpenImageIO
- Python

Export Compatibility

- Autodesk Apps (Flame, Lustre)
- Photoshop (ICC)
- Houdini (HDL)
- Most color correctors (all other apps which read 3DL, CSP, ICC)

More, coming soon.
OCIO 3rd Party Support

In discussions with hardware vendors about on-set integration.
Open Source Development Process
Github Contributors

imageworks / OpenColorIO
- markfickett / OpenColorIO
- malcolmhumphreys / OpenColorIO
- jeremyselan / OpenColorIO
- est77 / OpenColorIO
- jsiomka / OpenColorIO
- jrray / OpenColorIO
- oliverfarkas / OpenColorIO
- petecrossley / OpenColorIO
- dbr / OpenColorIO
- sqlboy / OpenColorIO
- rexbron / OpenColorIO
- heesoo / OpenColorIO
On Being Naked

• We really do our development in public
  (checkins, code reviews, discussions, issues)

• We’ve made some embarrassing mistakes

• Integrity, not fear

• Airing dirty laundry breeds trust amongst community
Top OpenColorIO Issues

Need Better App Support
(Maya, RV, Houdini)

Documentation Lacking

Installation is complex

Currently tailored to large facilities
(display profiling, etc)
Open Source in Visual Effects

- Large Studios now contributing to Open Source (SPI / Disney / Weta / ILM / Double Negative / etc.)

- Notable projects:
  - OpenEXR (1.0 / 2.0)
  - Alembic
  - PTEX
  - OpenImageIO
  - OpenColorIO
  - OpenShadingLanguage (OSL)
For More Info

opencolorio.org

opensource.imageworks.com

Sign up for our public mailing-list!
Acknowledgements

Imageworks

Rob Bredow, Joseph Slomka, Sean Looper

Open Source Contributors

The Foundry, Malcolm Humphreys (DNeg), Oliver Farkas, Ben Dickson (Rising Sun)

And all of our users!
QUESTIONS? / DEMO

Jeremy Selan
Sony Pictures Imageworks
System Architecture

Client App
Client App
Client App
Client App

OpenColorIO

User Color Configuration
Color Configurations

(download from: opencolorio.org)

- VFX Config
- Animation Config
- IIF Config
- Nuke-default
Client Delivery

Pick appropriate OCIO config.
(Or, make your own).

Delivery implies color workflow.

• How will images be viewed downstream?
• Will delivery have view lut baked in?
  (Hopefully, not)
Image Display

**Identify your reference ‘gold standard’ display.**

- All apps must be judged validated relative to this environment (or not used for color judgements)
- Apps must take image color space into account
- Display transform is device-specific (after calibration)
Color Configurations

(download from: opencolorio.org)

- VFX Config
- Animation Config
- IIF Config
- Nuke-default
End to end process must be no-op.

Input / Output plate conversions are simple (1D) and perfectly invertible.

Image visualization does heavy lifting (3D)
VFX Color Config

Linear working space is High dynamic range, scene referred.

Middle gray at 0.18. Most pixel values -8 to +8 stops
Film scan linearization based on ‘generic’ acquisition stock.

Middle gray at 0.18 (445)

\( \sim 78 \) code values / stop in linear portion

Toe at -5.5 stops (under gray)

Shoulder at +7 stops (over gray)
Animation Color Config

1-D approximation of filmic tone rendering

High dynamic range linear

Output Referred (Rec. 709 Primaries)

Conversion to display (DLP/sRGB) is simple (1D/MTX)
ACES (IIF) Color Config

- Created by AMPAS
- Standardized EXR space (ACES)
- Currently in Beta
ACES (IIF) Color Config

- Standardized viewing transform (RRT)

- **ColorSpace(s)** defined for common input devices

- Does not run CTL live

colorspsaces:
  - !<ColorSpace>
    name: lnf
    bitdepth: 32f
    allocation: lg2
    allocationvars: [-15, 6]
  - !<ColorSpace>
    name: lg10
    bitdepth: 10ui
    allocation: uniform
    allocationvars: [0, 1]
    to_reference: !<FileTransform> {src: lg10.csp}
roles:
  compositing_log: lgf
  scene_linear: lnf

displays:
  sRGB:
    - !<View> {name: Film, cs: srgb10}
    - !<View> {name: Raw, cs: nc10}
    - !<View> {name: Log, cs: lg10}
  DCIP3:
    - !<View> {name: Film, cs: dlpRGB10}
    - !<View> {name: Raw, cs: nc10}
    - !<View> {name: Log, cs: lg10}
roles:
  compositing_log: lgf
  scene_linear: lnf

displays:
  sRGB:
    - !<View> {name: Film, cs: srgb10}
    - !<View> {name: Raw,  cs: nc10}
    - !<View> {name: Log,  cs: lg10}
  DCIP3:
    - !<View> {name: Film, cs: dlprgb10}
    - !<View> {name: Raw,  cs: nc10}
    - !<View> {name: Log,  cs: lg10}
Rendering

Linear in / Linear out.  
(shading / illumination in linear)

OpenEXR out (16 vs 32bit)

Ideally, no color space conversions at render-time
## Rendering

2 common approaches to linear:

<table>
<thead>
<tr>
<th>Linearized Output Referred</th>
<th>Scene Referred “HDR”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixels proportional to display device / reflected screen luminance.</td>
<td>Pixels proportional to light in original scene.</td>
</tr>
<tr>
<td>Limited dynamic range</td>
<td>Useful in physically based rendering</td>
</tr>
</tbody>
</table>

- Linearized output-referred (gamma 2.2, etc.)
- HDR Scene-referred (0.18 middle gray, max white undefined)

**Inf** (HDR scene-linear, float)
Rendering

Requires tonemapping (S-Curve) for Display.

<table>
<thead>
<tr>
<th>Linearized Output Referred</th>
<th>Scene Referred “HDR”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixels proportional to display device / reflected screen luminance.</td>
<td>Pixels proportional to light in original scene.</td>
</tr>
<tr>
<td>Limited dynamic range</td>
<td>Useful in physically based rendering</td>
</tr>
</tbody>
</table>

Inf (HDR scene-linear, float)
CLOUDY WITH A CHANCE OF MEATBALLS
Texturing

- Photoshop (ocio2icc)
- Mari (native)
- Others?

Ideally WYSIWYG.
Texturing

Typically preferable to paint directly in monitor space

(i.e, inverse display transform)

(Of course, not for control maps)

<table>
<thead>
<tr>
<th>dt8</th>
<th>diffuse texture, uint8</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp8</td>
<td>matte painting, uint8</td>
</tr>
</tbody>
</table>
Texturing

Diffuse Modulation Textures

[0-255] maps to [0.0, 1.0]
Matte Painting

[0-255] maps to [0.0, >> 1.0]
Texturing

When to linearize?

• In the shader (coming soon to OIIO)

• Prior to mipmapping

OpenImageIO:  `maketx -colorconvert dt8 Inf`
Compositing

Linear renders in.

Plates in (device color space)

Delivery often in device color space (coming soon - float exr delivery using IIF’s ACES)

Plate handling requires perfect invertibility

Inf (HDR scene-linear, float)
Compositing

Ignore built-in compositor LUT handling

Use **OCIOColorSpace**, **OCIOLogConvert**

Ignore built-in compositor image viewing

Use **OCIODisplay**
Digital Intermediate

- Working color space for grade
- Visualization
- Output trim passes
Color Pipeline Gotchas

- Dealing with Plate Timing (working neutral)
- Not all ‘log’ spaces are equal
- HD Flavors (Full Range vs. Headroom)
- Gamma 2.2, 2.4, 2.6
- 16-bit int vs. 16-bit float roundtrip conversions
For More Info

opencolorio.org

opensource.imageworks.com

Sign up for our public mailing-list!
What applications do you use?