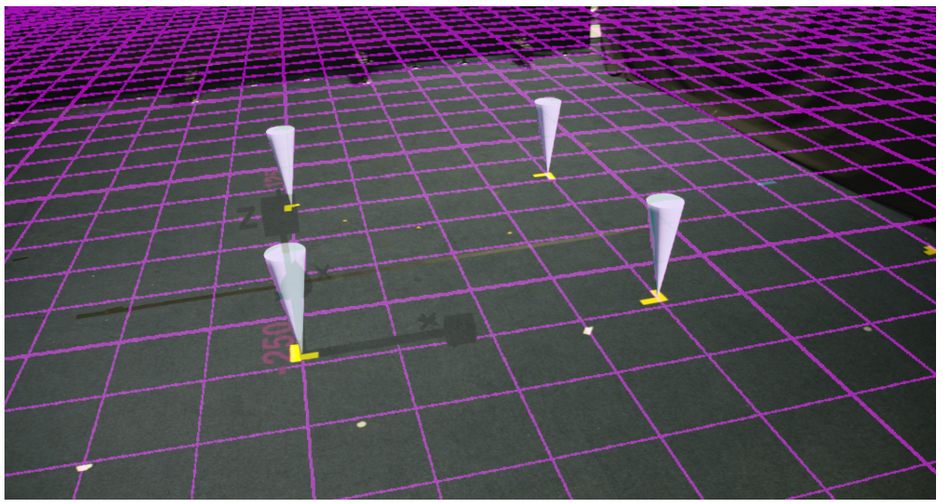
VFX Sample Bundle Manual - 5.3.2.2



Nuke Final Composite Video

1. Introduction

1.1. Purpose

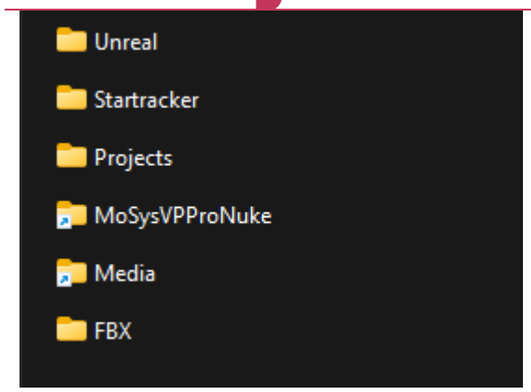
The purpose of this manual is to describe the VFX Sample Bundle. Once tracking data has been captured live there is a desire to bring this into post workflows. This allows us to rerender in DCC programs such as Maya (Projects/MayaSample.mb). Apart from having the camera track, the correct lens distortion must also be applied (Projects/NukeSample.nk). We can compare this higher resolution result to the original Previz (Media/Previz/vfx0315-05-31-010000.mov).

Rendered frames are normalised to start at frame 1001 with 10 handles on either side. The resolution of all projects in this bundle is UHD (3840x2160) at a frame rate of 25. The original Unreal Previz was done at HD so we upscale to match the camera footage. This is a typical use case as we often don't have the performance budget in Unreal to output higher than HD while live.

Note: The bundle will work with any resolution/frame rate.

1.2 Overview

The bundle comes with the following folders. Here we will briefly describe the contents of each.



Unreal

This contains the Unreal Engine project that was used to record the data and capture the Previz (Media/Previz/vfx0315-05-31-010000.mov). It is also used to export the FBX and render an EXR.

Startracker

Startracker can record the tracking data as a .f4 file. This can be imported into Unreal via the Take Importer. See the VP Pro manual for these steps. StarTracker is capable of high speed recordings and is good to have as a backup. Having access to them therefore provides a different route to get your tracking data into FBX.

Projects

Sample projects for each of the programs covered by this bundle. We can use Maya or Houdini for rerendering the 3D scene with our camera track. Nuke is used to reapply lens distortion. A DaVinchi Resolve project shows an EDL with our desired VFX Clip, Nuke composite (with distortion) along with the lineup of the Unreal render. See for more.

MoSysVPProNuke

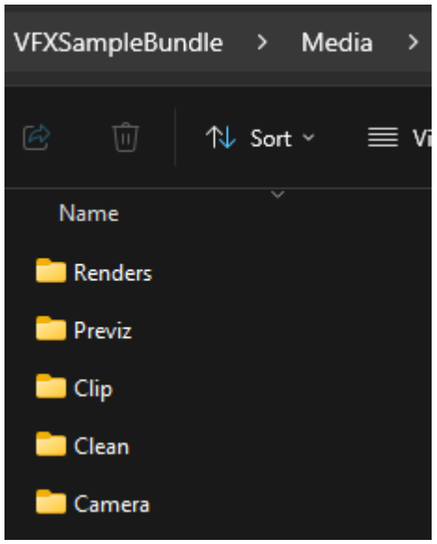
This contains the Mo-Sys Nuke Plugin for various different versions of Nuke. See for instructions on how to install this for use with your Nuke.

The plugin adds the ability for us to take a numerical approach to generate distort/undistort STMaps in Nuke based on lens data from the FBX file. The Unreal Mo-Sys VP Pro Plugin is capable of rendering EXR files with STMap layers (named MMoSysDistortionSTMap_ and MMoSysUndistortionSTMap_) but there are some scenarios where a numerical approach is desirable (e.g retimed shots, needing lens data to be baked in) and a render is unnecessary.

Media

A number of different clips and renders are contained in this folder. This folder comes as a separate VFXSampleBundleMedia download due to the large size of the files (70GB+). Once downloaded it should be dragged into the top of the _VFXSampleBundle folder as shown. Paths

in Nuke have been set to be relative to the Nuke script. If this folder is placed elsewhere, those paths need to be updated accordingly.



Renders

- An overscanned Unreal EXR render with STMap + Calibration layers. Normalised to start at frame 1001 with 10 frames of handles. **Note:** Only every 10th frame is rendered due to very large file sizes and the example clip being 1200 frames long.
- 3D content renders (without distortion) using tracked camera transform with a Houdini flipbook and a Maya playblast. Field of View and Sensor Size are also appropiately set and animated by the FBX import.
- A final Composite that was first rendered out of the end of Nuke pipeline. Converted to MP4 to reduce the file size for ease of sharing.

Previz

The AR output of the Unreal Engine recorded on by a SDI feed into a Blackmagic Hyperdeck recorder when the clip was originally shot and recorded. This is used as a reference to show we can rerender with external tools and match position, distortion and other lens parameters (FOV).

Clip

This has a 1 clip EDL which is the original camera clip cut down to 1200 frames. The first frame of this should match frame 1001 of the Unreal EXR render and the animations in the FBX.

Clean

Camera footage as recorded by a SDI feed into a Blackmagic Hyperdeck recorder.

Camera

Recorded camera clip directly on the camera.

Note: A Blackmagic G2 4.6k was used for this bundle. Recorded at UHD with a Sensor Size of 25.34mm x 14.25mm.

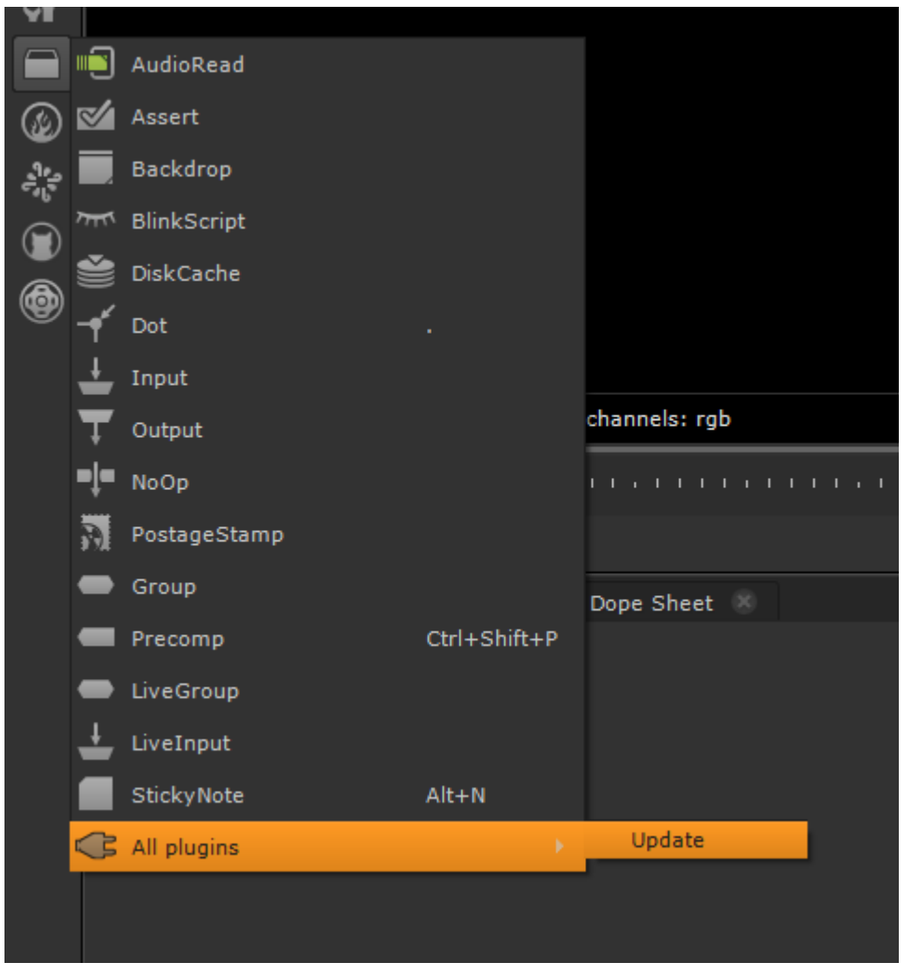
FBX

The FBX exported from the VFX Unreal project. Normalised to 1001 with 10 frame handles. The name of FBX is automatically set to the LevelSequence with the MoSysCamera it contains postfixed. In order to reliably transfer the Lens Distortion data between 3D packages, the parameters are embedded in additional unreal actors (Lens Actors) in their transform. Also included is a separate FBX of the cones used during the recording and visible in the AR Previz. This is a way to put the same 3D content into Houdini and Maya. A LevelSequence (LSCones_) was created in Unreal in order to export this.

3. MoSysNukePlugin

3.1 Installation

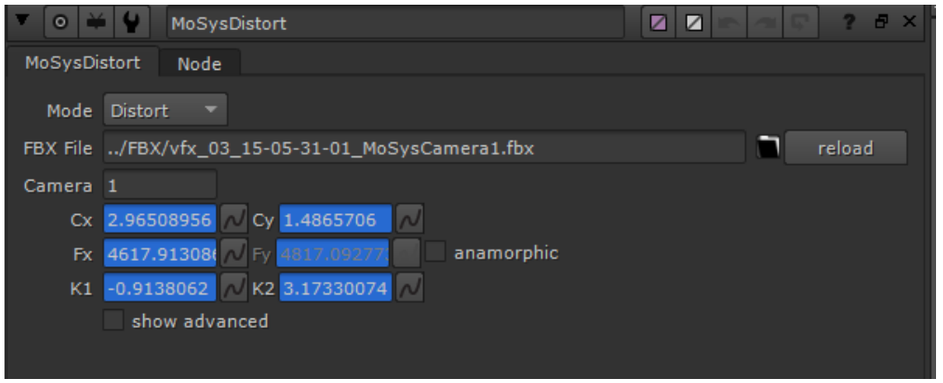
The Mo-Sys Nuke Plugin is available from here. This includes various Nuke versions for both Windows and Linux. If specific versions are required then please contact support@mo-sys.com. Once downloaded, the appropiate MoSysDistort.dll (Windows) or MoSysDistort.so (Linux) should be copied to the plugins folder in the Nuke installation directory, e.g C:\Program Files\Nuke14.0v5\plugins. After installation you must Update the node list within Nuke.



3.2 Usage

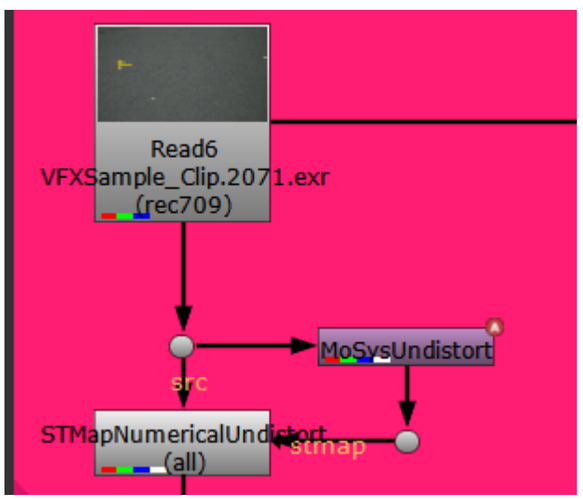
The MoSysDistort node is responsible for generating either a distort or undistort STMap from the Lens data in the FBX.

Properties

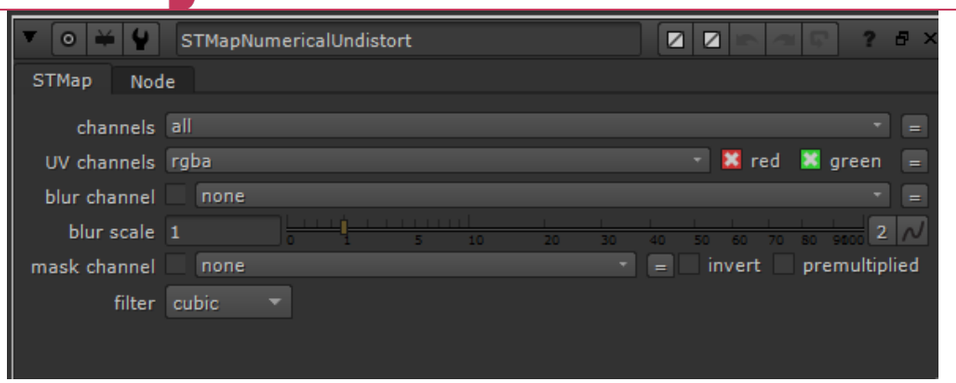


- Select Distort/Undistort mode.
- Browse to the FBX containing MoSysCamera tracking and lens data.
- Set the Camera field to the index of the MoSysCamera. Only relevant if multiple cameras are exported to one FBX. The default export behaviour from Unreal is one FBX per camera.
- Click reload to load the Lens Parameters from the Lens Actors of the FBX file and bake the animation into this Node.

Node Graph



- The input to the MoSysUndistort node in this example is the plate to Undistort. This input is used to set the format of the resulting STMap appropiately.
- The output contains the STMap in the RG channels.
- Use a Nuke STMap node to distort input based on the RG channels.



4. Unreal

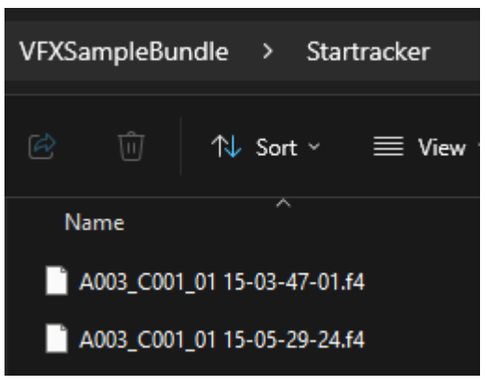
Please see the latest VP Pro manual for a guide to the operation of the Mo-Sys VP Pro Plugin for Unreal. This section specically references the more relative features.

4.1 Recording

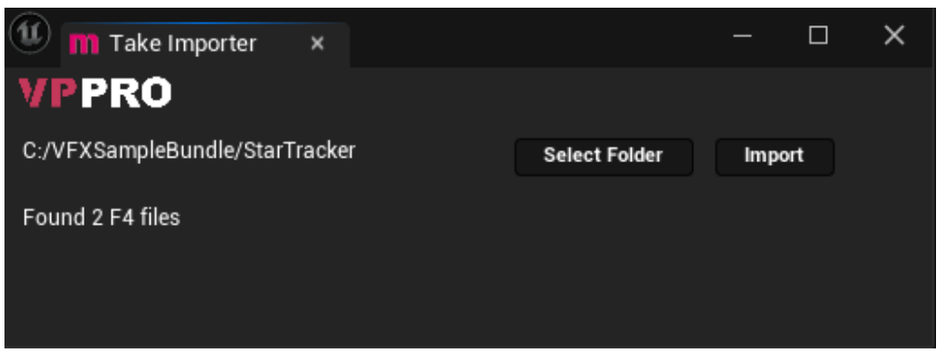
Record as per the VP Pro manual. Ensure each MoSysCamera ends with a digit e.g MoSysCamera1. This is so that when we export an FBX from the recorded data later, we can properly match the lens data to each camera. Unreal camera properties, such as Aperture and Sensor Width/Height, will be recorded so be sure to set these to the appropiate values. Once recording is done, there will be an Unreal LevelSequence output to the /Game/Takes/ folder.

4.2 Importing/Exporting

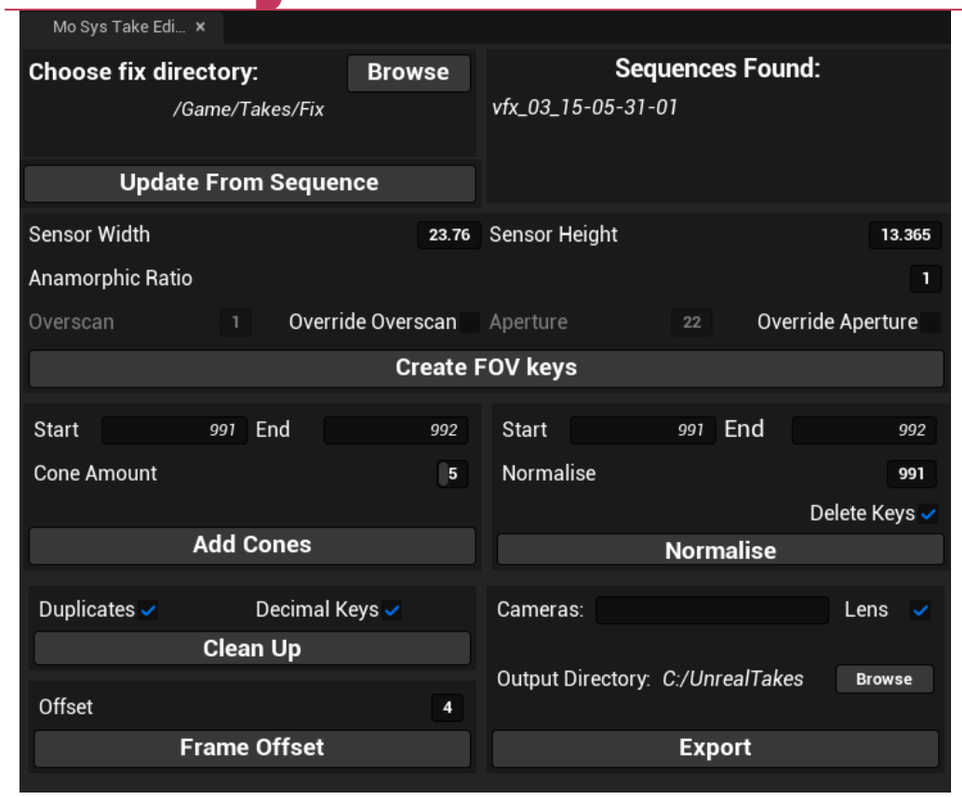
It is also possible to produce an Unreal LevelSequence by importing the tracking data from a .f4 file. This is the file type recorded on a Mo-Sys StarTracker when it has recording is enabled.



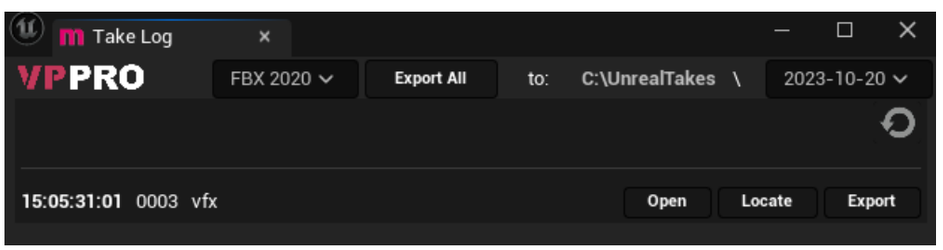
The Mo-Sys Take Importer Panel provided by the plugin allows you to specify a folder containing these F4 files and bring them into Unreal.



If required, you can use the Mo-Sys Take Editor Panel to adjust various parameters of the LevelSequence after recording (Sensor Size, Aperture).



An FBX can be exported using the Mo-Sys Take Log Panel. This FBX can now be brought into programs such as Maya and Nuke as demonstrated by the sample projects in the Projects folder.



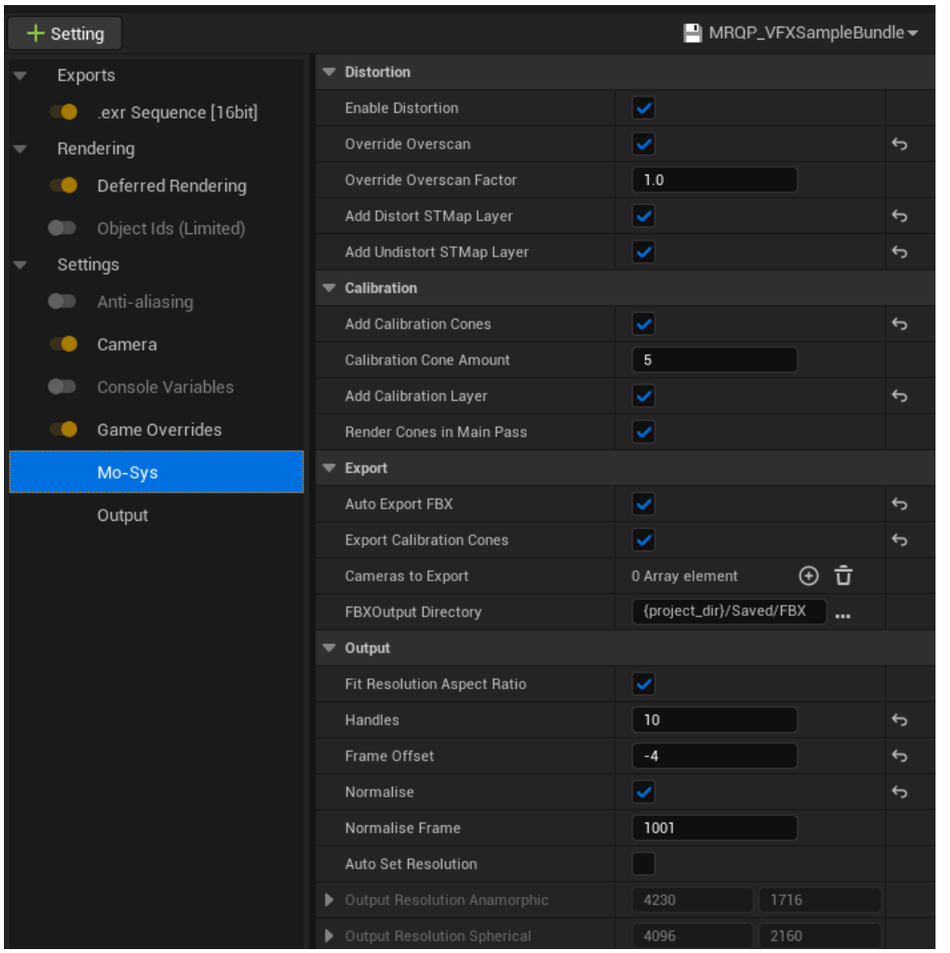
Note: The Mo-Sys settings in the _Movie Render Queue are also capable of exporting an FBX.

4.3 Rendering

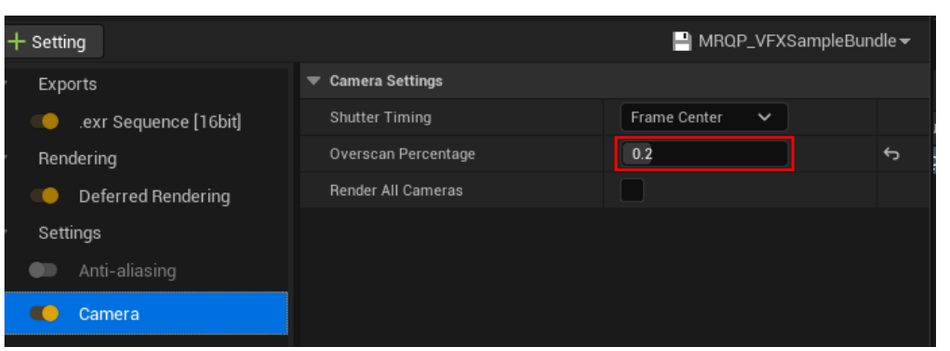
You can render recorded takes (LevelSequences) with Unreal's Movie Render Queue. The Unreal Mo-Sys VP Pro Plugin comes with some MRQ presets to showcase different quality levels and features. The MRQPVFXSampleBundle_ preset is what is used to produce the Unreal render used in the bundle. The Mo-Sys setting has many useful options:

- Adding the STMap Distort/Undistort layers.
- Adding Cone Calibration Layer.
- Auto exporting an FBX.
- Normalise frame number and Handles.

- Frame Offset constant used to match the right recorded tracking frame to a camera frame. Accounts for all delays in the chain from recording on the camera to what frame Unreal actually starts recording. Once found for your particular case, it does not need to be changed. We can then simply do an auto-align in the Resolve project to match the Unreal rerender to the VFX clip.



Of particular interest are the Unreal Camera settings. This is used to add overscan to the output render. A value of 0.2 means the render is scaled up by 20% for a border of 10% around the edges. When rendering EXR files this takes advantage of their property to have separate data and display windows. The result is that the overscanned pixels are hidden outside the display window. In the sample Maya/Houdini projects it will be shown how we also overscan them by this same amount.



NearTime is a cloud based system to enable automatic re-rendering of recorded takes at higher quality than in real time. This can prove very useful when dealing with many Unreal recorded LevelSequences. See the VP Pro manual for more information on its operation. If interested in using this system, please contact sales@mo-sys.com.

5. Projects

Sample projects that show:

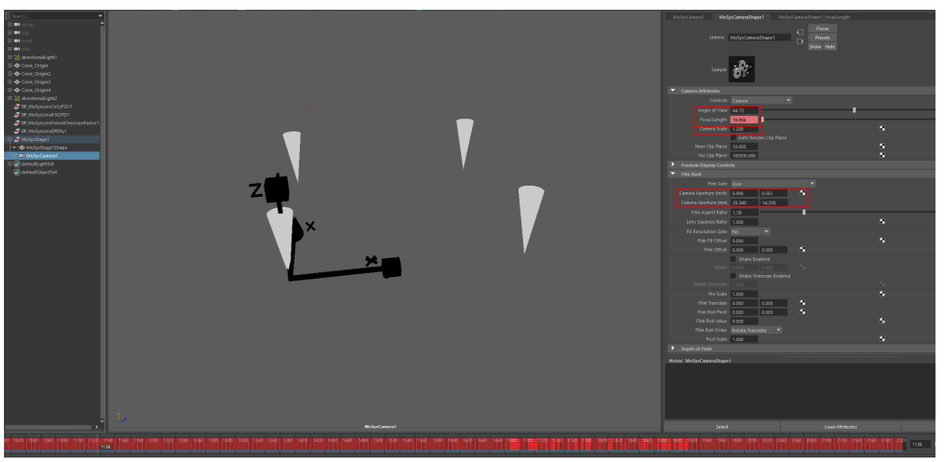
- How to render 3D content rendered in Houdini/Maya using the camera track from the exported FBX.
- How to undistort/distort in Nuke using the Mo-Sys Nuke Plugin.
- Calibration and verification of the data and lineup (Nuke/Resolve).

5.1 Maya

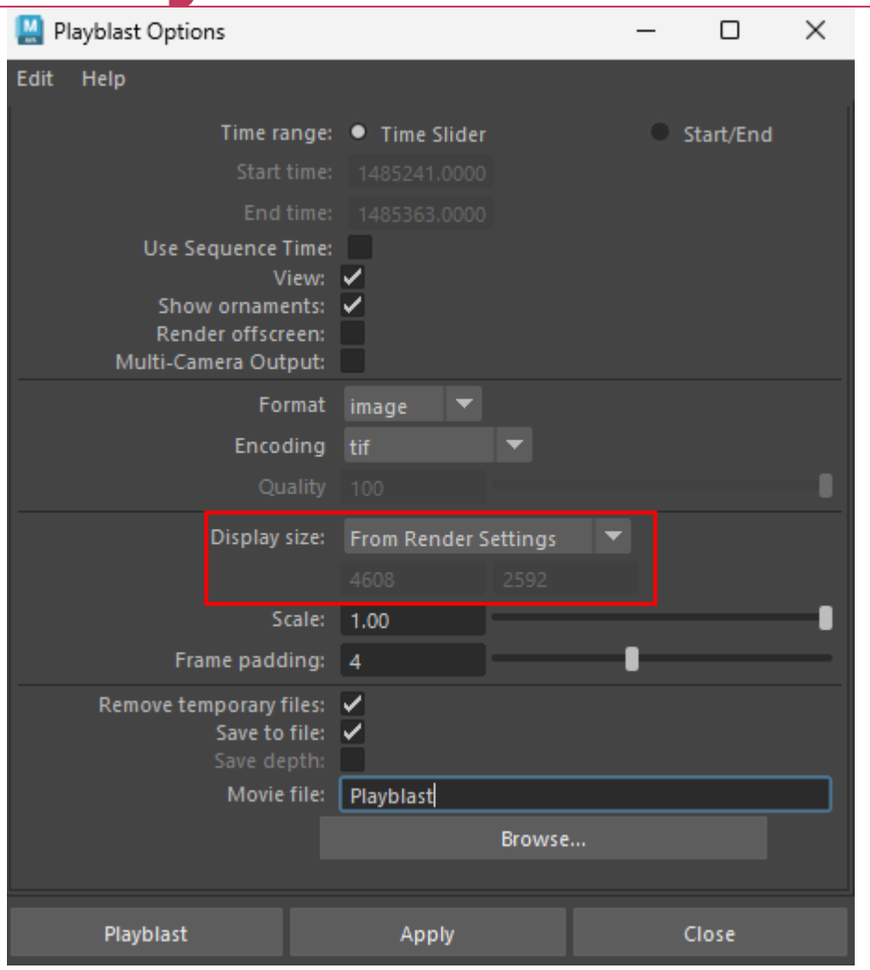
A summary of how this Maya project was created is as follows:

- FBX cones were imported for the 3D content.
- Recorded camera FBX was imported. This brings in the MoSysStage and MoSysCamera.
- Also imported with the camera FBX are Lens Actors which contain the Lens Parameters (K1, K2, etc.). The Lens Actors are only used in Nuke so they can be safely ignored in the Maya scene.
- Field of View (Focal Length) and Sensor Size (Camera Aperture) are set to the recorded values in the FBX.

- Camera Scale is adjusted to match our overscan percentage value used thoughout the bundle (20%).



- Render playblast with overscan by scaling the output resolution by the overscan (x1.2).



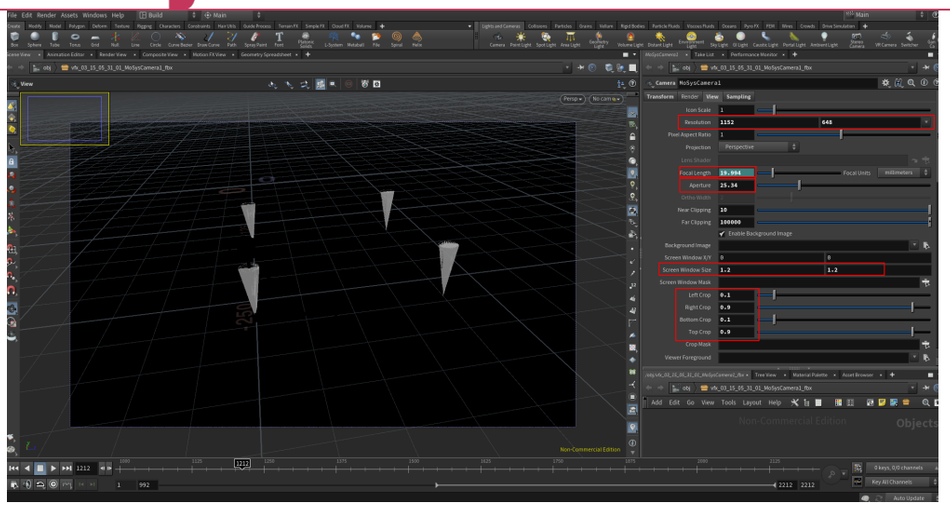
5.2 Houdini

A summary of how this Houdini project was created is as follows:

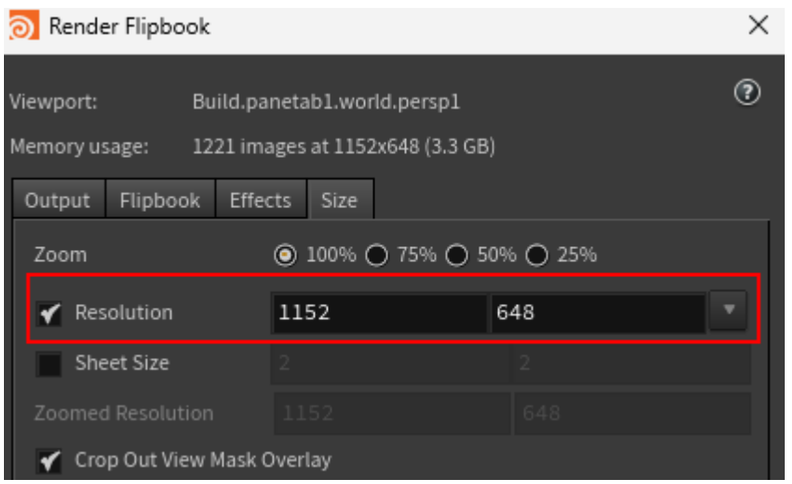
Note: Houdini limits the output render resolution to 1280x720. Therefore, instead of rendering at UHD (3840x2160) we use 960x540 which is 4 times smaller. In Nuke there is a Reformat node to scale back up by 4 to match the Maya workflow

- FBX cones were imported for the 3D content.
- Recorded camera FBX was imported. This brings in the MoSysStage and MoSysCamera.
- Also imported with the camera FBX are Lens Actors which contain the Lens Parameters (K1, K2, etc.). The Lens Actors are only used in Nuke so they can be safely ignored in the Houdini scene.
- Field of View (Focal Length) and Sensor Size (Camera Aperture) are set to the recorded values in the FBX.

- We adjust the Window Screen Size and Crop values to match our overscan percentage value used thoughout the bundle (20%).

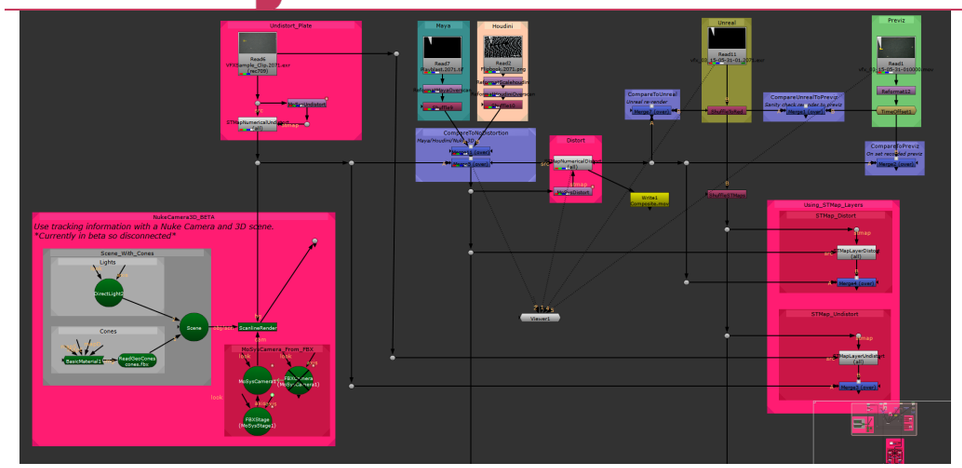


- Render flipbook with overscan by scaling the output resolution by the overscan (x1.2).



5.3 Nuke

The Nuke project showcases a workflow, from start to finish, to produce a composite using a render from Maya/Houdini and the Lens Parameters contained in the FBX. This flows left to right. Backdrop nodes within the script explain the pipeline. See for more details on the Mo-Sys Nuke Plugin.



Summary:

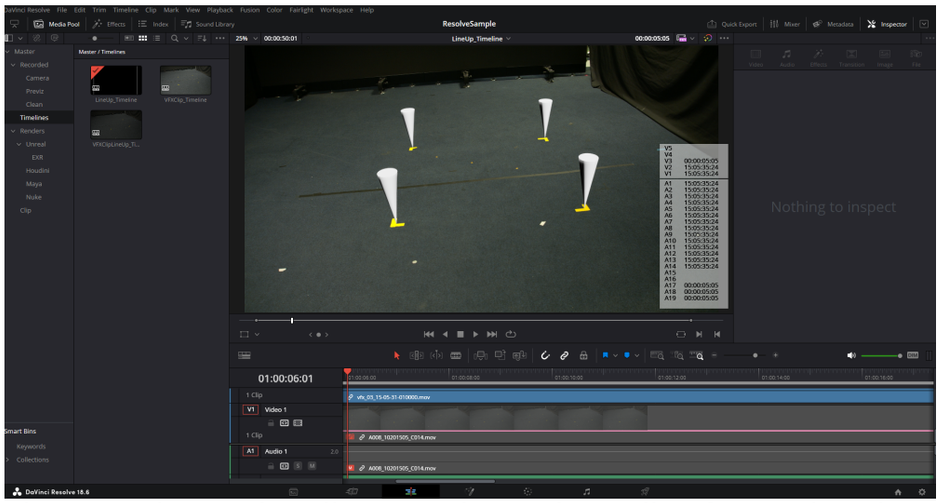
- Undistort the VFX clip (shortened Camera plate according to an EDL in Media/Clip/VFXSample.edl) using the MoSysDistort node provided by the Mo-Sys Nuke Plugin.
- Read overscanned Maya/Houdini and Reformat while keeping bounding box.
- Overlay over undistorted plate.
- Distort all together using the MoSysDistort node.
- UsingSTMapLayers backdrop section demonstrates the same result is achieved using the STMap layers contained within our Unreal EXR.
- The final section verifies our results by comparing our composite to the Unreal Render and the Previz.

Notes:

- The bottom left part of the script is a a 3D comp using a Nuke camera. This is currently in Beta and has been left disconnected as such.

- The bottom section marked as Testing and Calibration is a verification step only on the distortion data and is not needed in actual workflows. It uses the STMap layers from the EXR and those produced by the Mo-Sys Nuke Plugin to distort/undistort the cones in the Calibration Layer to confirm everything works as expected.

5.4 DaVinchi Resolve



The _Resolve project lines up the various clips for testing the results of our pipeline. There are a number of Timelines which serve different purposes:

LineUp_Timeline

This shows an overlay of the Camera, Clean and Previz Media clips.

VFXClip_Timeline

The VFXSample.edl was generated from this. It is a shortened version of the original Camera clip.

VFXClipLineUp_Timeline

This shows a lineup with an Unreal rendered MOV (so it has timecode) over the Camera clip. Auto-align on timecode can be done to match these.

Further along the timeline there is the shortened VFX Clip. Over this is the Unreal Rendered EXR (every 10th frame held for 10 frames). The render of the Nuke Composite.mp4 is also overlayed. This shows them all lining up and handles working as expected (we get 10 extra frames before and after VFX Clip video range).
Note: The Frame Offset option in the Mo-Sys setting part of the Movie Render Queue discussed in is crucial to getting everything to line up so auto align is possible.