

USD Camera

Resource Links:

[LabCamera](#) concept

[Camera and Lens Metadata Standard](#) /i

[Eric Reinecke's example of OTIO adapter for Lens/Camera metadata](#)

[Original \(YAML\) output example from Cooke Lens + Camera system](#)

[David Stump's "Commandante" workflow project presentation from the Real Time Conference](#)

[VES Tech - Comparison of DCC Camera parameters](#)

[VES Tech - Comparison of camera metadata](#)

Introduction:

Solving for a common Camera

Summary

This document describes a proposal to create a unified Camera schema which simplifies workflows in computer generated imagery (CGI).

Problem Statement

A fundamental requirement for Computer Generated imagery is that there is a viewport into the image. The Camera, as it is often called. Due to technical limitations, the process of CGI has historically relied on compartmentalized efforts that are often disconnected. Assets and elements of the work are handed off between disciplines, much like virtual paperwork shuffled between departments. This disconnectedness has resulted in highly-specialized software that optimizes the efforts within a particular discipline. However, this makes it extremely challenging to create a continuous, consistent viewport through which to work. One application's methods to establish a Camera may be very different from the another's. As a result, there is additional complexity added to the workflow in setting up a Camera Rig asset that has features and attributes that can be shared among the departments. Each application then needs engineering efforts to be able to interpret the Camera Rig into its own set of Camera solutions. Any changes to the Camera Rig settings or attributes requires a cascade of re-interpretations into the various departments. As well, the modern Camera Rig used in VFX or Animation productions have highly complex data sets that may or may not be transportable between applications, sometimes requiring significant engineering effort.

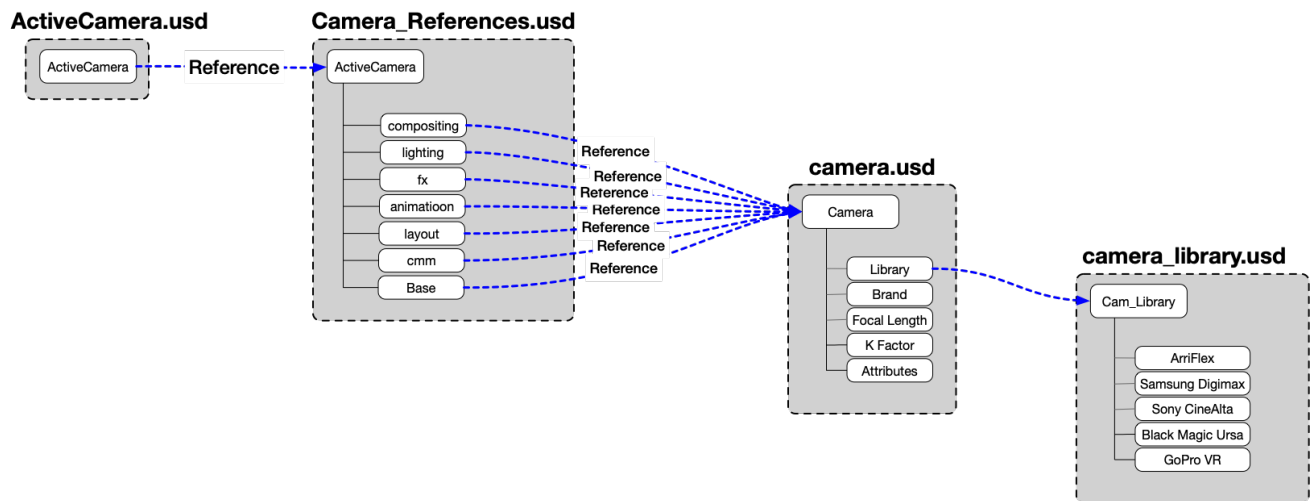
Solution Statement

A recent open-source innovation in the CGI industry is known as Universal Scene Description (USD). The added benefits of USD includes methods to further organize and reference structures to increase functionality. USD is rapidly becoming a standard method of data interchange between various application platforms. As well, USD is helping to conform disparate workflows among disciplines by creating a singular source of truth in the data model. There is a unique opportunity to leverage these considerations to unify the Camera and its constituent dependencies. Applications that are leveraging USD will be able to pick up the Camera attributes directly, using inherent workflows. The cascade of interpretive pipeline dependencies on proprietary Camera rigs would be obsolesced.

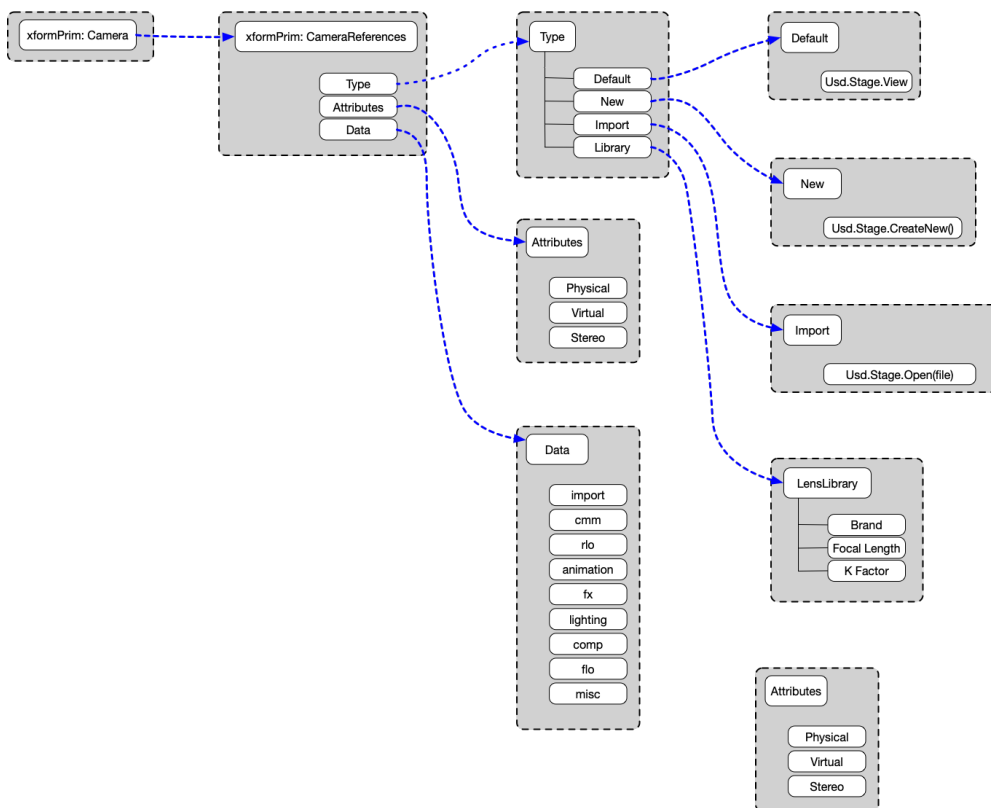
Schema Examples

These are a few sample schemas that show how one might create Camera abstractions using base USD referencing capabilities. These are purely examples, and not meant to represent any fixed methodology in representing a USD Camera schema. As we all know, there is an Stage camera but having a tangible, functional Camera that we can structure and share has real meaning and value for productions. In this diagram, there is a working camera that is one abstract layer above the various camera roles that are then references to the base USD Camera. Departmental cameras may be overloaded, etc.. with specific discipline-focused attributes.

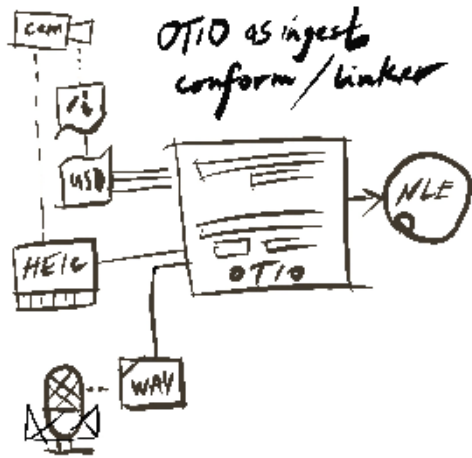
Ex. 1



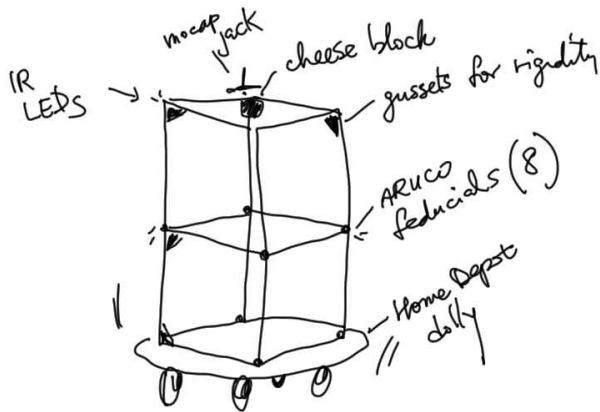
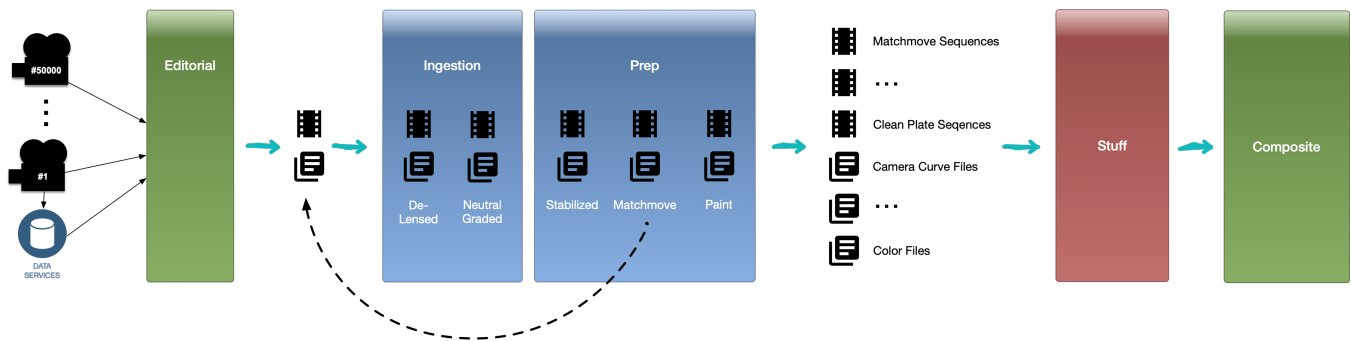
Ex. 2



Possible Role of OpenTimelineIO in a physical camera data transport pipeline



Essential Workflow of Camera/Lens Data from Set through to VFX Pipeline; Highly Simplified and Pipelines Vary:



Calibration gadget. vertices are spaced at exactly 1m joined by aluminum extrusion