Cesium 5.6

Reference Guide

September 2025

SINCE
DAY
ONE.

Table of Contents

WHAT’S IN THIS DOCUMENT

This document describes how to use features specific to CESIUM.

Help and support

For contact information or our online helpdesk, please visit our **support page** at **https://chyron.com//support/**.

Purpose and Terminology

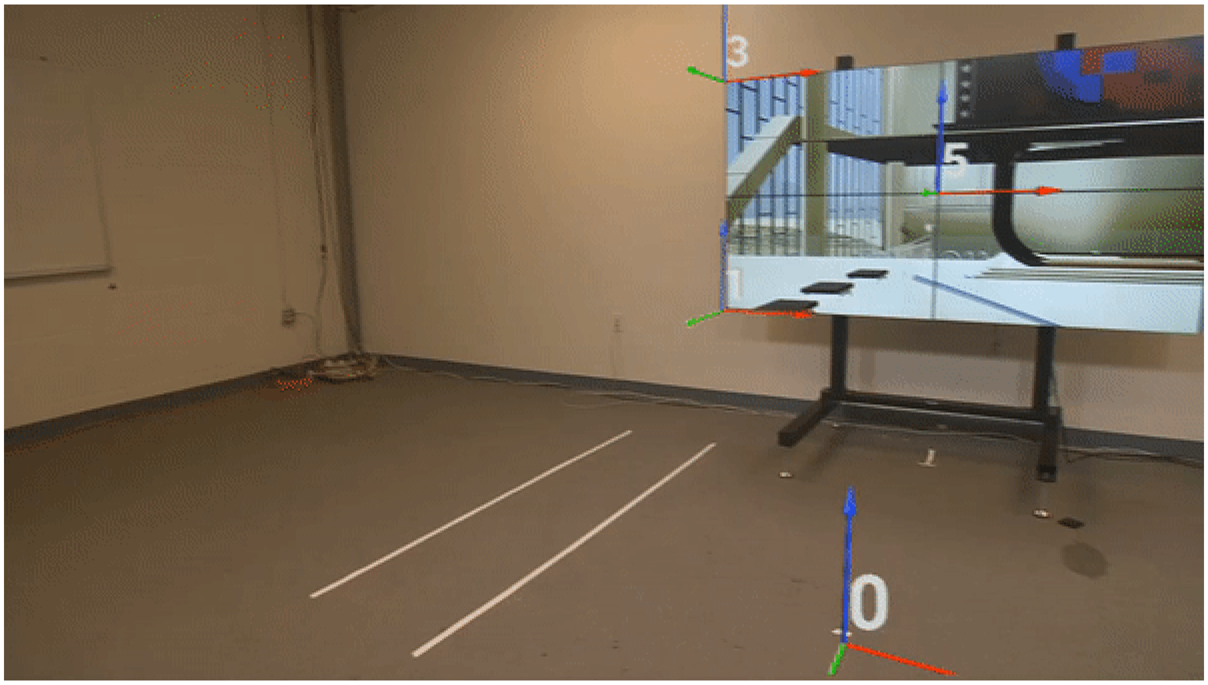
Overview

This document is a guide for using Cesium software.

Cesium is Chyron's application for managing tracking data from any providers and converting them into regular and normalized 3D camera parameters (coordinates and optical ones).

Cesium is a software that delivers cooked camera (geometrical and optical) data to a partner 3D software, for example PRIME VSAR or PRIME AR in a uniform way.

For this purpose information on camera data is gathered from different vendors (eg: Chyron robotics, NCam, Mo-Sys, Vinten, …) and then one or several camera data are computed and sent to the 3D software.

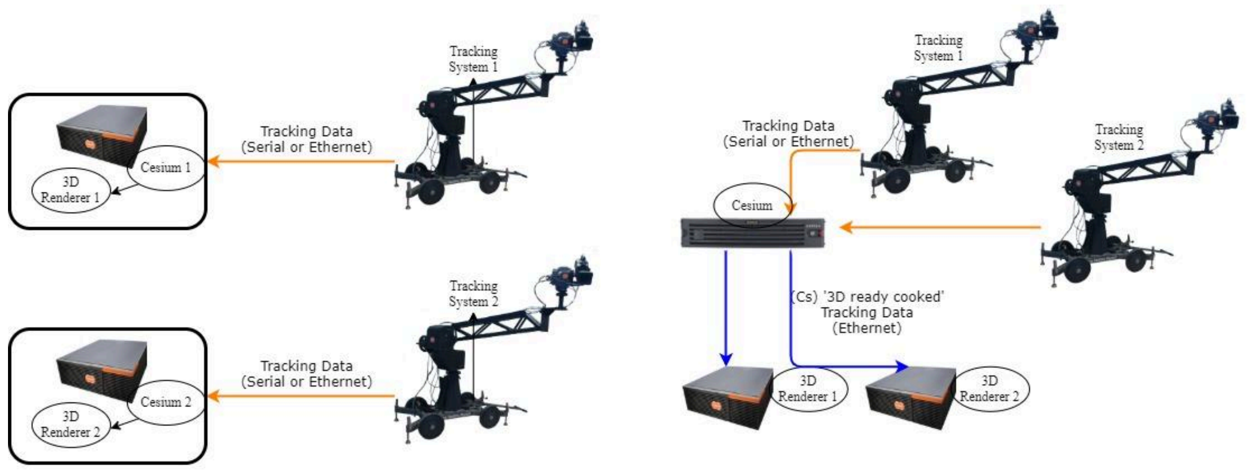


Global Operation

Cesium runs on a host computer connected to one or several camera tracking systems, providing real-time data of their own format, while cameras are moving.

From the data received, Cesium will compute instantly a set of parameters characterizing the virtual camera that is best fit to the actual real camera, including focal angle, and make that set continuously available to supported 3D application, such as Fresh/VSAR, **PRIME** This last can run either on the same computer or on any distant machine connected to it over Ethernet.

Many camera tracking systems are supported by Cesium, obviously - but not only, those provided by Chyron.


Tracking System 1
Tracking Data (Serial or Ethernet)
Cesium 1
3D Renderer 1
Tracking System 2
Tracking Data (Serial or Ethernet)
Cesium 2
3D Renderer 2
Tracking System 1
Tracking Data (Serial or Ethernet)
Tracking System 2
Cesium
(Cs) '3D ready cooked' Tracking Data (Ethernet)
3D Renderer 1
3D Renderer 2

Option 1
Local transmission between Cs and 3D app. (Several Cameras)

Option 2
Ethernet transmission between Cs and 3D app. (Several cameras)

Terminology

Driver

In Cesium terminology, a driver is an interface to a vendor's tracking solution. This driver gathers information from the specific device, usually a tracking camera device. It manages connections, protocols, as data reception and eventual normalization. When data is gathered from the driver, it is then processed by the Rig to produce cooked camera data. Cesium also introduces the notion of calibration of Rigs.

Rig

A Rig is the structure model that converts data from a driver into standardized 3D coordinates and optical info. It can be seen as an articulated camera system, but it can also be a big screen to achieve parallax effect. As such, it is modeled with a series of basic mechanical (sequentially organized/evaluated) and optical transformation steps. Each rig sends its information to the external 3D rendering software.

⚠️Please note that optical and mechanical transformations are performed by matrix multiplications, so they depend on the order in which they are present in the rig in Cesium, as matrix multiplication is not a commutative operation.

Targets

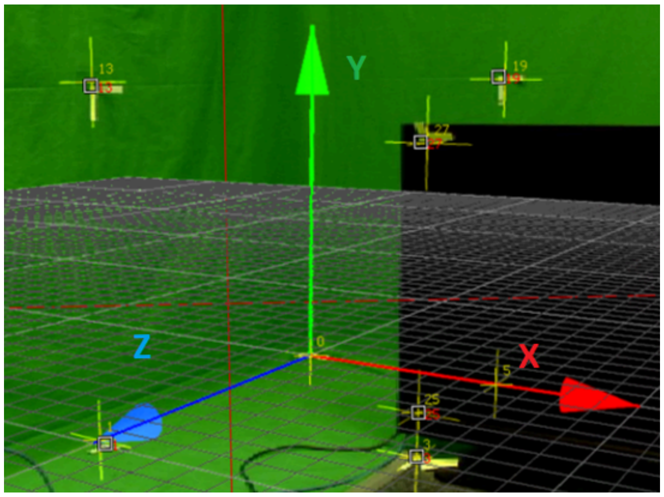
Targets are used for helping matching real and virtual. That is to say:

See in the 3D scene where real parts of the studio like a big screen, a real desk, the talent main position, etc are. This is very helpful for the positioning of 3D elements.

Control the quality of the calibration. When calibration is achieved the virtual and real points should match.

It consists of a list of known 3D points in space. A target has a label, and 3 known coordinates X,Y,Z. This target list defines a 3D space, with an absolute origin, and orientation. In Cesium the Y axis points upwards, X and Z axis are clockwise. Note that the 3D rendering engine may have different conventions.

Just to show you an example from our calibration:



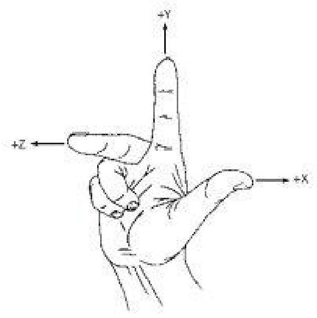
A target list is dedicated to a specific physical studio.

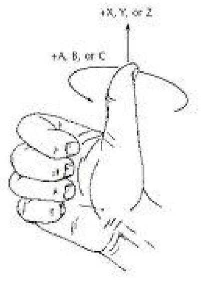
Calibration

Calibrating a Rig is finding all the parameters that makes this rig available in the 3D space defined by the target list. The calibration is the same for all the vendors, what may differ are the parameters to be found. (For every vendor you need to do the calibration again, for every lens as well. But these files can be saved and loaded afterwards and swapped as easily as the lens itself.)

Coordinate system orientation

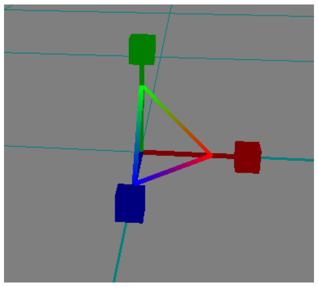
Cesium coordinates system orientation is right-handed, and Y-up.





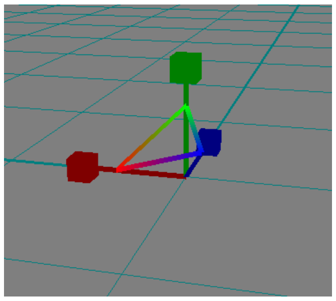
Convenient references may be:

1. (Pan angle of tracking device at 0 degree when facing the set)
    - X pointing to the right (Red)
    - Y up (Green)
    - Z going toward the view (Blue)



2. (Pan angle of tracking device at 180 degree when facing the set)

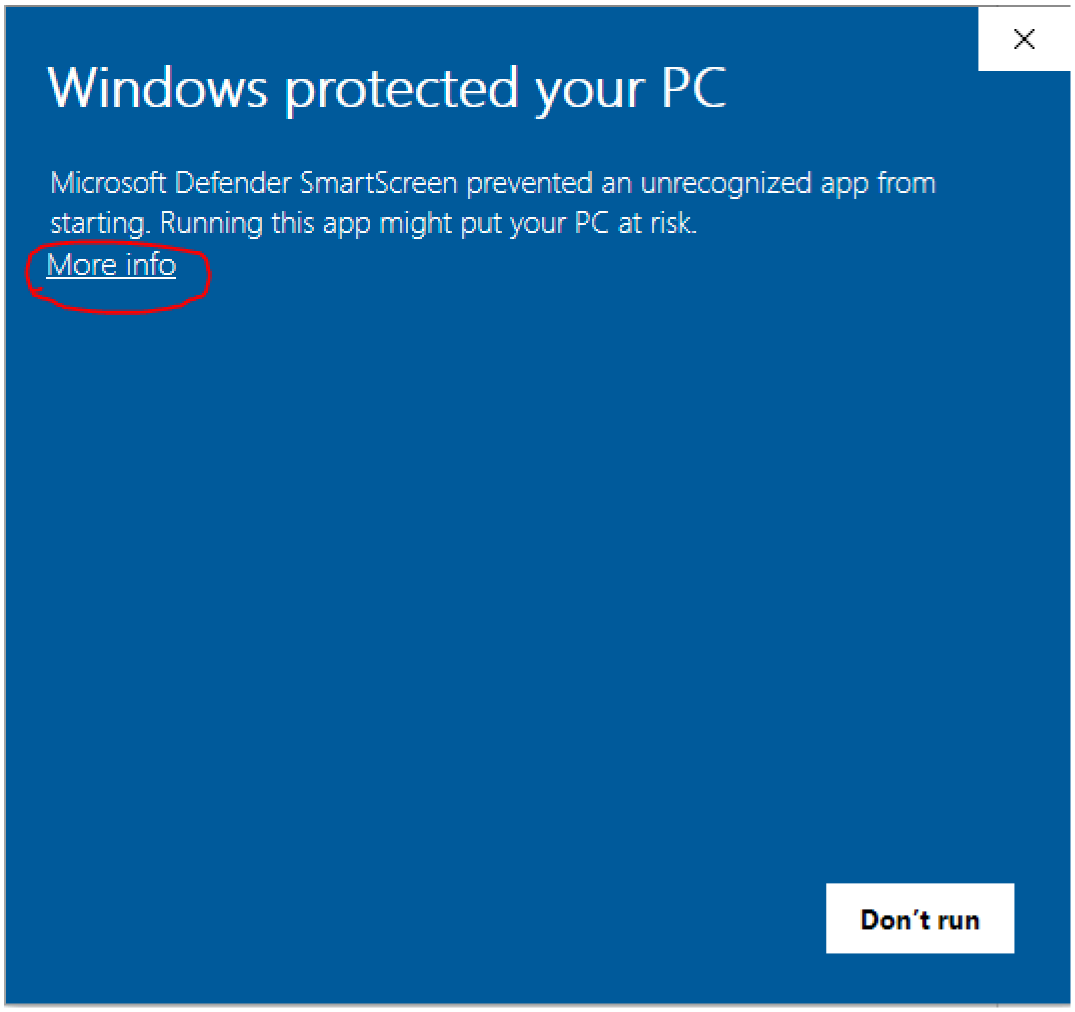
- X pointing to the left (Red)
- Y up (Green)
- Z going backward the view (Blue)



Installation

- Double click on the download Cesium-XXXX-Setup.exe file you received or downloaded. XXXX stands for your version number.

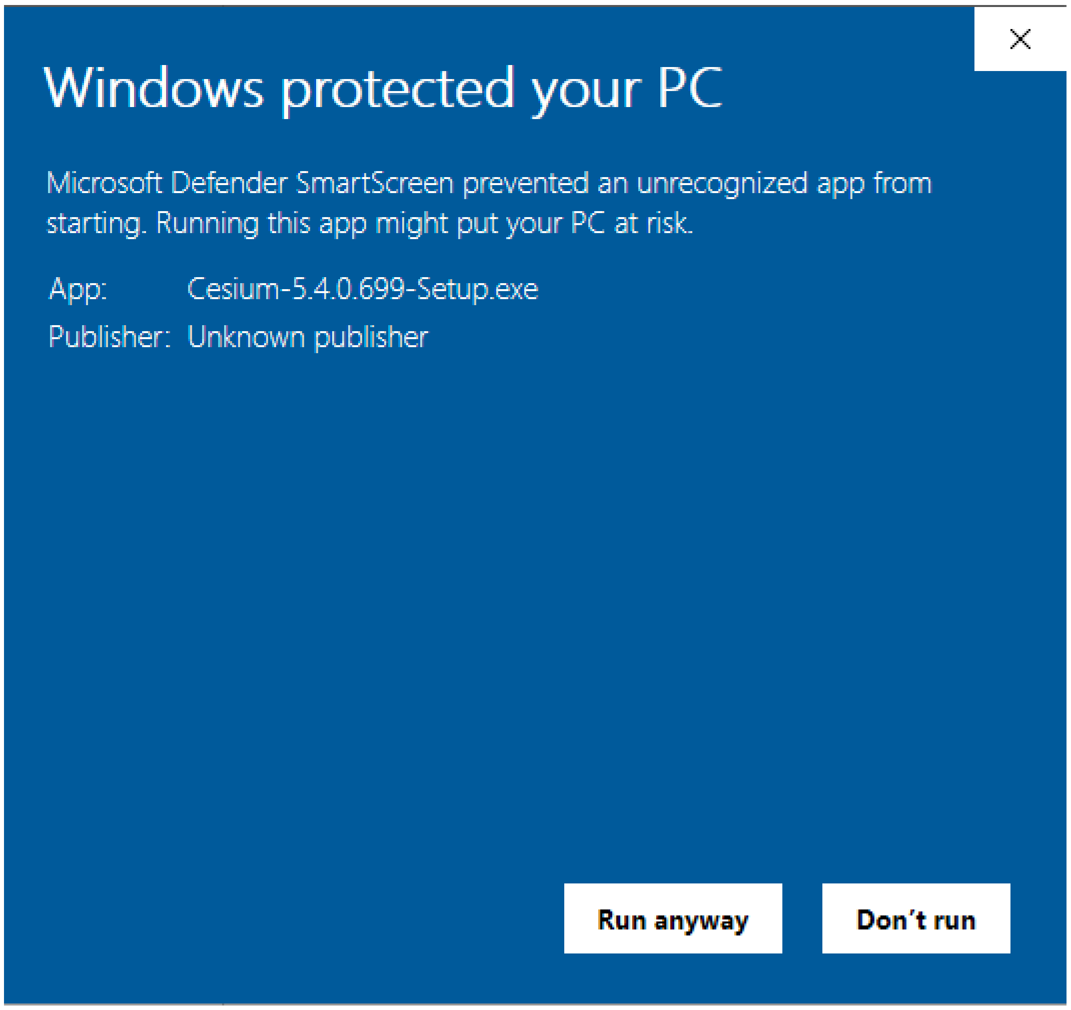
- If you are prompted with the following dialog, click the More info



Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

- Then click Run anyway



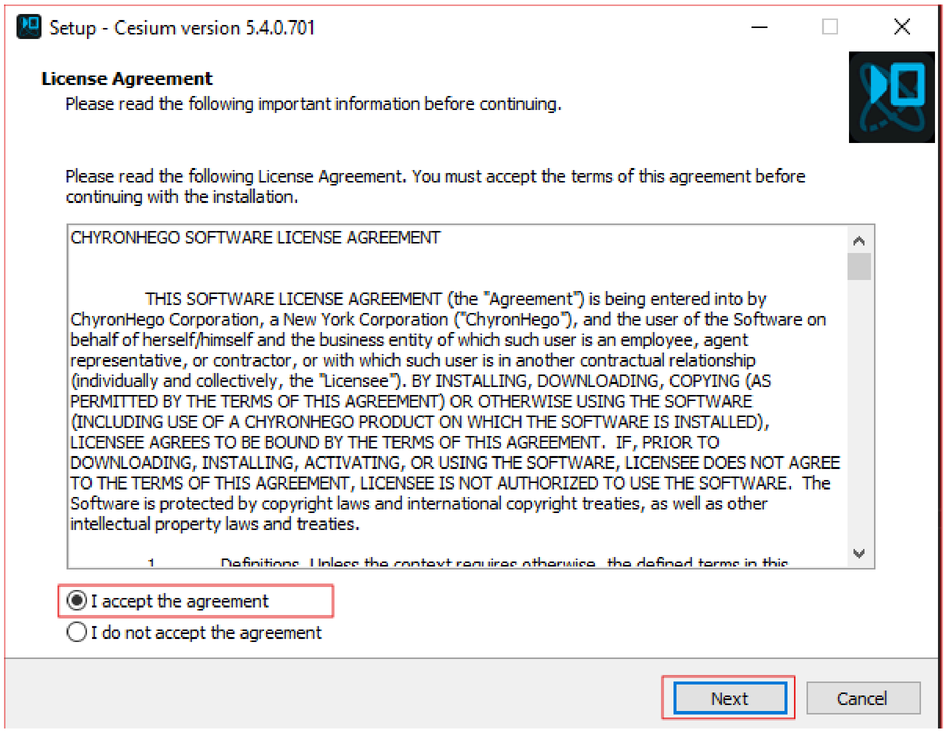
Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

App: Cesium-5.4.0.699-Setup.exe

Publisher: Unknown publisher

- Check “I accept the agreement”, and click Next



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

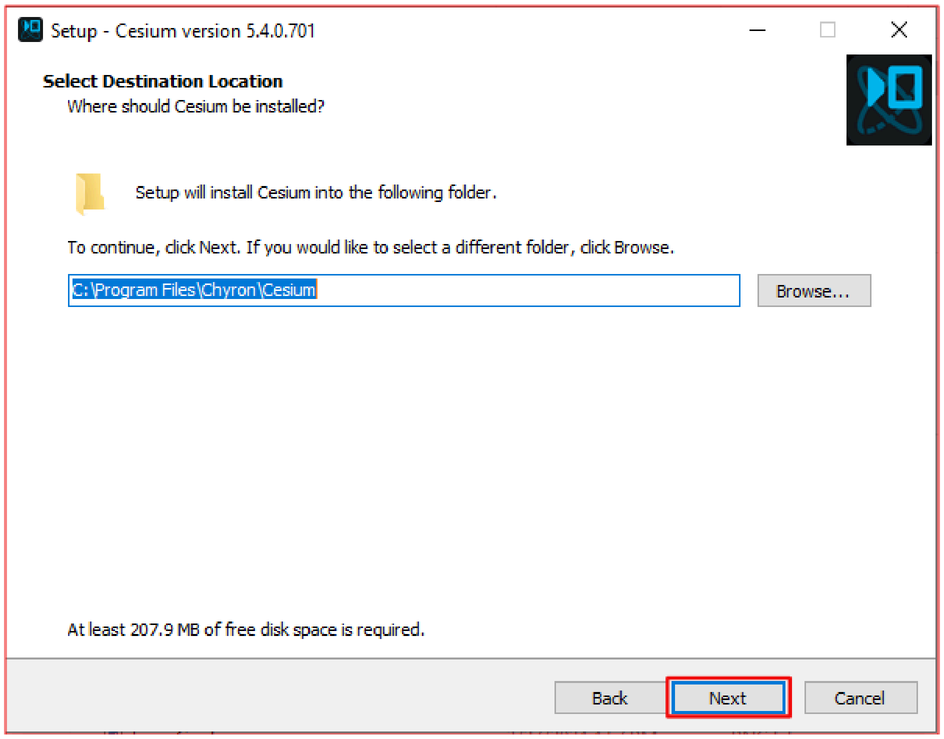
CHYRONHEGO SOFTWARE LICENSE AGREEMENT

THIS SOFTWARE LICENSE AGREEMENT (the "Agreement") is being entered into by ChyronHego Corporation, a New York Corporation ("ChyronHego"), and the user of the Software on behalf of herself/himself and the business entity of which such user is an employee, agent representative, or contractor, or with which such user is in another contractual relationship (individually and collectively, the "Licensee"). BY INSTALLING, DOWNLOADING, COPYING (AS PERMITTED BY THE TERMS OF THIS AGREEMENT) OR OTHERWISE USING THE SOFTWARE (INCLUDING USE OF A CHYRONHEGO PRODUCT ON WHICH THE SOFTWARE IS INSTALLED), LICENSEE AGREES TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF, PRIOR TO DOWNLOADING, INSTALLING, ACTIVATING, OR USING THE SOFTWARE, LICENSEE DOES NOT AGREE TO THE TERMS OF THIS AGREEMENT, LICENSEE IS NOT AUTHORIZED TO USE THE SOFTWARE. The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

[✓ I accept the agreement]

I do not accept the agreement

- Accept the default installation folder or enter a preferred one and click “Next”, we highly recommend to keep the default path, as Cesium installation is rather small.


Setup - Cesium version 5.4.0.701

Select Destination Location

Where should Cesium be installed?

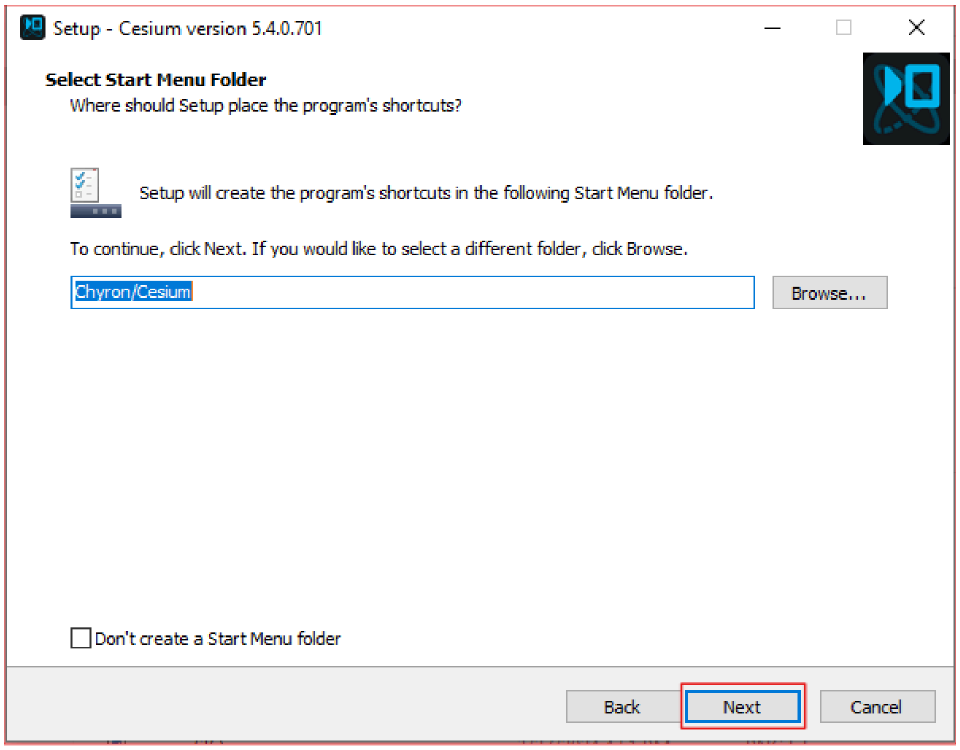
Setup will install Cesium into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Chyron\Cesium

At least 207.9 MB of free disk space is required.

- Repeat the operation for the Start Menu Folder


Setup - Cesium version 5.4.0.701

Select Start Menu Folder

Where should Setup place the program's shortcuts?

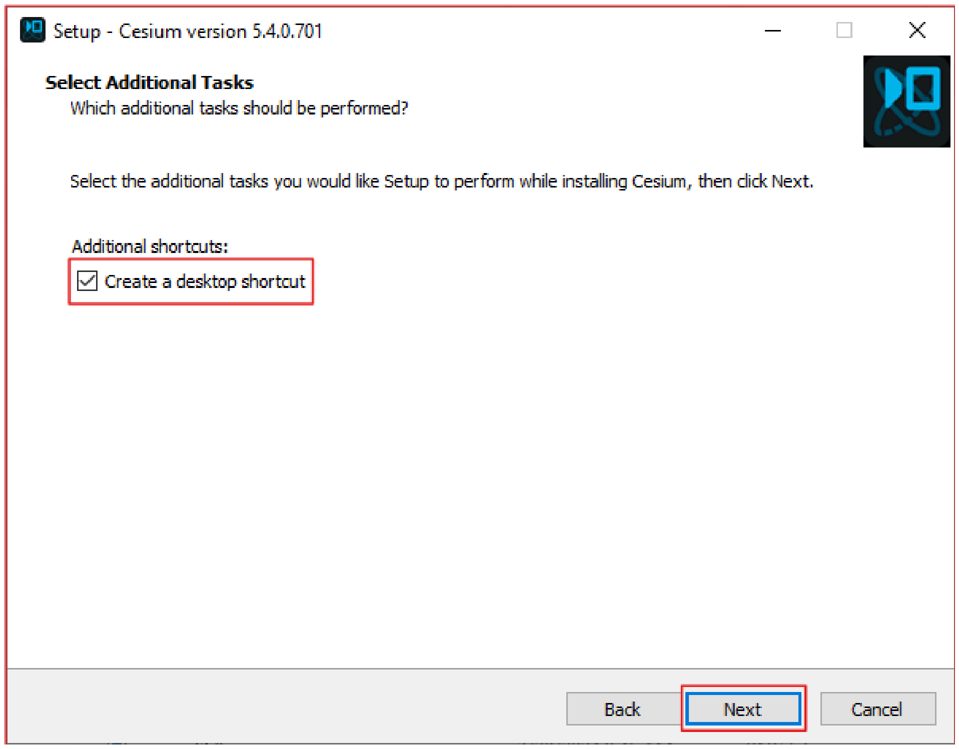
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Chyron/Cesium

☐ Don't create a Start Menu folder

- Optionally create a desktop shortcut and click Next.



Setup - Cesium version 5.4.0.701

Select Additional Tasks

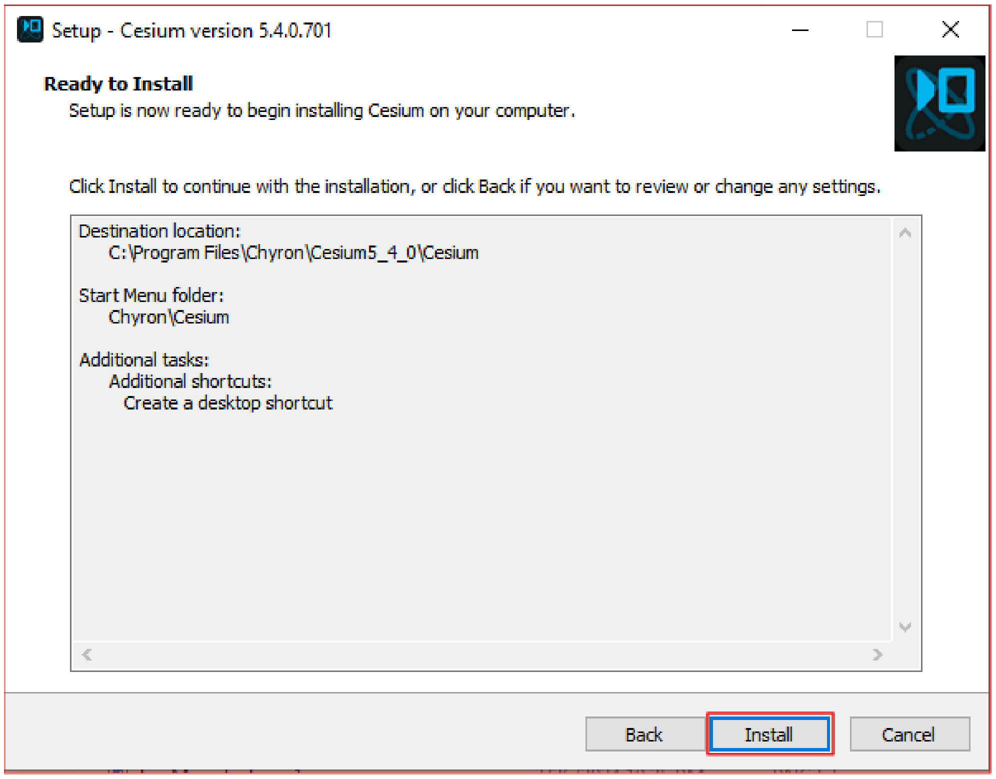
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Cesium, then click Next.

Additional shortcuts:

[✓] Create a desktop shortcut

- And install by clicking “Install”


Setup - Cesium version 5.4.0.701

Ready to Install

Setup is now ready to begin installing Cesium on your computer.

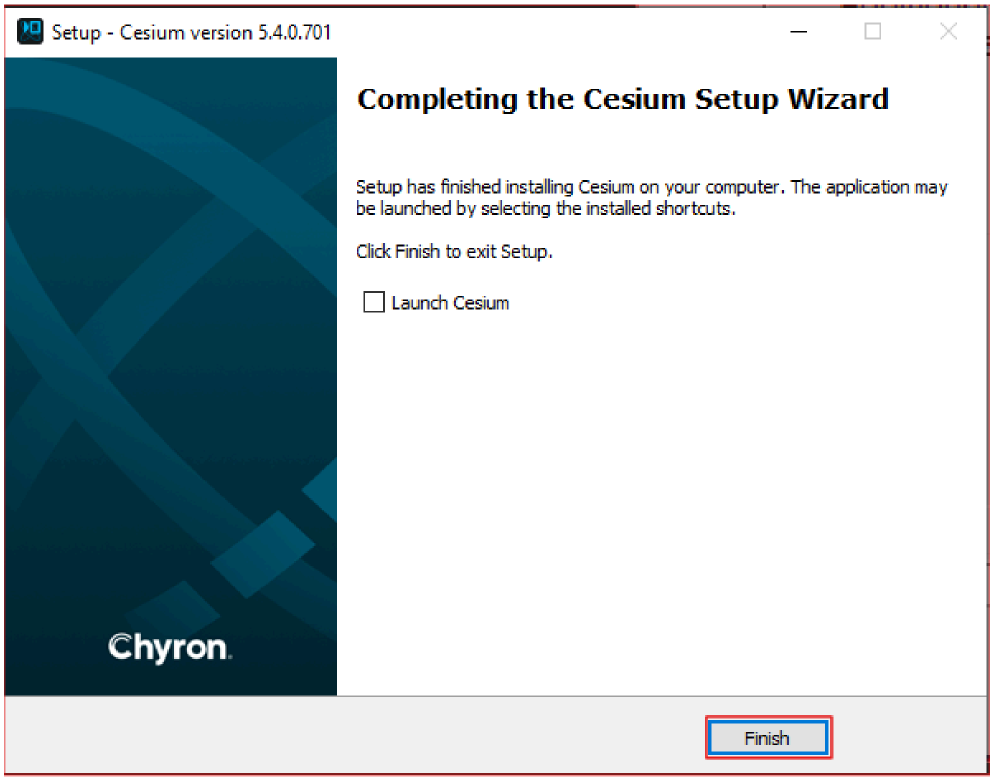
Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files\Chyron\Cesium5_4_0\Cesium

Start Menu folder:
Chyron\Cesium

Additional tasks:
Additional shortcuts:
Create a desktop shortcut

- Software gets installed, click “Finish” to end the installation process.



Setup - Cesium version 5.4.0.701

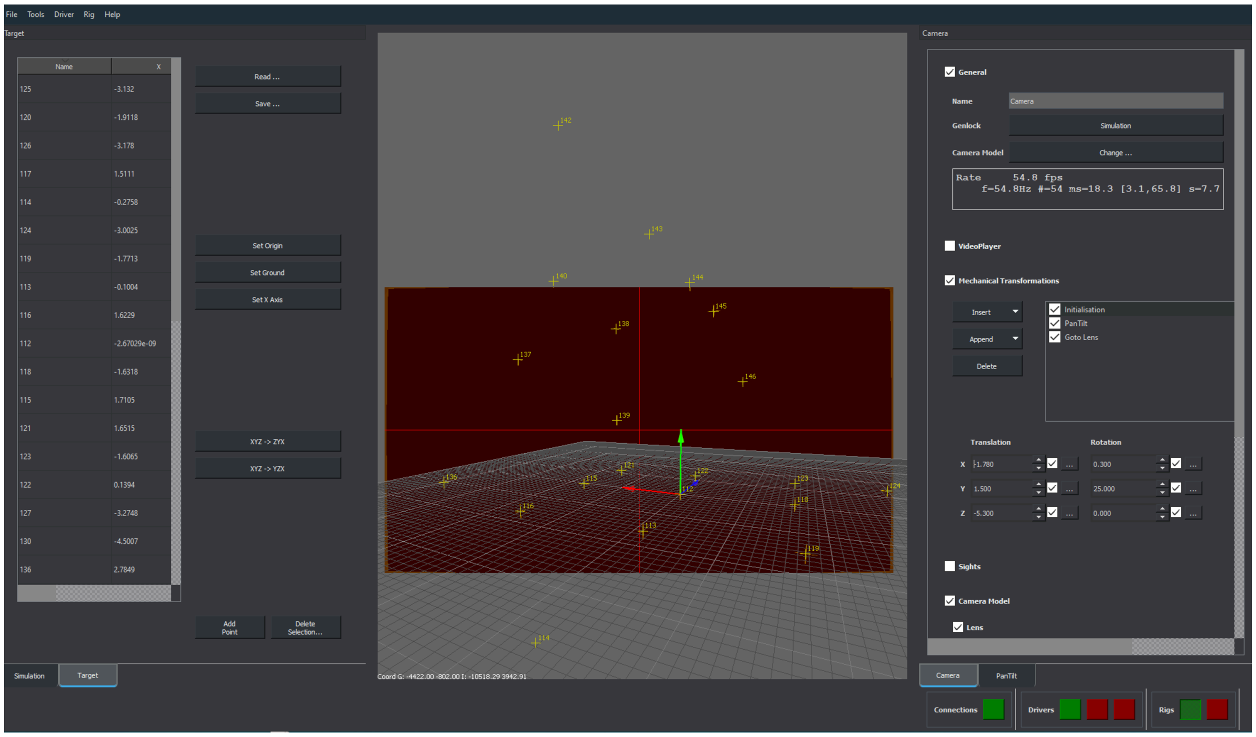
Completing the Cesium Setup Wizard

Setup has finished installing Cesium on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Launch Cesium

GUI Overview



Name	X
125	-3.132
120	-1.9118
126	-3.178
117	1.5111
114	-0.2758
124	-3.0025
119	-1.7713
113	-0.1004
116	1.6229
112	-2.67029e-09
118	-1.6318
115	1.7105
121	1.6515
123	-1.6065
122	0.1394
127	-3.2748
130	-4.5007
136	2.7849

Coord G: -4422.00 -802.00 I: -10518.29 3942.91

✓ General

Name: Camera
Genlock: Simulation
Camera Model: Change ...

```
Rate      54.8 fps
    f=54.8Hz #=54 ms=18.3 [3.1,65.8] s=7.7
```

☐ VideoPlayer

✓ Mechanical Transformations

✓ Initialisation
✓ PanTilt
✓ Goto Lens

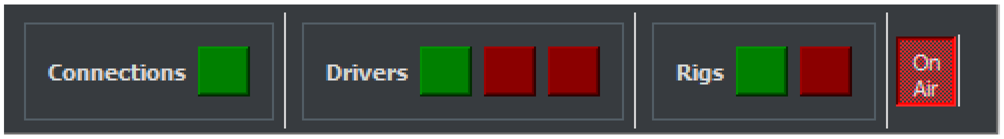
	Translation	Rotation
X	-1.780 ✓	0.300 ✓
Y	1.500 ✓	25.000 ✓
Z	-5.300 ✓	0.000 ✓

☐ Sights

✓ Camera Model

✓ Lens

Here is a sample of Cesium GUI. You can add any floating widgets to the top, left, right and bottom areas. In this picture, there is a Target widget on the left, the Viewer in the center, and a rig on the right. Since they are floating widgets they can also be detached from the main window and act as separated windows. The only widget which is not floating is the viewer widget.

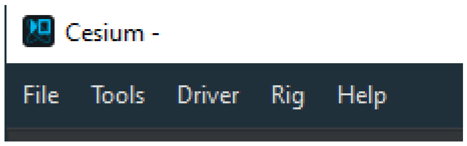


There is an area at bottom - right, that contains Red and Green push buttons for Connections, Drivers and rigs.

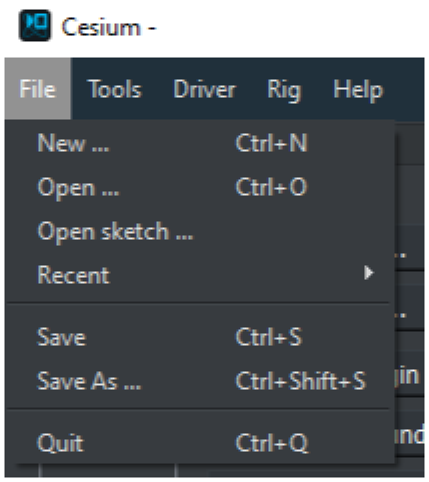
- The most left area is dedicated to showing connections to external renderers. Here there is one connection. Clicking one of these buttons does nothing.
- The middle area is dedicated to drivers. Here it contains 3 drivers, the first one is connected and provides data (green) while the two other ones are disconnected (red). Clicking one of these buttons makes the driver selected and opens or focuses the window for the corresponding driver.

- The right area is dedicated to rigs, here there are 2 rigs, the first one (green) properly delivering data, the second one (red) disconnected. Clicking one of these buttons makes the rig selected, and displayed in the viewer.
- The On Air button, when clicked the viewer is deactivated (to save resources when on air), this button gets periodically auto activated.
- If the buttons are orange the driver/rig is stalled, which is better described further in the **Drivers** section.

Menus



Toolbar menus

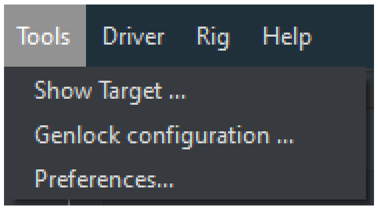


File

- new... - creates a new .cs file
- Open... - loads .cs file
- Open Sketch... loads predefined .cs files,
  Note: that when you open a Sketch file, it is not possible to save it in the same place: use

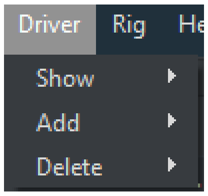
another location to save your file. Most of the time your needs will be covered by the Sketch files, to give you a starting point that you will have to adapt to your proper situation.

- Recent sub-menu for loading last used files
- Save saves current .cs file
- Save As… saves current as new .cs file
- Quit closes the application



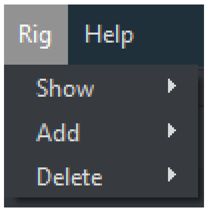
Tools

- Show Targets… Makes the Target dialog visible and available. See The **Targets widget**
- Genlock Configuration … Opens the **Genlock configuration** dialog.
- Preferences… Makes the Preference dialog visible and available. See **Preference Dialog**



Driver

- Show> list of already added drives. Makes the selected driver dialog visible and available.
- Add> Creates a new driver from a list of existing ones. See The **Driver widget** below.
- Delete> Deletes a driver from the list of existing drives.

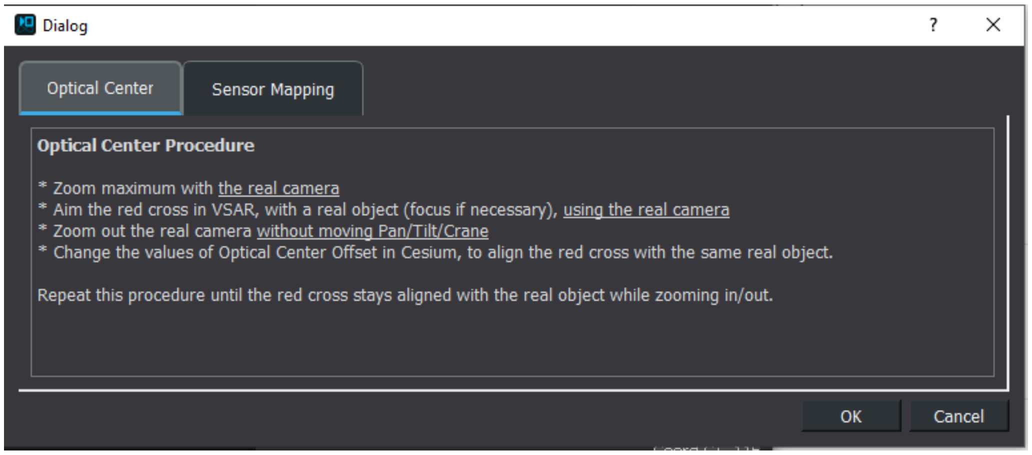


Rig

- Show> list of already added rigs,Makes the selected rig dialog visible and available.
- Add> Creates a new rig from a list of existing ones. See **The Rig widget** below.
    - Pan/Tilt Head creates an almost ready to use Pan and Tilt head rig
    - Pedestal creates a Pan and Tilt head rig + Position X, Y and Z encoders
    - ChromeJib: creates an almost ready to use Chyron robotic jib mechanical rig.
    - Advanced Pan/Tilt: as Pan/Tilt but it contains split Pan and Tilt stages.
    - Advanced ChromeJib: as ChromeJib but it contains Split Boom. Pan and Tilt stages.
    - Parallax: creates a Parallax (aka Screen) rig.
    - NCam : creates a prefilled template to use with the NCam rig.
    - Stype: creates a prefilled template to use with the Stype rig.
    - Mo-Sys: creates a prefilled template to use with the Mo-Sys rig.
- Delete> Deletes a rig from the list of existing rigs.

Help

- Show Help displays some ‘emergency’ help features.
    - Optical Center Procedure



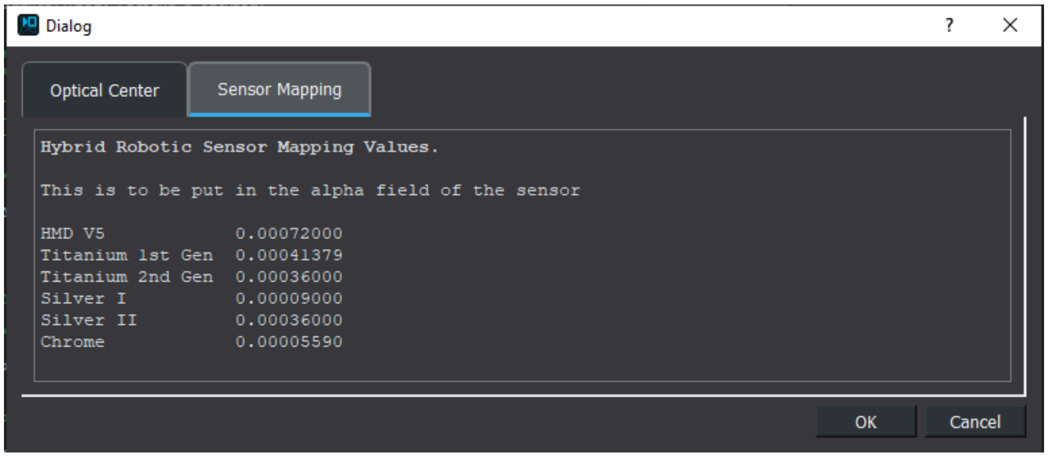
Dialog

Optical Center Procedure

* Zoom maximum with the real camera
* Aim the red cross in VSAR, with a real object (focus if necessary), using the real camera
* Zoom out the real camera without moving Pan/Tilt/Crane
* Change the values of Optical Center Offset in Cesium, to align the red cross with the same real object.

Repeat this procedure until the red cross stays aligned with the real object while zooming in/out.

- Sensor Mapping provides a cheat sheet for Alpha values for Chyron's robotics.



Dialog

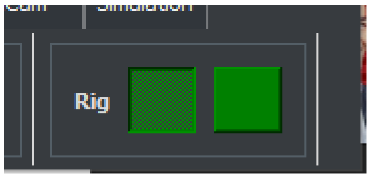
```
Hybrid Robotic Sensor Mapping Values.

This is to be put in the alpha field of the sensor

HMD V5            0.00072000
Titanium 1st Gen  0.00041379
Titanium 2nd Gen  0.00036000
Silver I          0.00009000
Silver II         0.00036000
Chrome            0.00005590
```

- About - cesium version and build

The Viewer

Introduction

The viewer widget displays some 3D information viewed from a given rig. Only a single rig can be monitored at a given time. The rig is selected by clicking a button (green/orange/red rectangle) in the bottom right corner of the main window, or by using the menu “Rig>Show”.



Left mouse allows to translate the viewport, while the wheel will zoom-in and zoom-out.

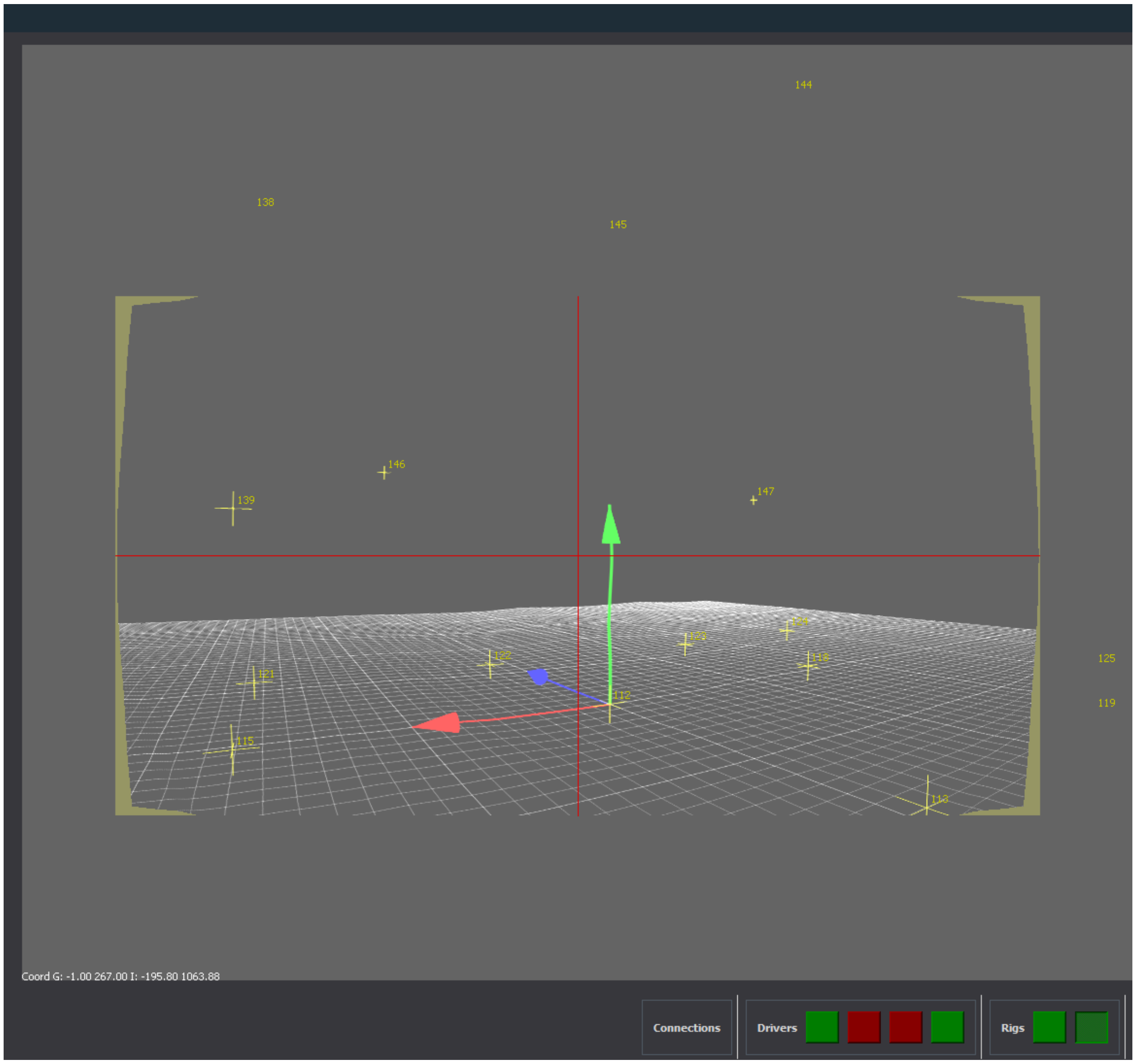
Different elements can be displayed; in the following list each item is preceded by the shortcut key to toggle it:

- [E] Toggle video EE mode
- [F] Fit image to viewport
- [V] Background video image (Matrox or NDI) / Toggle video background
- [G] Ground grid and reference coordinate system.
- [O] Toggle optical center
- [I] Ccd properties / Toggle image Center
- [R] Reset 1:1 scale factor.
- [S] Save Image
- [Ctrl <-] Navigate to the previous image
- [Ctrl ->] Navigate to the next image

The menu with the help above can be shown by right clicking into a viewport(if it is not in On **Air mode** - and from the context menu you select Help)

Various Display modes

The following image show a typical display without the video and ccd properties, this is the default:

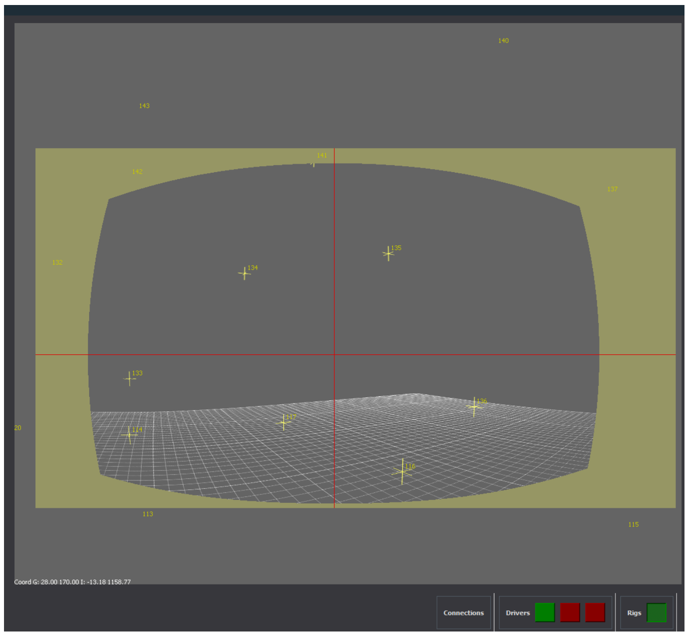


Coord G: -1.00 267.00 I: -195.80 1063.88

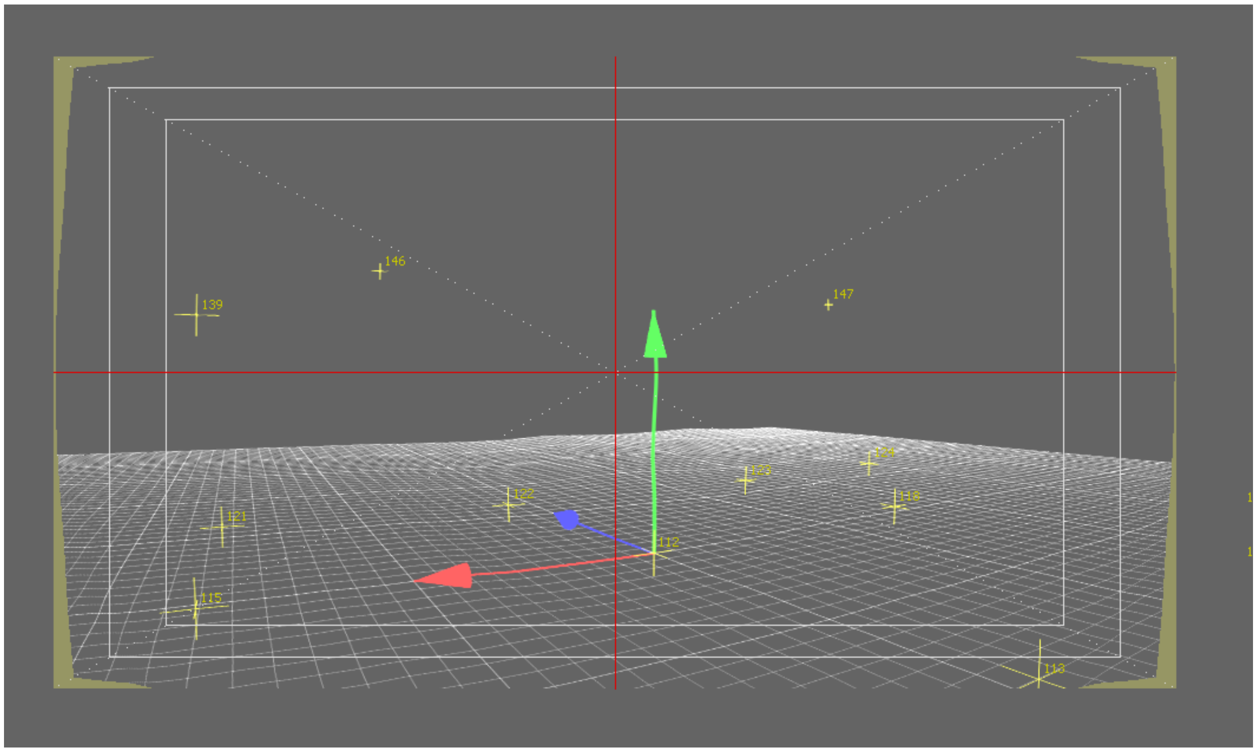
- Targets are 3D crosses and labels displayed in yellow
- Optical center is displayed in red. Note that optical center usually doesn't match the center of the image

- The full frame image (1920x1080) is displayed in dark yellow. This is what the render engine will output.
- The curved area inside the full frame image represents the distorted image.
- The ground grid is displayed as a grid in gray
- The reference coordinates system is displayed in (X)red/(Y)green/(Z)blue

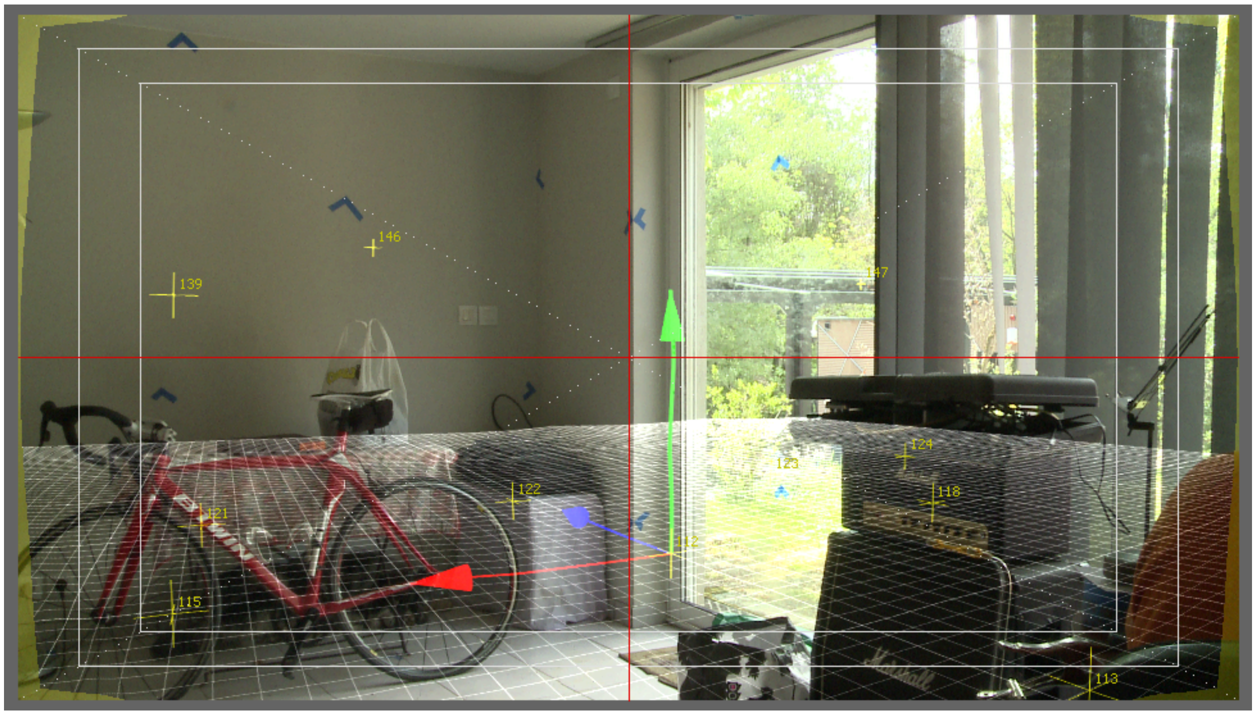
Here is an example with a lens having very large distorsions:



If we add the CCD properties, 2 white frames,respectively 90% and 80% of full frame are displayed. The 2 diagonals of the frame are displayed in white stippled lines.



If we add the video backdrop, we obtain:

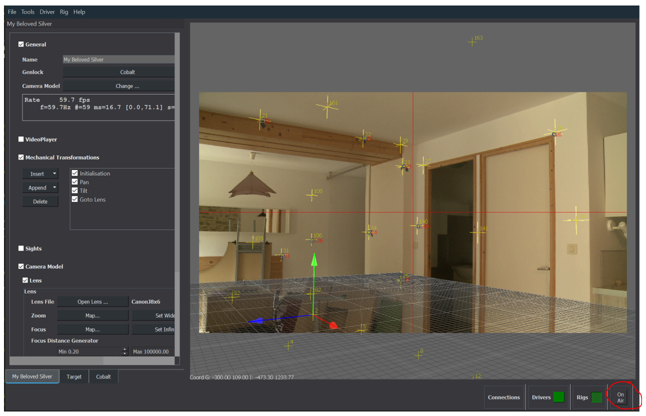


On Air

Displaying in real time graphics in the viewer area is quite GPU consuming, and can penalize VSAR or other 3D rendering systems. Also it is not possible to use HAL/Matrox for Cesium and for VSAR at the same time. If there is no need for the viewer to be continuously updated you need to check the OnAir tool button in the bottom right corner.

⚠️Note that this button triggers itself automatically after a certain amount of time which can be set in Tools->Preferences->Rendering->OnAir Timeout - this setting is in minutes. After this time Cesium will switch to OnAir - the same as if you would click on the button in bottom right.

OnAir mode timeout is also settable from Configuration file (see **Configuration file**)



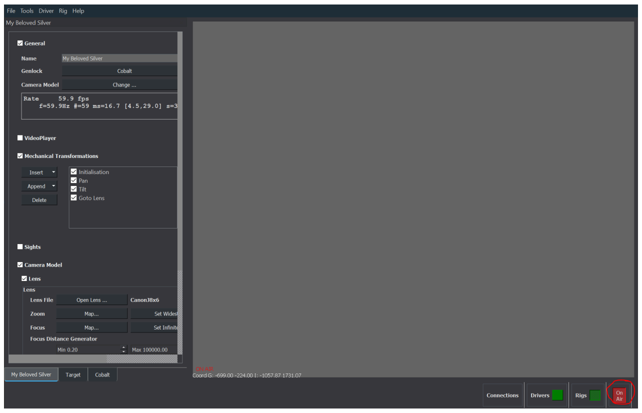
Name: My Beloved Silver
Genlock: Cobalt

```
Rate      59.7 fps
    f=59.7Hz #=59 ms=16.7 [0.0,71.1] s=
```

☑ Initialisation
☑ Pan
☑ Tilt
☑ Goto Lens

Lens File: CanonJ8x6
Min 0.20   Max 100000.00

Coord G: -300.00 109.00 I: -473.30 1233.77



```
Rate      59.9 fps
    f=59.9Hz #=59 ms=16.7 [4.5,29.0] s=3
```

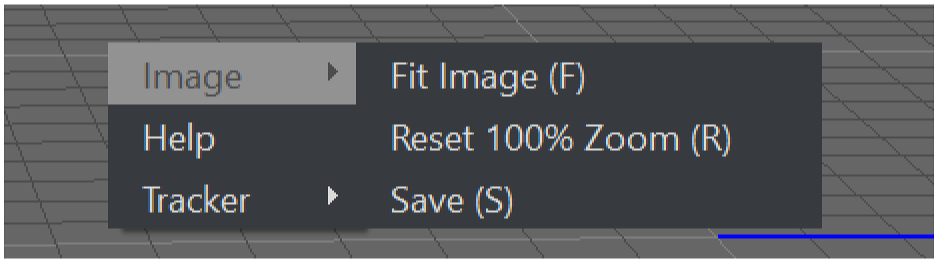
```
Coord G: -699.00 -224.00 I: -1057.87 1731.07
```

When the OnAir button is checked:

- On Air button gets red
- ON AIR is displayed, and rig rendering is disabled in the viewer area.
- Menus are disabled in the viewer area.

Contextual menu

Right clicking in the viewer widget pops a contextual menu:

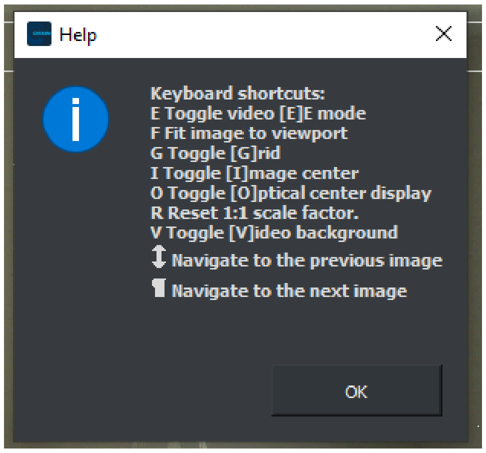


Image

- Fit image: makes the full frame image fits the viewport
- Reset 100% zoom: displays the full frame image with a 1:1 scale factor, that is one pixel of the full frame image is one pixel of the viewport
- Save: Saves the currently rendered image to the hard drive.

Help

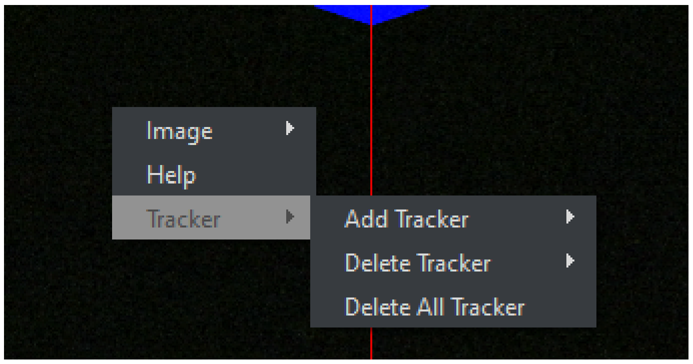
Displays the list of available shortcuts


Help

Keyboard shortcuts:
E Toggle video [E]E mode
F Fit image to viewport
G Toggle [G]rid
I Toggle [I]mage center
O Toggle [O]ptical center display
R Reset 1:1 scale factor.
V Toggle [V]ideo background
↕ Navigate to the previous image
▮ Navigate to the next image

OK

Tracker



This menu allows you to add/delete trackers, trackers are points used for calibration. See **Trackers**. It is only available in non EE mode (see **The Rig "Video Player"**)

- Add Tracker: add a new tracker from the list of available trackers.
- Del Tracker: deletes a tracker from the current trackers.
- Delete All Tracker deletes all trackers

Using trackers

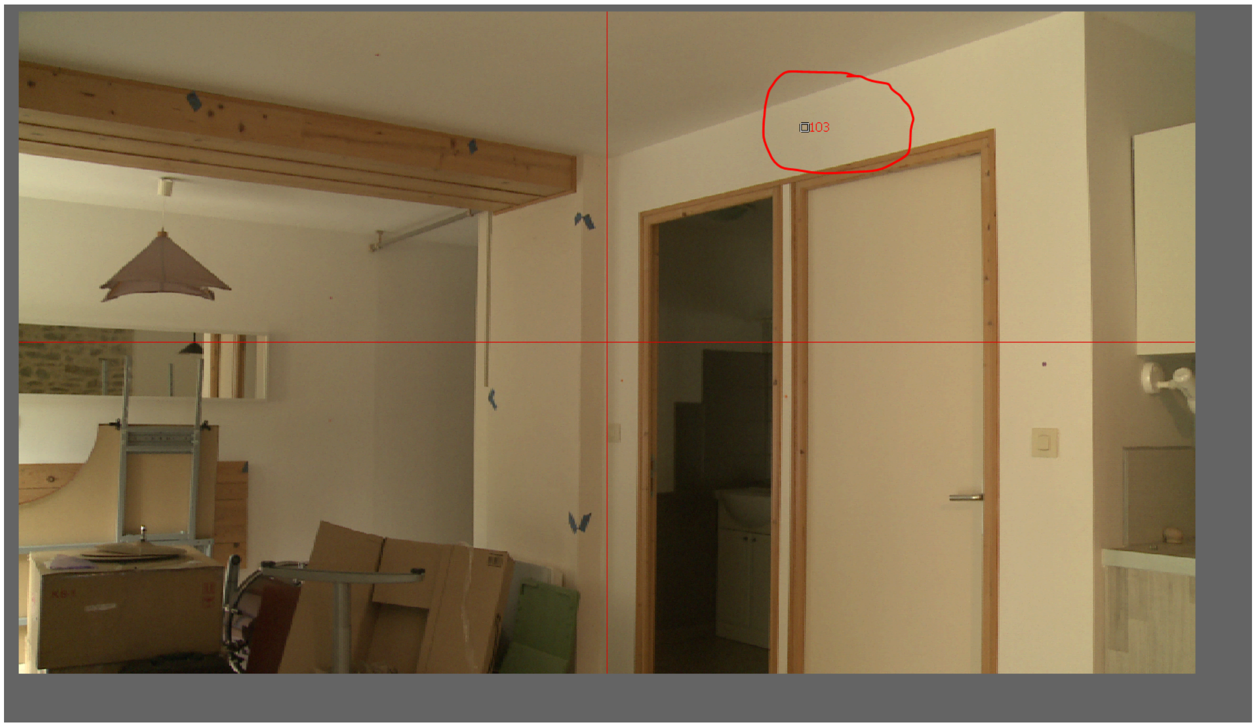
Trackers are used for calibration (see **Calibration**). Trackers are 2 points that you specify on recorded frames, EE mode should be unchecked for this feature to work (see **The Rig "Video Player"**).

In the remainder of this chapter we will assume that a captured frame is selected(The frame is highlighted in blue as you can see in the picture in **this chapter**).

Add a tracker

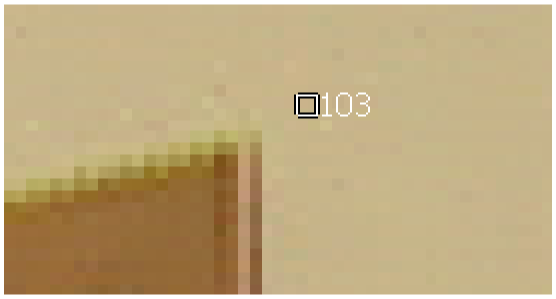
To add a tracker, use the contextual menu and select RightMouse>Tracker>Add Tracker. A list is proposed and contains all the points that have been entered as Targets, and that are not trackers already. So you need to have a valid target set to work with trackers. (see **Target**).

When a target is added, it's displayed as a white square and a red label.

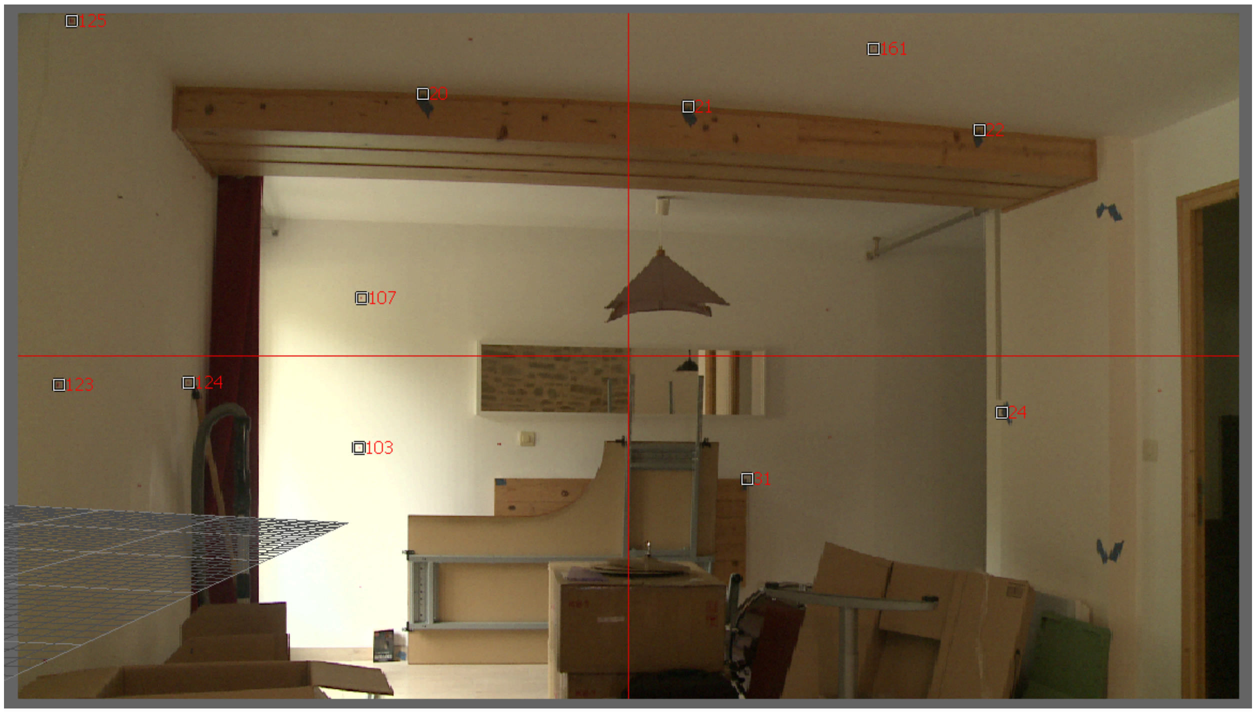


Move a tracker

You need to precisely move the tracker on its corresponding real position. For this select the tracker with the left mouse button and drag it. You will need to zoom to precisely move it, use the mouse wheel for this to zoom-in and out of the image. While the tracker is selected its color switches to white.



Here is an example of a frame filled with 11 trackers:



Delete a tracker

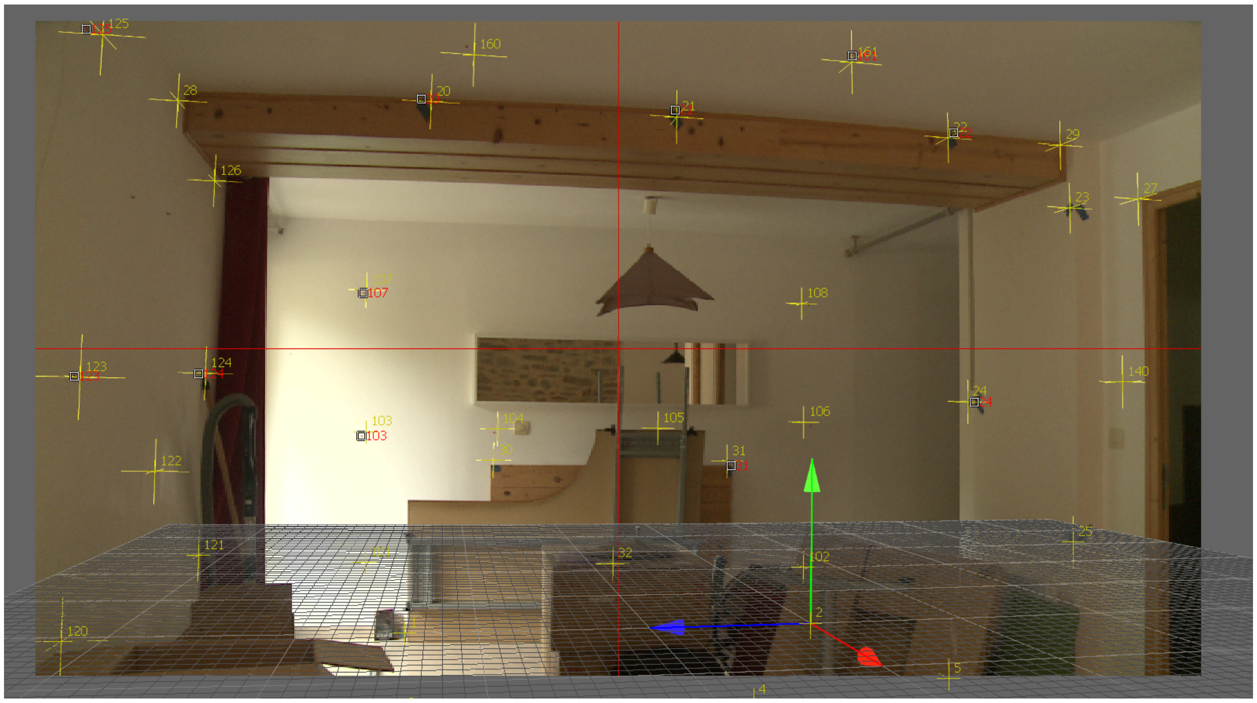
To delete a tracker, use the contextual menu and select RightMouse>Tracker>Delete Tracker. A list is proposed and contains all the current trackers.When a target is added, it's displayed as a white square and a red label.

Delete All Tracker

To delete all tracker, use the contextual menu and select RightMouse>Tracker>Delete all Tracker.

Trackers and targets

When the calibration process is done, targets (displayed in yellow) should be close to their tracker counterparts. For example, this is a relatively well calibrated camera:



Advanced configuration

This chapter can be omitted for non advanced users.

You can change some advanced parameters of the viewer by entering the Preferences … dialog, and modifying the settings in “Rendering”.

Focal Multiplier: When distorsions are applied, it could happen (quite often) that when folding the undistorted image, some image is ‘missing’, this is why there may be a yellow area between the distorted image and the full frame image. To remedy part of this effect, you can increase the ‘Focal multiplier’ parameter; the drawback is that the rendered image might become more blurry as you are basically stratching the image to cover for the distorted areas.

CPU Enabled: If checked parts of the rendering will be done by the CPU. In this mode, targets, grids, … are displayed using approximations for the distorsions. Though fully exploitable the displayed result is less nice that it’s GPU counterpart. In this mode, the 3D elements (for example the rectangles used to test Stype) cannot be displayed.

GPU Enabled: If checked, rendering is done using the GPU. This mode leads to faster and nicer renderings. 3D elements are properly rendered (see upper). It may happen that on some computers the 3D acceleration is not available, thus you would have to switch to CPU Enabled mode.

In GPU mode, since distorsions are rendered by applying a shader to a texture, it happens that if you zoom too much in the viewport, pixelation happens; this does not happen in CPU mode.

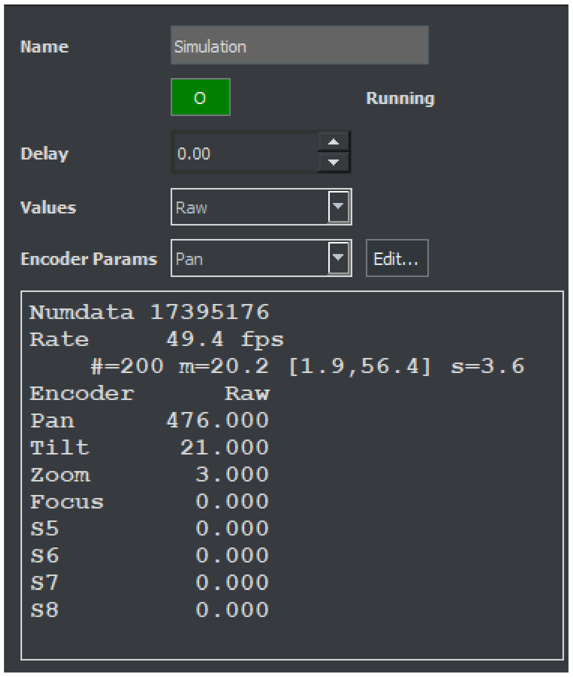
Finally, you can perfectly have both CPU and GPU enabled at the same time, which in most cases is recommended, as Cesium decides itself what to use on what if both are enabled.

Drivers

A driver is responsible for gathering data from 3rd party vendors, and delivering these data to the connected rig(s) in a uniform manner.

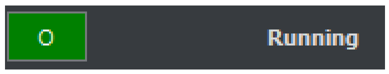
The currently recognized list of drivers is:

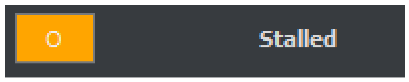
Common Driver Interface

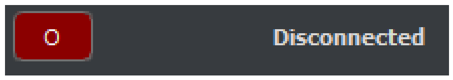


```
Numdata 17395176
Rate      49.4 fps
    #=200 m=20.2 [1.9,56.4] s=3.6
Encoder       Raw
Pan       476.000
Tilt       21.000
Zoom        3.000
Focus       0.000
S5          0.000
S6          0.000
S7          0.000
S8          0.000
```

All the drivers have a common part, located in the upper part of the driver Gui.

- Name allows to change the name of the driver (here “Simulation”)
- Status can have 3 states:
  - 

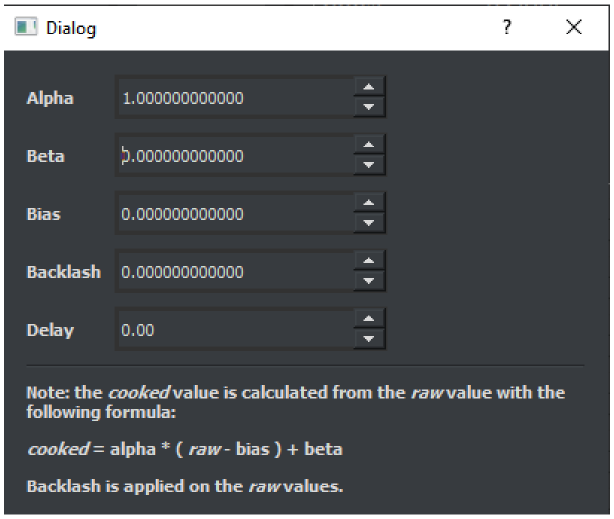
    The driver is connected and is receiving data.
  - 

    The driver is connected but does not receive data.
  - 

    The driver is not connected.
- You can connect a driver by pressing on the green/orange/red “O” pushbutton. By default drivers are created in an unconnected state.
- Delay sets the overall delay of the driver. This delay is expressed in driver frames (driver time-stamps). For example for a 50fps genlocked driver the delay set to 50.0 will result in ~1 second delay. The daily value allows for fractions of a driver frame. When setting this value,

all the encoders are delayed by the same amount. There is a way to set a specific delay per encoder (see Encoder Params).

- Values select the type of values you want to display in the information panel. You can display:
    - Raw: Those are the raw (which means - not normalized) values of the encoder.
        - Eg.: a sensor on silver head provides data that has to be normalized by a coefficient of 0.00036000 to get normal - degree data from the head about Pan and Tilt. (For Chyron robotics the above mentioned constant can be found in the Help menu mentioned above, for other vendors these values should be provided by the manufacturer).
    - Cooked: Those are the normalized values of the encoders. Those values are then passed to the rigs.
    - Coeff: displays the parameters used to convert Raw to Cooked values.
        - As above mentioned for Silver 2 head this would be Alpha = 0.00036000
        - These parameters are inserted in the dialogue described below
- Encoder Params allow to change the parameters used to convert Raw to Cooked values. First start by choosing the relevant encoder in the drop down list (here “Pan”) and click “Edit...”. The meaning of the different values is explained in the dialog window. Delay allows to set a specific delay for this encoder, different from the other ones from the same driver (see Delay above)



Alpha	1.000000000000
Beta	0.000000000000
Bias	0.000000000000
Backlash	0.000000000000
Delay	0.00

Note: the *cooked* value is calculated from the *raw* value with the following formula:

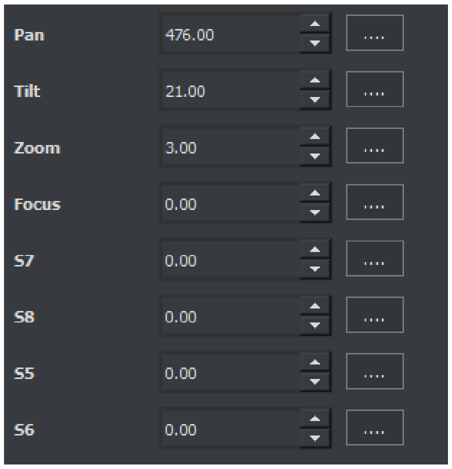
cooked = alpha * (*raw* - bias) + beta

Backlash is applied on the *raw* values.

- Display area: Displays the driver parameters. The content of this differs from driver to driver, but basically has the following information:
    - NumData: this value is incremented by the Vendor when a new message is sent. If the vendor cannot provide such information, this value is incremented when Cesium receives a new message. When the driver is properly running, this value should increase steadily.
    - Rate: display the rate (fps) and other statistical information about the reception of messages. Example: Rate 50fps #=200 m=20 [11.8,32.4] s=2.4 means: reception rate is 50fps, number of message received to compute stats = 200, period is 20ms, min duration between 2 messages is 11.8ms, max duration between 2 messages is 32.4ms, and standard deviation is 2.4ms. In practice, just check that the displayed frame rate is what you expect: usually 50 or 60.
    - Encoder Raw|Cooked|Coeff then displays the value of each encoder, depending on the chosen format (see Values above).

Simulation

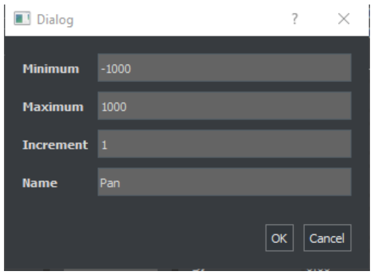
Allows to generate 8 emulated values so that one can work without a real camera tracking device. Datums are delivered at a rate determined by the global **Genlock Configuration**.



Pan	476.00	....
Tilt	21.00	....
Zoom	3.00	....
Focus	0.00	....
S7	0.00	....
S8	0.00	....
S5	0.00	....
S6	0.00	....

The simulator driver allows you to manually change the encoder values. You can set the range, sensitivity and name of each encoder by clicking on the "..." configuration button.

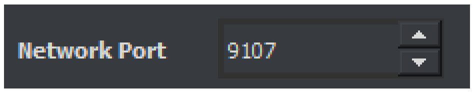
Simulation driver is used to test out whether Cesium is correctly set up and connected into VSAR on a Cesium Camera. You can play with it freely, without the need of having a real camera and tracking system/solution.



Minimum -1000
Maximum 1000
Increment 1
Name Pan

Chyron Robotics - Cobalt

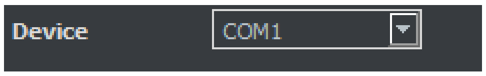
Allows to handle a Chyron Cobalt device, which in turn can be connected to several Chyron robotic cameras. You can just set the port onto which the Cobalt emits its data, default is 9107. If you want to receive data from the Cobalt, you need to enter the machine's IP in Cobalt configuration.



Network Port 9107

Free D Serial

Handles the D1 serial protocol (38400/odd/8bit/2stop). Select the serial port from the drop down list. D1 protocol sends 29 bytes long messages, and uses 0xD1 as synchronization byte.
Handles the FreeD serial protocol.



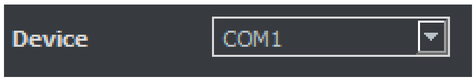
Device COM1

Free D IP

Handles the D1 IP protocol. Select the IP port your device is sending to (Default is 6302). D1 protocol sends 29 bytes long messages, and uses 0xD1 as synchronization byte.

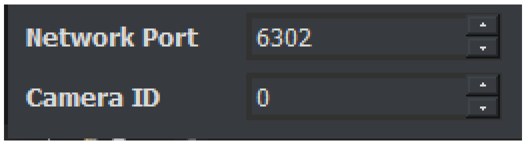
MoSys Serial

This driver is used for the serial line MoSys StarTracker (but also for other MoSys products). Select the COM device the MoSys is connected to.


Device COM1

MoSys IP

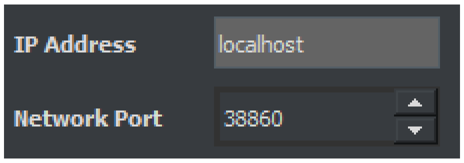
This driver is used for IP MoSys StarTracker (but also for other MoSys products). Select the port onto which MoSys is emitting the data, default is 6302. Note that you need to register Cesium's IP in StarTracker to have it for sending data to Cesium.


Network Port 6302
Camera ID 0

Camera ID is set in MoSys (in MoSys -> advanced)

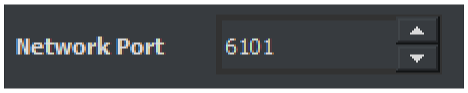
NCam

Select the host IP and port, onto which NCam (camera or emulator) is running. Defaults are "localhost" and 38860. Data Stream mode to be used is "SDK Full Features".


IP Address localhost
Network Port 38860

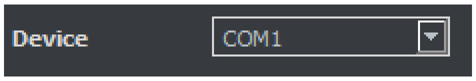
Ross

Select the Ross IP port onto which data are transferred. Default is 6101.


Network Port 6101

Shotoku A0

Implements the shotoku protocol using A0 only data blocks. Enter the serial port onto which Shotoku is connected to. Line speed is set to 38400 baud.


Device COM1

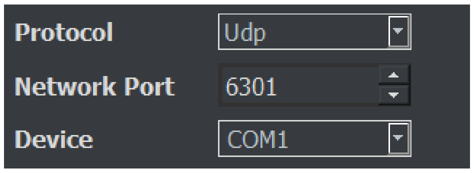
Stype

Select the Protocol between “Udp” and “Serial”. Nowadays “Udp” is more commonly used than “Serial”.

If “Udp” is used, then select the Network Port onto which Stype emits its data, default is 6301.

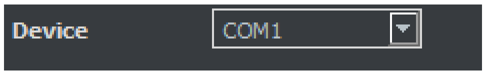
If “Serial” is used, then select the serial Device to which Stype is connected to.

Note: only the “HF” protocol is implemented for Stype communication.


Protocol Udp
Network Port 6301
Device COM1

Vinten

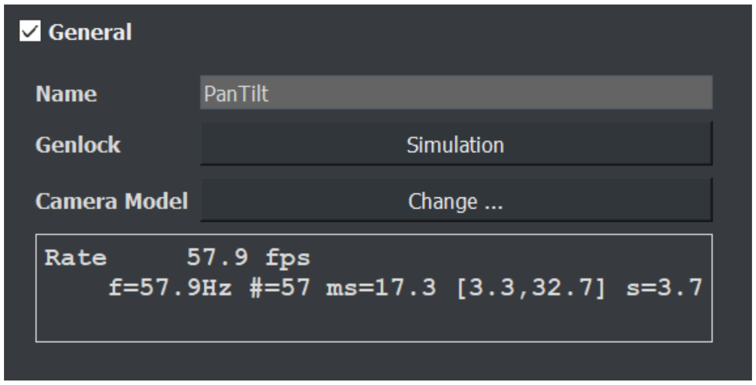
Handles the Vinten serial protocol. Select the serial port from the drop down list. Vinten protocol sends 15 bytes long messages, and uses 0x04 as synchronization byte.


Device COM1

Rigs

Rigs come in mainly two flavors, Mechanical and Parallax, they have their own specific user interface, but have in common the “General Tab”. The “Ncam” and “Stype” pre-defined rigs are variations around the generic Mechanical rig.

The Rig “General Tab”



Name: PanTilt
Genlock: Simulation

```
Rate      57.9 fps
    f=57.9Hz #=57 ms=17.3 [3.3,32.7] s=3.7
```

“Name”: allows you to change the name of the rig.

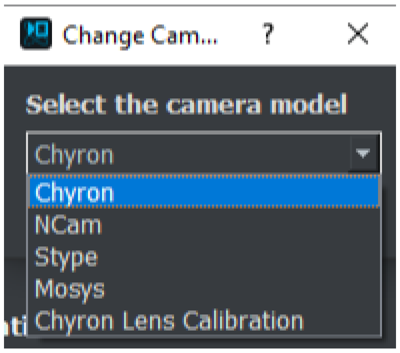
“Genlock”: selects the genlock source that will trigger the sending of the camera data to external 3D viewers (VSAR/**PRIME**). The genlock source could be either:

A driver for mechanical rigs.

A Rig for parallax rigs.

Note that all genlock sources are not available to any rig, it depends on the rig type and on the declared genlock sources.

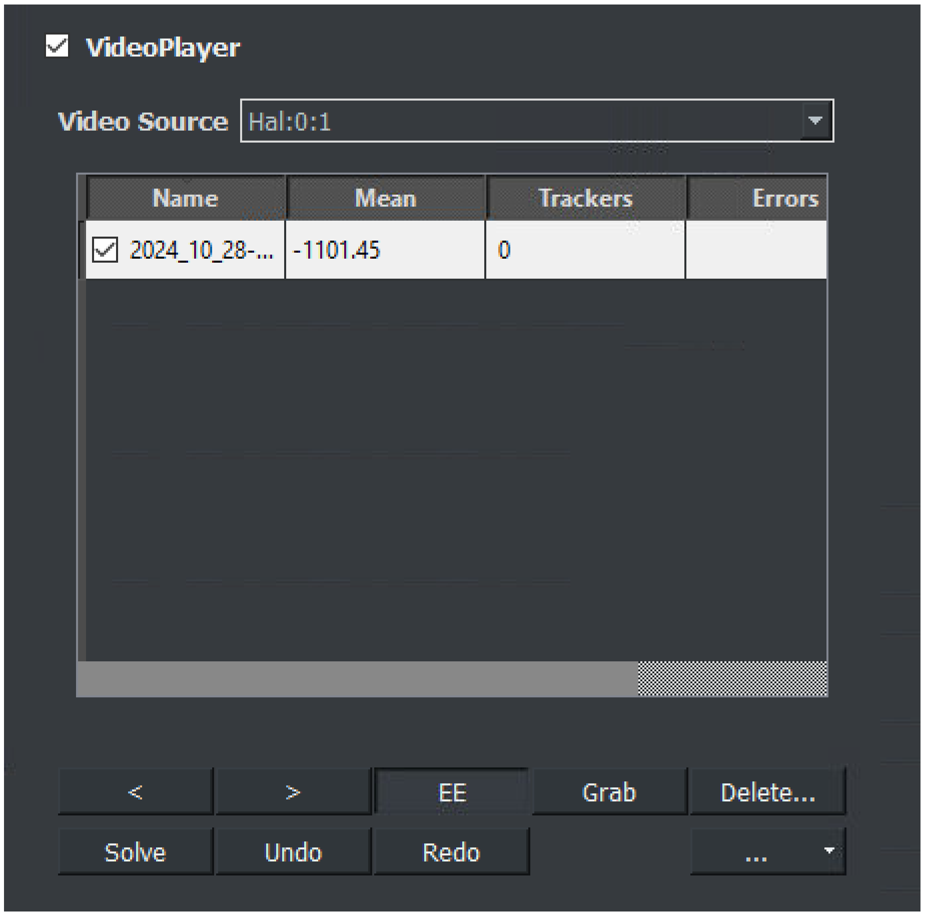
“Camera Model”:



allows to change the camera/ccd/lens model used in mechanical rigs. There are 4 types of camera model available, cf infra for more information on each model:

- Chyron
- NCam
- Stype
- Mo-Sys
- Chyron Lens Calibration

The Rig "Video Player"

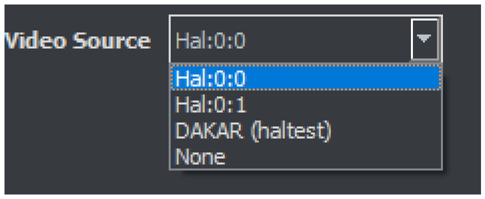


Name	Mean	Trackers	Errors
2024_10_28-...	-1101.45	0	

Video Source

You can attach a video source to each rig, this video source will be displayed in the viewer widget as a background layer.

You can change the video source by clicking the drop down menu.



In this example, there are 4 video sources available:

- Hal:0:0 is the first video source declared in the **Hal.xml file**.
- Hal:0:1 is the second video source declared in the **Hal.xml file**.
- DAKAR(haltest) is the first detected NDI video source. When a new NDI source appears or disappears the drop down box is automatically updated, after a slight latency due to the network discovery process. You can set the default pixel and video format for Ndi in the **configuration file**.
- None disable the background video source.

Image browser

The image browser allows you to navigate through recorded frames. This is particularly useful for calibration. Each line contains the following information:

- Name: If the checkbox is checked, it means that the frame is enabled for calibration computing.
- Name: name of the image
- Mean: Mean error over all the picked points in pixel
- Trackers: Number of active trackers in the image
- Errors: A decreasing list of errors. In the example the worst point for the third image is 24, with an error of ~24.8 pixels. This list can be used to identify 'bad' trackers.

Transport buttons

- < navigates to the previous image [Left arrow in the viewer]
- > navigates to the next image [Right arrow in the viewer]
- "Solve"- try to find the calibration solution parameters from the picked trackers.

- “EE” when checked the viewer displays the live video (not the recorded frames), when unchecked the selected recorded frame is displayed. Note: EE comes from analog VTR, it means Electronic to Electronic, or bypass.
- “Grab” captures the current video input, together with the relevant encoders and adds it to the image browser.
- “Delete..” deletes the currently selected image from the image browser.
- “...”
    - “Delete Trackers” for current Frame - removes trackers from selected image
    - “Delete Trackers for All Frames” - removes trackers from all image
    - “Delete All Frames” - removes all image

Notes

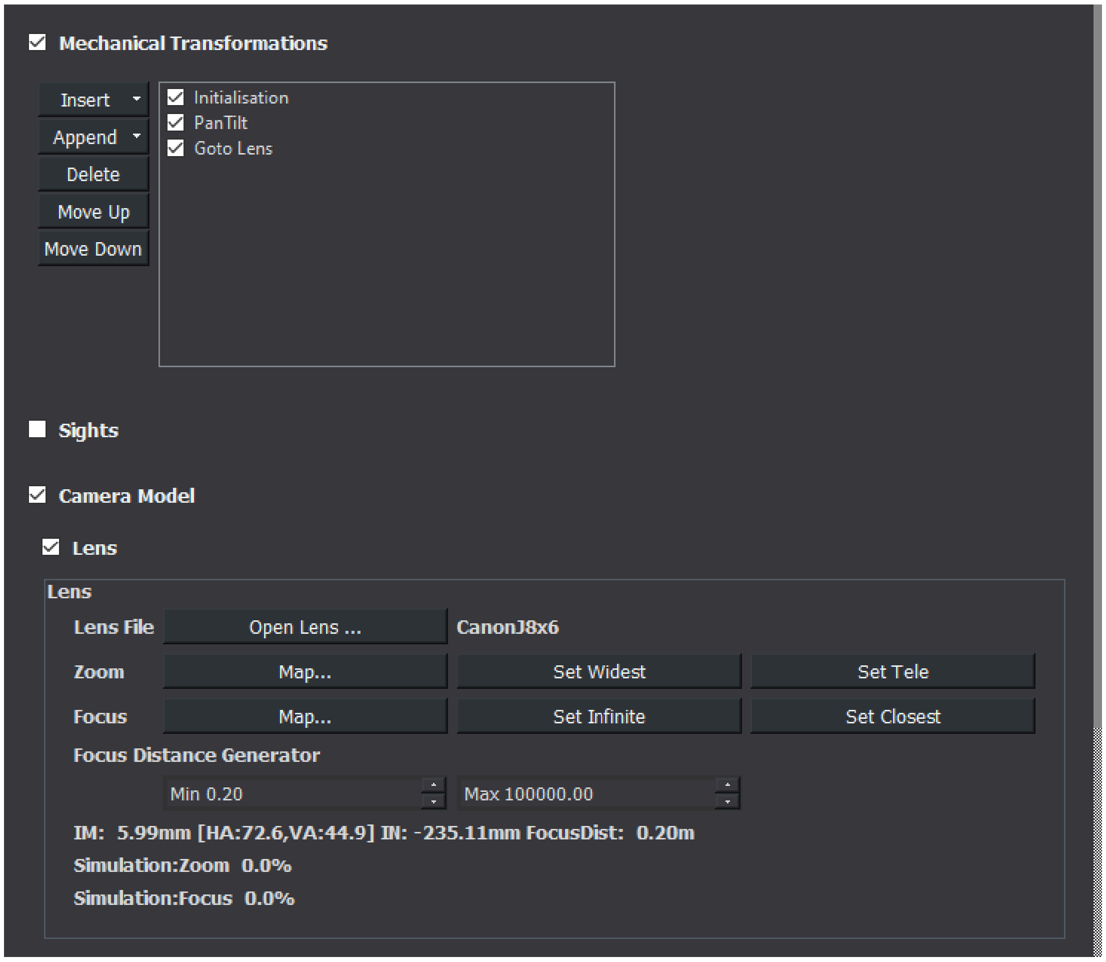
- Images could be saved embedded in the .cs file, or externally on disk. This behavior is controlled by the “Tools→Preferences→System→Embed Images in .cs file” checkbox. We recommend leaving it checked for simplicity.
- If EE is not selected, then you can add/remove trackers to the currently selected image (see Viewer section). If EE mode is selected, then trackers can’t be added/removed.

Mechanical Rigs

Mechanical Rigs (as opposed to Screen/Parallax rigs), are used to modellize traditional camera systems.

An mechanical rig contains:

- A “Mechanical transformations” subpanel which allows the user to describe the list of mechanical transformations used to locate the camera in world coordinates.
    - Note that these transformations are dependent on their order, as internally they are represented as matrix multiplications, which are not commutative operations.
- A “Lens and Reference image” subpanel, which depends on the kind of lens/camera model is used on the camera system. The Lens/Camera model is selected by clicking on the “Camera Model” pushbutton. Depending on the “Lens/Camera” model selected the UI may differ from what’s displayed below.


☑ Mechanical Transformations

☑ Initialisation
☑ PanTilt
☑ Goto Lens

☐ Sights

☑ Camera Model

☑ Lens

Lens

Lens File: CanonJ8x6

Focus Distance Generator

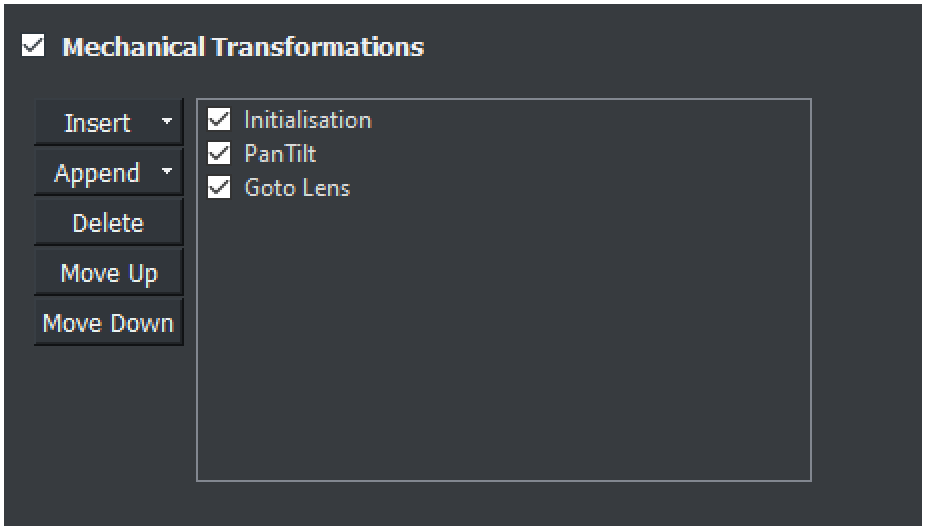
Min 0.20 | Max 100000.00

```
IM:  5.99mm [HA:72.6,VA:44.9] IN: -235.11mm FocusDist:  0.20m
Simulation:Zoom  0.0%
Simulation:Focus  0.0%
```

This displays an Online rig with a "Chyron Hego" lens model (cf infra for other lens models)

Mechanical Transformations

"Mechanical transformations" models the list of transformations necessary to compute the final camera position and orientation. It does not include lens/receptor properties.



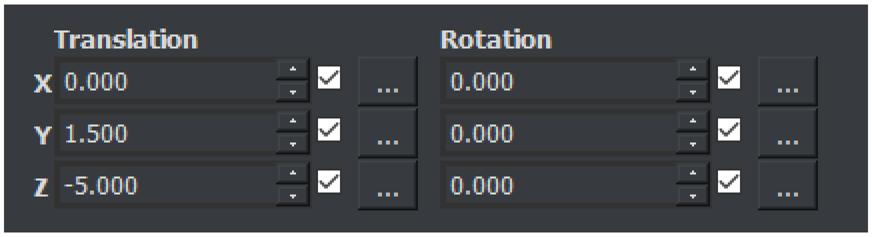
The transformation list is read from top to bottom. In the above example, starting from the world coordinate system, we first apply the “Initialisation” transform, then the “Pan Tilt” transform and finally the “Goto Lens” transform.

Each transformation applies first a translation and then a rotation.

You can add, before by clicking Insert or after by clicking Append, a new transformation to the currently selected transformation. Delete removes the currently selected transformation.

Here is the list of available transformations:

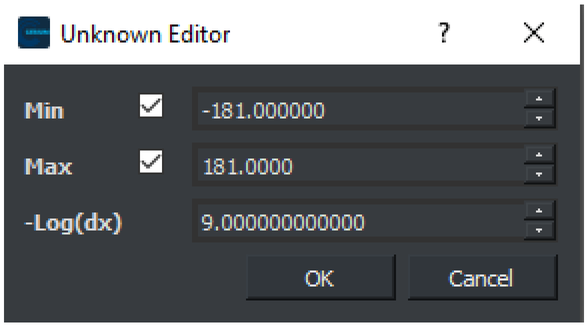
- Fixed Transform: this is a static transformation, in the sense that it is not encoder driven. Field values are quite obvious, translations are in meters, rotations in degrees.



	Translation	Rotation
X	0.000	0.000
Y	1.500	0.000
Z	-5.000	0.000

Each of the field can be:

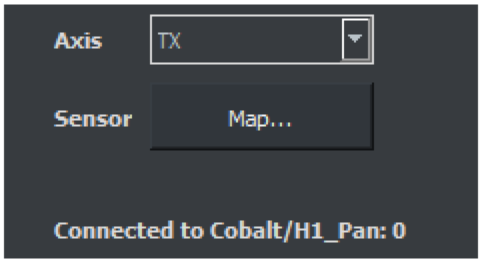
  - checked: the entered value is locked and cannot be changed by future computations
  - unchecked: the entered values will be sought for during the computations
  - “...” click tool button opens the Unknown Editor


Unknown Editor

Min	✓	-181.000000
Max	✓	181.0000
-Log(dx)		9.000000000000

which allows to edit range of the value and the -Log(dx) used for calculation

- Sensored transform: this is a dynamic transformation that depends on a single encoder value. It can be used to modelize Pan, or Tilt, or Travelling, …

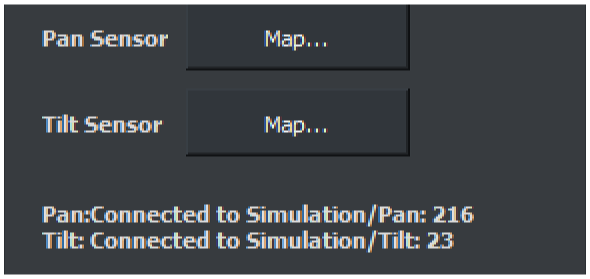

Axis: TX

Sensor: Map…

Connected to Cobalt/H1_Pan: 0

Select the axis associated with the encoder by changing the Axis combo box. For example, usually Pan sensored transforms are on the RY axis. Click on Map… to tell the transform which encoder to use. Mapping values (alpha/beta/…) have to be set in the **encoder configuration panel**. When properly connected the actual cooked value is displayed in the bottom label.

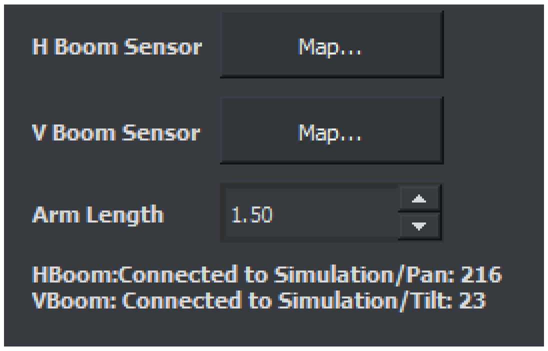
- Pan Tilt Transform: this is a dynamic transformation that depends on 2 encoder values. It is used to modelize a perfect Pan/Tilt connection. Pan is a rotation around the Z axis, and Tilt a rotation around the Y axis. Click Map.. to attach sensors to the pan or tilt stage (see above).


Pan Sensor: Map…

Tilt Sensor: Map…

Pan:Connected to Simulation/Pan: 216
Tilt: Connected to Simulation/Tilt: 23

- Jib Transform: this is a dynamic transformation that depends on 2 encoder values Horizontal Boom and Vertical Boom. It is used to model a perfectly aligned parallelogram jib arm (this is the case of the Chrome robotic jib); this transform is more complex than a pan/tilt because you need to keep the parallelogram aligned.



HBoom:Connected to Simulation/Pan: 216
VBoom: Connected to Simulation/Tilt: 23

Click Map.. to attach sensors to the HBoom or VBoom stage. Enter the length (in meters) of the boom in Arm Length: 1.5 for the Chrome robotic Jib.

Sights

Sights are not used for computation internal parameters and calibration, it is obsolete: Use Video Player block and grab snapshots.

OBSOLETE:
Sights are used to compute some internal parameters of the “Mechanical transformations”. Most of the time sights are used at calibration time. A sight is achieved when the camera looks at a known target at the optical center (which is close but is not the image center); before doing the sight procedure the optical center procedure should have been done.

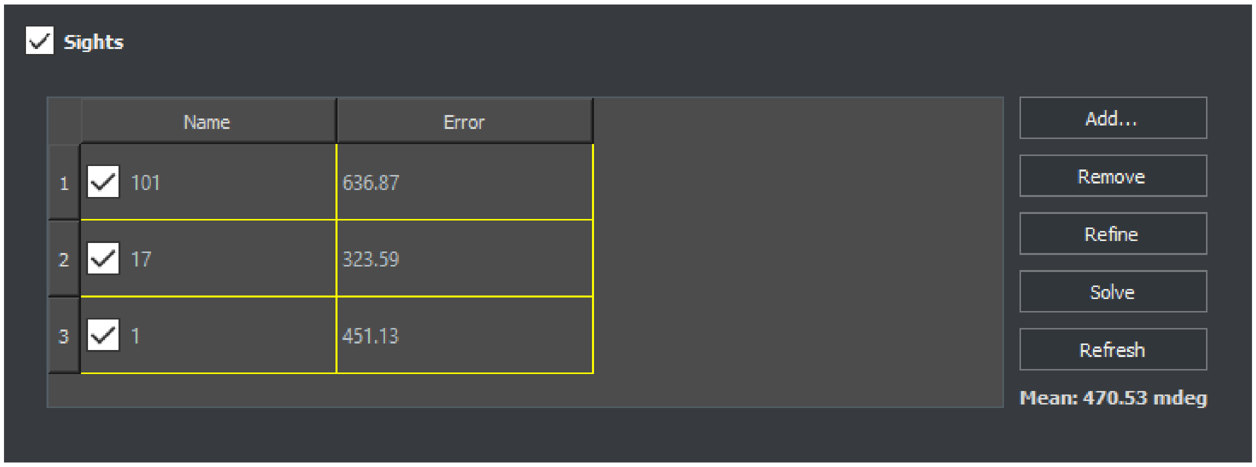
The sighting procedure can find parameters which are not lying on the optical center, for the same reason it cannot find lens parameters. In particular the last ‘TZ’ cannot be calculated with sights.

- You can Add/Remove sights by clicking on the corresponding buttons.
- Clicking ‘Refresh’ will recalculate all the errors. This may be necessary when values are externally changed.
- Clicking ‘Refine’ triggers parameter computations from the current parameters set.
- Clicking ‘Solve’ triggers the re-computation from scratch of the unknown parameters.

When a solution is found, it’s good practice to use ‘Refine’ only. If you drastically change some parameters, pressing ‘Solve’ could be a good idea.

Errors for each sight are given in 1/1000th degree. The same applies for the ‘Mean’ error.

When a Sight is unchecked, then it won’t be used for future computations.

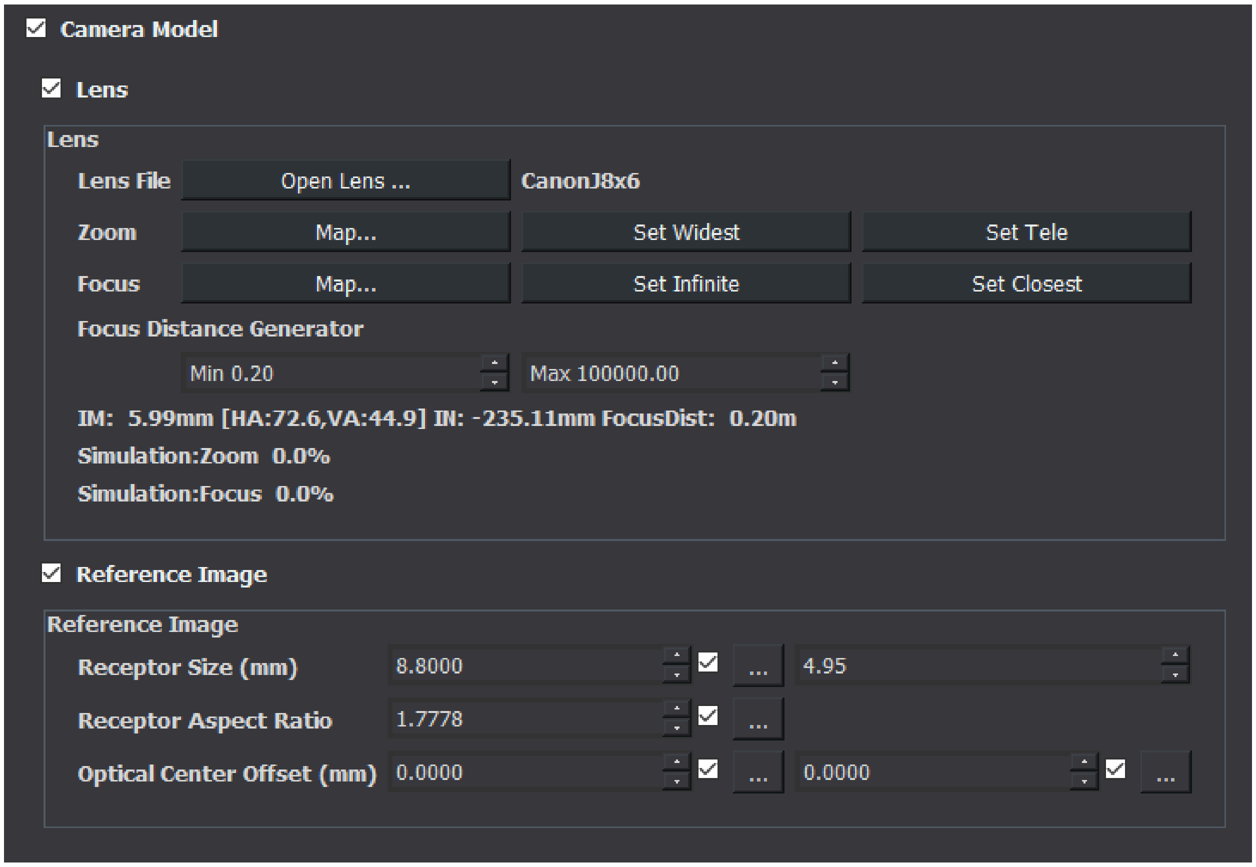


☑ Sights

		Name	Error
1	✓	101	636.87
2	✓	17	323.59
3	✓	1	451.13

Mean: 470.53 mdeg

ChyronHego Lens model



☑ Camera Model

☑ Lens

Lens

Lens File	Open Lens ...	CanonJ8x6	
Zoom	Map...	Set Widest	Set Tele
Focus	Map...	Set Infinite	Set Closest

Focus Distance Generator

Min 0.20 | Max 100000.00

IM: 5.99mm [HA:72.6,VA:44.9] IN: -235.11mm FocusDist: 0.20m

Simulation:Zoom 0.0%

Simulation:Focus 0.0%

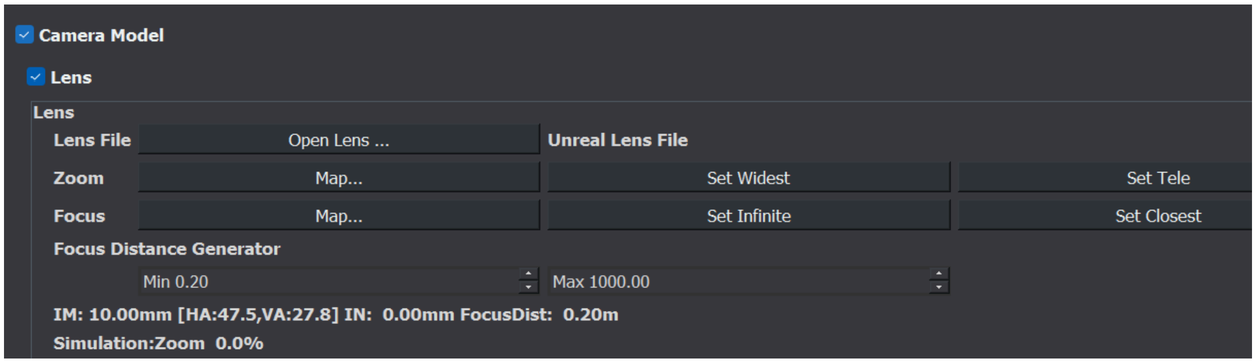
☑ Reference Image

Reference Image

Receptor Size (mm)	8.8000 ✓ ...	4.95	
Receptor Aspect Ratio	1.7778 ✓ ...		
Optical Center Offset (mm)	0.0000 ✓ ...	0.0000 ✓ ...	

The ChyronHego lens model : both the lens itself and the camera receptor (aka CCD). The Lens model uses a calibrated lens file that describes the lens parameters (focal length, distorsions, ...).

Unreal Engine Lens Model



IM: 10.00mm [HA:47.5,VA:27.8] IN: 0.00mm FocusDist: 0.20m
Simulation:Zoom 0.0%

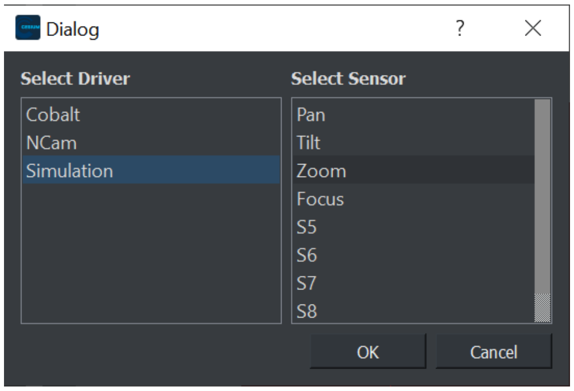
When using an Unreal Engine lens file (.ulens) loaded into the Cesium Camera within VSAR, the "Unreal Lens File.lns" file must be selected as Lens File within Cesium. This file is always located in *C:\Program Files\Chyron\Cesium\data\lenses\Misc.*

Lens

Open Lens... allows you to open a lens (.lns) file. When a file is properly loaded, then the lens' nickname is displayed on the right. In the above example, it's CanonJ8x6.

Zoom and Focus allow you to map the zoom and focus encoders to the lens calibration file.

Map... sets the encoder to which the zoom (resp. focus) is connected to. When clicking, a dialog appears where you select the desired encoder.



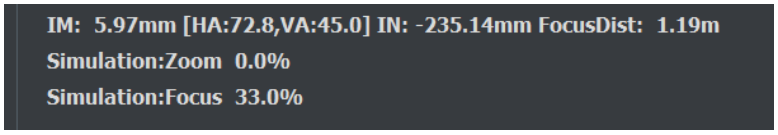
Set Widest, Set Tele allow you to easily map the zoom encoder values to the internal parameter of the lens file. To achieve this, do the 2 following steps in any order, but both steps have to be done.

- move the Zoom ring to the widest angle, shortest focal length: click Set Widest
- move the Zoom ring to the maximum zoomed value, longest focal length: click Set Tele.

Set Infinite, Set Closest allow to easily map the focus encoder values to the internal parameter of the lens file. To achieve this, do the 2 following steps in any order, but both steps have to be done.

- move the Focus ring to the infinite position: click Set Infinite
- move the Focus ring to the minimum focus value: click Set Closest.

When the lens encoders are properly mapped, and when the lens file is properly read, Cesium displays information about the current lens:



```
IM:  5.97mm [HA:72.8,VA:45.0] IN: -235.14mm FocusDist:  1.19m
Simulation:Zoom  0.0%
Simulation:Focus  33.0%
```

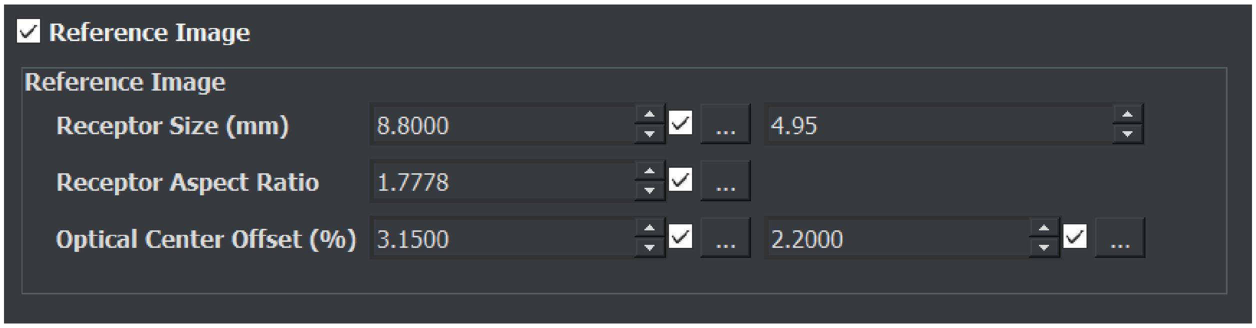
This information is read:

- IM is the focal length in mm. Here 5.97 mm
- HA and VA are respectively the horizontal and vertical angles in degrees. These angles depend on the receptor size (see below).
- IN is the nodal point shift in mm.
- FocusDist is the focus distance (not to be confused with focal length) in meters. If it's not available, then -1 is displayed. Here the focus distance is 1.19m.

Focus Distance Generator: allows to change the min and max values used to automatically compute the focus distance from the Focus Sensor, using an hyperbolic model. When the focus sensor is 0% the min value is returned, when the focus sensor is 100% the max value is returned. In between an hyperbolic interpolation is performed.

The horizontal slider allows setting the other parameter in normalized coordinates [0,100]. If X is chosen as Zoom (resp. Focus), then the slider allows setting the Focus (resp. Zoom) value. This is needed as the displayed parameters depend on both Zoom and Focus.

Reference Image



✓ Reference Image

Reference Image

Receptor Size (mm)	8.8000 ✓	4.95	
Receptor Aspect Ratio	1.7778 ✓		
Optical Center Offset (%)	3.1500 ✓	2.2000 ✓	

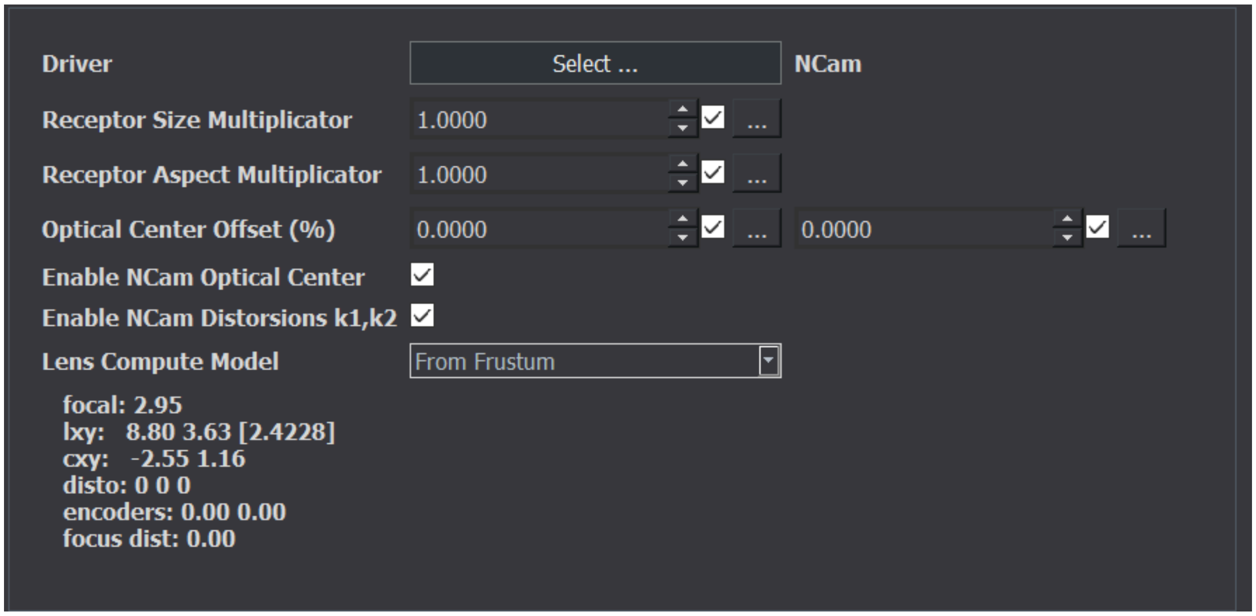
This tool describes the camera receptor (called CCD for video cameras).

Receptor Size is the actual size, horizontal and vertical, in mm of the CCD. Only the horizontal size is editable. Changing the vertical size is done by changing the …

Receptor Aspect Ratio is the ratio of the horizontal by the vertical receptor size. For HD video cameras, this value is close to 16/9 ~ 1.7777.

Optical Center Offset (%) modelizes the fact that the center of the image is most of the time not aligned with the optical axis. It's an offset in the receptor plane.

NCam Lens model



Driver	Select ...	NCam		
Receptor Size Multiplicator	1.0000	✓	...	
Receptor Aspect Multiplicator	1.0000	✓	...	
Optical Center Offset (%)	0.0000	✓ ...	0.0000	✓ ...
Enable NCam Optical Center	✓			
Enable NCam Distorsions k1,k2	✓			
Lens Compute Model	From Frustum			

```
focal: 2.95
lxy:   8.80 3.63 [2.4228]
cxy:   -2.55 1.16
disto: 0 0 0
encoders: 0.00 0.00
focus dist: 0.00
```

Driver: allows selecting the NCam driver connected to this lens model. NCam driver will deliver proper focal (among other parameters) to the lens model. The name of the currently connected driver is displayed on the right of the "Select…" push button (NCam in the above picture).

Receptor Size Multiplicator: allows to bias the NCam receptor size. Default is 1.0

Receptor Aspect Multiplicator: allows to bias the NCam receptor aspect ratio. Default is 1.0

Optical Center Offset: allows biasing the NCam optical center. Default is 0.0 0.0.

Enable NCam Optical Center: when checked the NCam optical center is used. If unchecked, NCam optical center is unused.

Enable NCam Optical Distortions: when checked NCam distorsions are used, if unchecked distorsions are disabled.

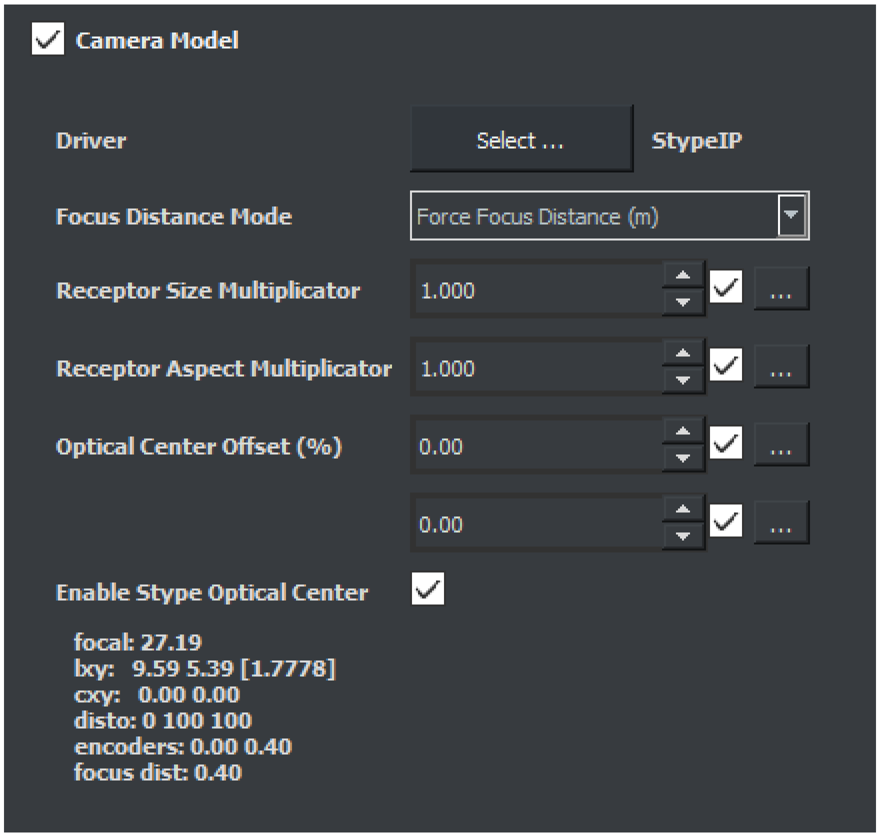
Lens Compute Model:

- From Aspect: Aspect is computed from the transmitted ratio. Optical center is computed from the transmitted optical center values.
- From CCD: Aspect is computed from the transmitted CCD Sizes. Optical center is computed from the transmitted optical center values.
- From Frustum: Lens parameters are computed from the transmitted LeftRight/Bottom/Top values. This setting allows you to handle over framed images. This is the default and recommended value by NCam.

Important Note: NCam emulator wrongly displays aspect ratio, you can mimic (even if it's not recommended) this behavior by enabling "Use buggy NCam Aspect" in the preference Dialog. Note that preferences are automatically saved, and reloaded when Cesium is started.

The computed values in Cesium units (focal length, CCD size, optical center, …) are displayed in the bottom of the widget.

Stype Lens model



Camera Model ✓

Driver	Select ...	StypeIP
Focus Distance Mode	Force Focus Distance (m)	
Receptor Size Multiplicator	1.000	✓
Receptor Aspect Multiplicator	1.000	✓
Optical Center Offset (%)	0.00	✓
	0.00	✓
Enable Stype Optical Center	✓	

```
focal: 27.19
bxy:   9.59 5.39 [1.7778]
cxy:   0.00 0.00
disto: 0 100 100
encoders: 0.00 0.40
focus dist: 0.40
```

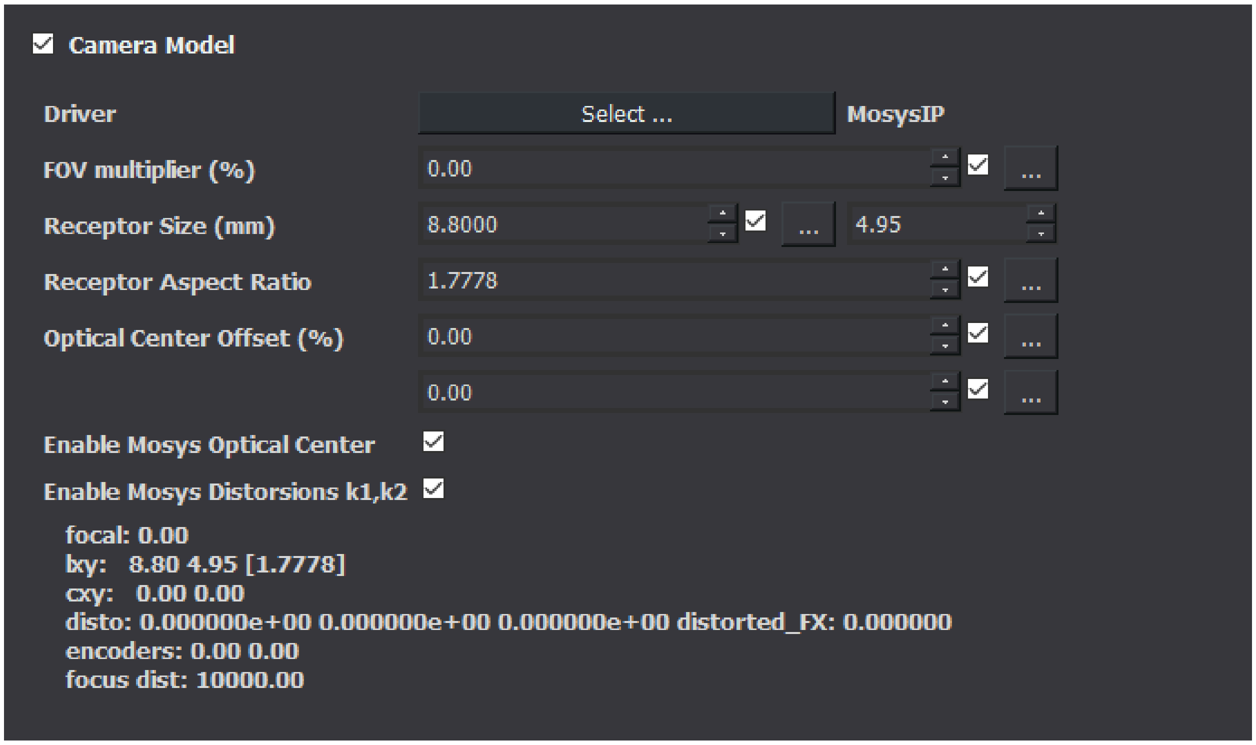
Driver: allows you to select the Stype driver connected to this lens. Stype driver will deliver proper focal (among other parameters) to the lens model. The name of the currently connected driver is displayed on the right of the “Select...” push button (Stype in the above picture). If no driver is connected then “Disconnected” is displayed.

Focus Distance Mode: allows to change the way the focus encoder is interpreted. Indeed, Stype has 2 modes that can be set mainly for RedSpy.

- Force Focus Distance: in this mode the focus encoder value will always be interpreted as a focus distance (in meters).
- Force Encoder: in this mode the focus encoder value will always be interpreted as the normalized encoder value.
- Auto: automatic detection between the 2 upper modes will be performed (it’s the default value).

The actual focus distance (in meters) sent to 3rd party rendering software is displayed in the ‘focus dist’ field.

Mo-Sys Lens model



☑ Camera Model

Driver	Select ...	MosysIP		
FOV multiplier (%)	0.00	☑	...	
Receptor Size (mm)	8.8000	☑	...	4.95
Receptor Aspect Ratio	1.7778	☑	...	
Optical Center Offset (%)	0.00	☑	...	
	0.00	☑	...	
Enable Mosys Optical Center	☑			
Enable Mosys Distorsions k1,k2	☑			

```
focal: 0.00
kxy:   8.80 4.95 [1.7778]
cxy:   0.00 0.00
disto: 0.000000e+00 0.000000e+00 0.000000e+00 distorted_FX: 0.000000
encoders: 0.00 0.00
focus dist: 10000.00
```

Driver: allows you to select the Mo-Sys driver connected to this lens. Mo-Sys driver will deliver proper focal (among other parameters) to the lens model. The name of the currently connected driver is displayed on the right of the “Select…” push button (Mo-Sys in the above picture). If no driver is connected then “Disconnected” is displayed.

FOV multiplier (%): changes the size of the FOV by % (example 10 means 10% if 100 FOV will result in 110 FOV)

Receptor Size: is the actual size, horizontal and vertical, in mm of the CCD. Only the horizontal size is editable.

Receptor Aspect Ratio: is the ratio of the horizontal by the vertical receptor size. For HD video cameras, this value is close to 16/9 ~ 1.7777.

Optical Center Offset (%): allows the change of optical axis by percentage. It’s an offset in the receptor plane.

Receptor Size Multiplicator: allows to bias the Mo-Sys receptor size. Default is 1.0

Receptor Aspect Multiplicator: allows to bias the Mo-Sys receptor aspect ratio. Default is 1.0

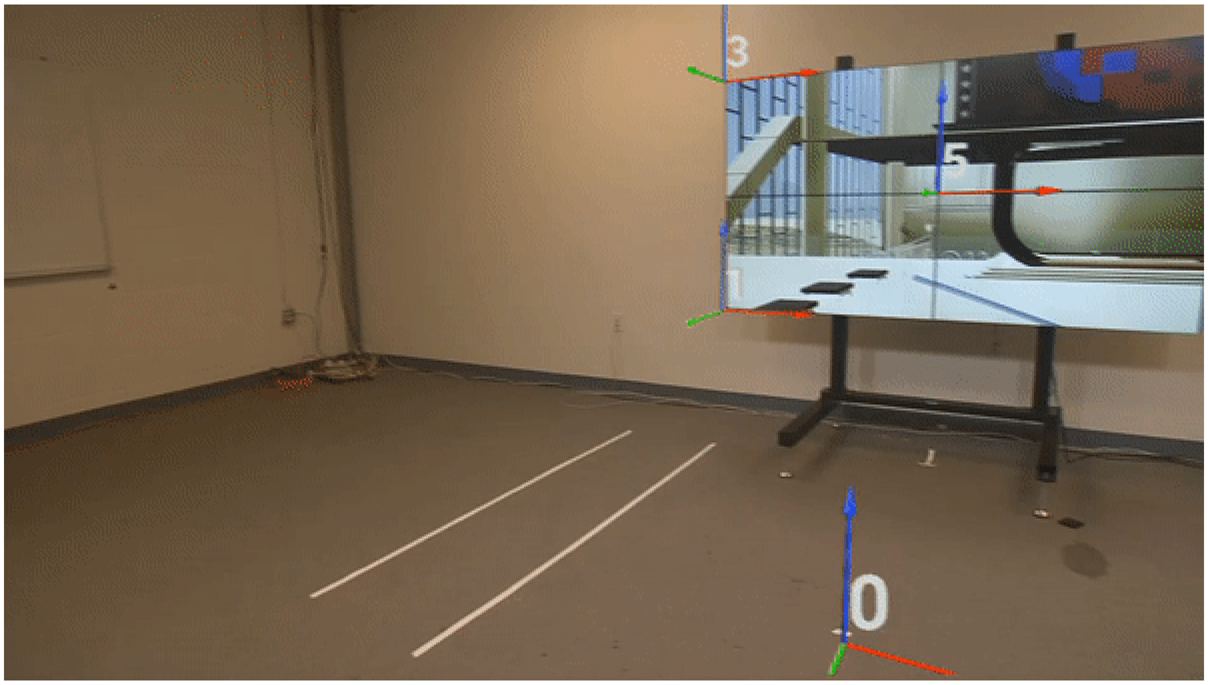
Optical Center Offset: allows the bias of the Mo-Sys optical center. The default is 0.0 0.0.

Enable Stype Optical Center: when checked the Mo-Sys optical center is used. Default is checked.

The different computed values in Cesium units (focal length, ccd size, optical center, …) are displayed in the bottom of the widget.

Parallax (aka Screen) Rigs

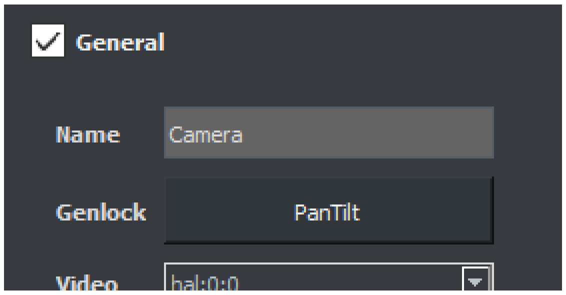
A Parallax (aka Screen) rig, is used to display an image in an on stage big screen. The image displayed in the screen is typically a virtual set, this screen is shot by the real camera, thus the image displayed in the screen should ‘in some way’ follow the camera movement. This is this ‘some way’ that the parallax rig takes care of. Indeed, the geometrical transformations are more complex than a simple mechanical rig.



A parallax rig is defined by:

- a mechanical rig, the actual tracked real camera
- the position of the big screen in studio space.

Set the tracked camera



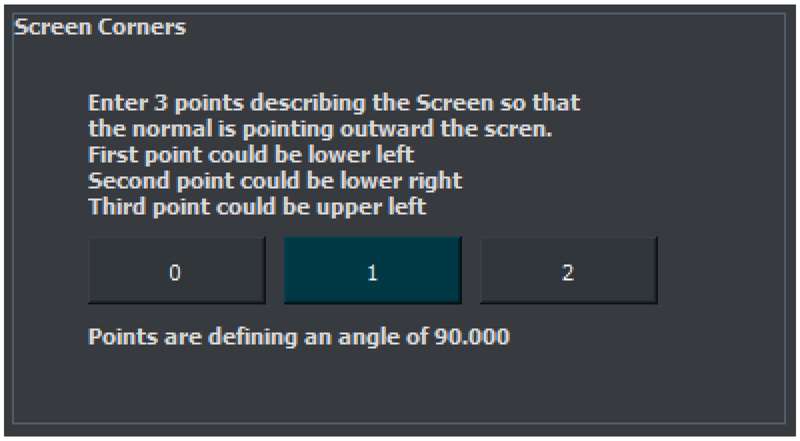
Name: Camera

Genlock: PanTilt

Video: hal:0:0

To set the mechanical rig to which the parallax rig is connected to, simply change the “Genlock” in the General tab of the Parallax Rig (when dealing with Parallax rigs, only another rig can then be selected). In the following picture the Parallax rig is dependent on the “PanTilt” rig.

Set the screen position



Screen Corners

Enter 3 points describing the Screen so that
the normal is pointing outward the scren.
First point could be lower left
Second point could be lower right
Third point could be upper left

Points are defining an angle of 90.000

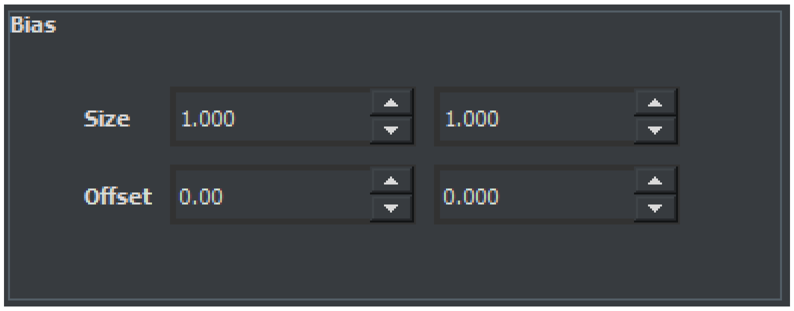
To set the position of the screen, select 3 targets that describe that screen, and fill the "Screen Corners". The points should be set with the correct winding, for example:

- lower left -> lower right -> upper left
- lower right -> upper right -> lower left
- upper right -> upper left -> lower right
- upper left -> lower left -> upper right

When the points are set, Cesium checks that they make a rectangle by displaying the angle between them. Obviously it should be around 90deg. The following picture shows a valid screen made of the targets 0,1 and 2.

Note: even if the angle is not 90deg the calculations will be done, but may lead to unexpected results. The closer to 90deg the better.

Fine tune the receptor size and optical center



Bias

Size	1.000	1.000
Offset	0.00	0.000

In normal operation, the resulting optical center and receptor size are computed from the tracked camera and the screen position. Nevertheless, to add flexibility, you can tune those parameters:

Bias Size adds a multiplicative factor to the parallax receptor size. Default value is 1. 1.

Bias Offset adds an additive offset to the parallax optical center. Default value is 0. 0.

Creating Rigs

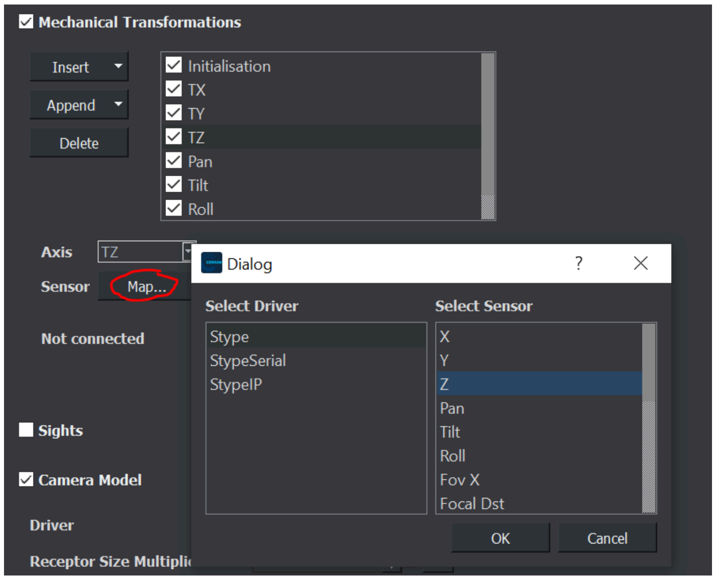
There are several pre-packed rigs that you can create. Get in Menu>Rig>Add>

- Pan/Tilt head: Creates a standard Pan/Tilt head rig. The default Lens Model is set to Chyron. You just need to connect the sensor to the Pan and Tilt encoders, and to fill the necessary information in the Lens Model.
- Pedestal: Creates a standard Pan/Tilt head rig with X, Y, Z sensor position . The default Lens Model is set to Chyron. You just need to connect the sensor to the Pan and Tilt encoders, and to fill the necessary information in the Lens Model.
- Jib (or Silver Head): Creates the necessary mechanical stages for a HBoom/VBoom Jib and it's Pan/Tilt head. The default lens model is set to Chyron. Once created, fill the Boom, PanTilt and Lens with the appropriate encoder source.
- Advanced Pan/Tilt, Pedestal and Jib: In the Advanced tab, you can create Pan/Tilt, Pedestal and HBoom/VBoom Jibs with all the necessary steps that can be then customized. For example the jib parallelogram is split in 5 transforms instead of a single transform for the non advanced model. This allows you to introduce new transformations in between when necessary.
- Parallax aka Screen: Creates a Parallax rig (see above)
- NCam: Creates a NCam compatible Mechanical Rig. The default Lens Model is set to NCam. You just need to connect the 6 TXYZ and RXYZ to the appropriate NCam source. You may also need to fill the necessary information in the Lens Model, in particular fill the Driver information..

An "Initialisation" stage is automatically inserted to the calibration; depending on your usage, you can:

  - leave all the values to 0, thus using the Stype coordinate system
  - modify manually the values, to use another coordinate system.
  - do sighting to automatically fill these values to use your own targets coordinate system.

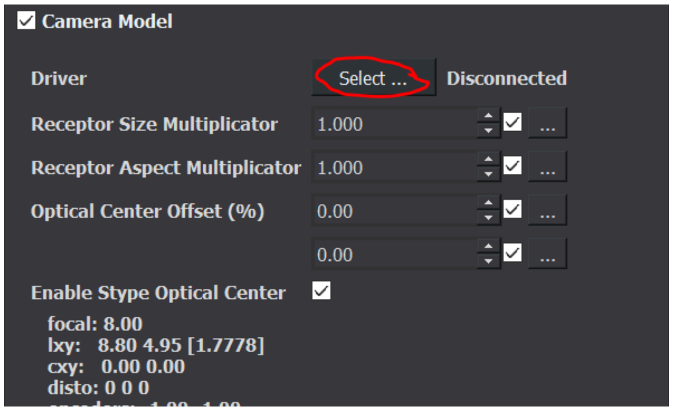
- Stype: Creates a predefined Stype compatible Mechanical Rig. The default Lens Model is set to Stype. You need at least to:
    - Select the Stype driver as the Genlock source.
    - Connect the 6 mechanical transforms TX TY TZ RX RY RZ to the relevant encoders in Stype driver. The picture below shows how to connect the TZ transform to the corresponding encoder.



Axis: TZ
Sensor: Map...
Not connected

Select Driver: Stype, StypeSerial, StypeIP
Select Sensor: X, Y, Z, Pan, Tilt, Roll, Fov X, Focal Dst

- Select the Stype driver to feed the Stype camera model.



Driver: Disconnected
Receptor Size Multiplicator: 1.000
Receptor Aspect Multiplicator: 1.000
Optical Center Offset (%): 0.00
0.00
Enable Stype Optical Center: ✓
focal: 8.00
lxy: 8.80 4.95 [1.7778]
cxy: 0.00 0.00
disto: 0 0 0

An “Initialisation” stage is automatically inserted to the calibration; depending on your usage, you can:

- leave all the values to 0, thus using the Stype coordinate system
- modify manually the values, to use another coordinate system.

    - do sighting to automatically fill these values to use your own targets coordinate system.
- Mo-Sys: Creates a predefined Mo-Sys compatible with Mo-Sys IP Driver. configuration is similar to Stype Rig.

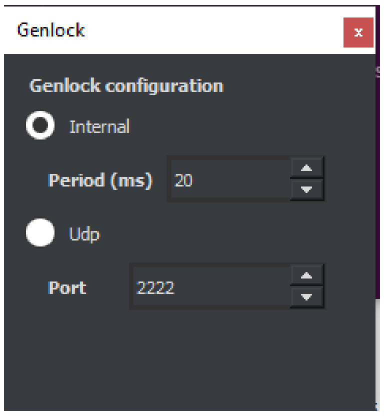
Genlock Configuration

Rigs emit data to 3rd party rendering software whenever they are triggered by an external event. This event is called a synchronization event, and the synchronization process is called genlock.

- Parallax rigs are genlocked to a mechanical rig.
- Mechanical rigs are genlocked to a Driver.
- Drivers are genlocked either to:
    - The underlying hardware they are connected to (for example, Chyron Cobalt, Stype, NCam, MoSys, …)
    - The internal Cesium genlock mechanism. This is mainly true for Simulation drivers.

This chapter describes the internal Cesium genlock configuration.

To open the genlock configuration tools, open the menu “Tools>Genlock Configuration …”. Genlock settings are saved in the configuration file whenever you do a modification in the dialog. Genlock settings are not attached to the .cs project you are working on.


Genlock

Genlock configuration

[✓ Internal]

Period (ms) 20

Udp

Port 2222

Internal Genlock

In this mode you can set the period (Period (ms)) at which you want sync pulses to be emitted. 20 milliseconds corresponds to a refresh rate of 50fps. If you want to be close to 60fps, enter 17 ms.

Important note: this mode is an INTERNAL genlock, it's absolutely impossible to synchronize any equipment with it (cameras, mixer, …).

Udp Genlock

In this mode, Cesium is waiting for synchronization messages on a given UDP port (default 2222). This mode allows synchronization with external equipment but most of the time this requires specific development. In particular, Prime VSAR is able to emit such messages.

Needs configuration to be set in VSAR Hal.xml (AppData\Local\Chyronhego\Fresh\Hal.xml): tcpcesium="true" cesiumip="127.0.0.1" cesiumport="2222" for output.

Targets

As stated earlier (**Targets**), while this is not a strictly necessary step, it is still recommended to be done, especially for helping controlling and creating calibration or when working with multiple cameras.

A Target System is a set of reference points surveyed in the physical studio.

Creating the 3D targets

Some criteria for quality of the 3D target set are :

The targets should be visible, stable and precisely identified. They should be as close as possible to the area where action is going to take place (the blue-screen), and should fill the 3D space as much as possible.

Unfortunately, some of these qualities may be difficult to get simultaneously. Filling the 3d space in the blue-screen means, for instance, that some targets are to be placed on the floor and on the walls. These targets have then the following requirement, hard to fulfill because paradoxically: to be visible since they are targets and to be invisible since they are in the field of view of the camera. It might also be quite difficult to keep them stable in the middle term, since blue-screens are regularly re-painted. However, imaginative solutions can always be found, involving gaffer and drills.

Finally, it is wise to draw a sketch of target positions and labels (and to keep the document in a safe place) for future use.

Measuring the 3D targets

In the general case, the 'theodolite/telemeter' is the perfect instrument for the measurement of the targets. This instrument, used by surveyors for field survey, is simple to learn, precise and fast to operate. Furthermore, one can inexpensively rent this equipment in most cities everywhere in the world.

At the end of the measuring process, one should write an ASCII file, with the suffix .tgt including for each target :

- a label, which must be a string not containing spaces (previous releases of Cesium need that this label is a positive integer).
- the x coordinate (meters),(In most cases this coordinate will represent a right direction of the camera's view)
- the y coordinate (meters),(In most cases this coordinate represents up/down direction of the cameras's view)

- the z coordinate (meters),(In most cases this coordinate will represent forward direction of the camera's view)
- Scale of the coordinates: when creating the coordinates this has to be number 1, for compatibility with VSAR. (This value is not used but required for backwards compatibility reasons).

Since Cesium uses the metric system, coordinates must be expressed in meters. In the data/target directory, a sample target file, sample.tgt, is given which can be used as a template for new target files.

Target values are separated by end of line, when multiple targets are on one line only one is read.

The values can be separated either by space or tab.

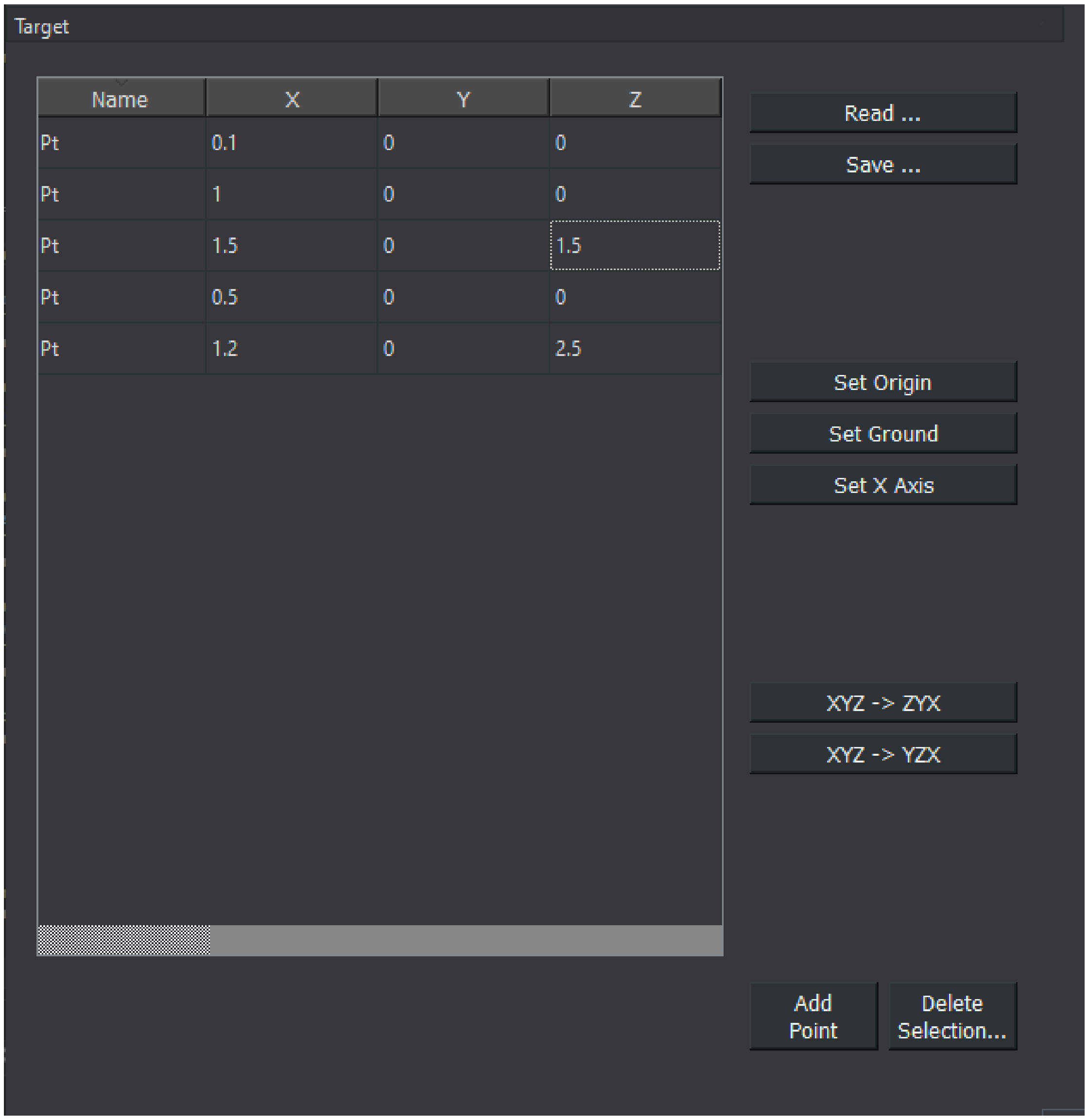
Example of target file content:

```
0       0           0.073962     0          1
1       0.11585     0            2.82766    1
2       -0.58404    -0.0063484   2.83833    1
3       -0.57676    -0.0029638   2.32839    1
10      1.74101     0.0823975    -2.00817   1
102     -0.86777    0.324715     2.17128    1
103     -0.96777    0.424715     2.14628    1
105     -5.08277    0.424715     2.14628    1
106     -5.18277    0.324715     2.17128    1
201     0.12199     0.999682     2.85215    1
202     -0.57604    0.992656     2.88221    1
```

Loading a target file in Cesium

To load a target file in Cesium, select Tools>Target... This will open the Target Dialog window.

In this window, select Read... which opens a browser to load the target file (.tgt).

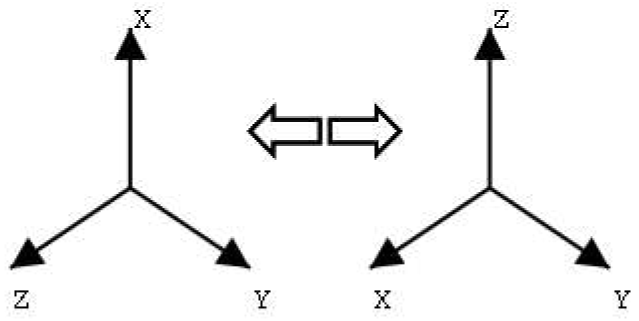


Name	X	Y	Z
Pt	0.1	0	0
Pt	1	0	0
Pt	1.5	0	1.5
Pt	0.5	0	0
Pt	1.2	0	2.5

The list of targets (Name, x,y,z coordinates) is now displayed.

Target editing

As said earlier, Cesium assumes that the World Coordinate System, as any other coordinate system used in Cesium, is a right-handed Coordinate System Remember: when using your right hand[2], X axis is your thumb, Y axis your index finger, Z axis your middle finger. In most cases, coordinates generated by a theodolite are oriented the right way, with the Z axis vertical, but this may not be true with other equipment (a ruler, for example, together with some skills in Cartesian geometry).



The XYZ -> ZYX or XYZ -> YZX buttons of the TargetDialog window allows you to permute coordinates in your target file.

If your world coordinate system is not left-handed, click XYZ -> ZYX. This transforms a left-handed coordinate system into a right-handed one, and conversely.

Once your coordinate system is right-handed, you should have the Y axis vertical with positive direction upward, because this will show to be very convenient in Cesium calibration.

Use the circular permutation XYZ -> YZX button. Remember, circular permutation of axes keeps the orientation of the coordinate system.

You may also want to move the origin of your coordinate system to some specific target in the list. To do so, select one or several target(s) by clicking on the corresponding line (use Shift and Ctrl for multi line editing) then click Set Origin. All coordinates in the list are changed so that all the selected targets are as close as possible to the origin..

In the same manner you can also select a list of targets to define the ground of the studio floor. Use the Set Ground button to achieve this. To do this, you may need to sort the targets by the Y coordinate (click on the Y column to do so).

You may now save the target file for future use, by clicking the Save... button..

Configuration file

General configuration

Cesium has a configuration/preference file, default location is ${HOME}/.hmc/cesium/cesium.ini.

If no file is already present a new one is created with the default parameters. So to get the actual list of available parameters, a solution is to remove/move the current configuration file, and relaunch cesium.

Here is a default Cesium configuration file, comments are added after each block

[dataengine] describes the connection to the data engine. Currently the data engine is only used to remote control cesium.

```
[dataengine]
hostport=127.0.0.1:4300
lua=hmc.cesium/lua
statsUpdate=1
```

- hostport stands for the host and port onto which data engine is running.
- lua is the bucket/key into which cesium is reading the lua commands to execute.
- statsUpdate is obsolete.

[default] describes the default file to load on startup. The default computed filename is “file/prefix”.

```
[default]
file=
prefix=${HOME}/Projects/Cesium
```

[file] sets how images from “Video Player” are saved

```
[file]
saveImagesInline=true
```

- if saveImagesInline is true, grabbed images are saved inside the .cs file. If false images are saved as independent files on disk. Setting to true can lead to very large .cs files, but it’s easier to transfer; setting to false leads to faster loading times. Default is true.

[genlock] describes the configuration of the global genlock. These values can be changed by using the Tools>Genlock Configuration… tool.

```
[genlock]
internalPeriod=20
source=0
udpPort=2222
```

- internalPeriod is the period in milliseconds of the internal genlock. Default is 20ms and corresponds to a refresh rate of 50fps.
- source, 0 stands for internal, and 1 stands for Udp
- udpPort is the network port onto which cesium is reading its sync messages.

[gui] enables/disables specific features.

```
[gui]
show_developper=false
show_experimental=false
```

- show_developper displays debugging features.
- show_experimental enables experimental features that may not be supported in the future.

[logging] configures the logging subsystem.

```
[logging]
config=${HOME}/.hmc/logging.ini
```

- config is the path to the logging configuration file.

[lua] configures the lua sub-system. Lua subsystem is used to remote control cesium.

```
[lua]
main=${HOME}/.hmc/cesium/main.lua
```

- main contains the path of the lua file to execute upon startup.

[General] global configuration

```
[General]
recent=${HOME}/.hmc/cesium/recent.txt
```

- recent contains the path of the file that contains the recently opened/saved file names.

[renderer] contains all the rendering configuration, some of these parameters are settable in the Preference Dialog.

```
[renderer]
cpuEnabled=false
focalMultiplier=1
gpuEnabled=true
shaderPath=${BINDIR}/shaders
onAirTimeoutMinute=10
```

- if cpuEnabled is true then the display of graphics elements will be done using the CPU. Some GPU specific features (like stype test patterns) won't be displayed in this mode. Default is false.
- if gpuEnabled is true then the display of graphic elements will be done using the GPU. Default is true.
- focalMultiplier allows to manually add an extra-focal angle bias to the camera to have a better display of distorsions on the borders of the image. Default is 1.
- shaderPath is the pathname to the GPU shaders. Changing this value only makes sense for debugging purposes. Default is ${BINDIR}/shaders
- onAirTimeoutMinute defines the timeout (in minutes) after which the OnAir mode is automatically clutched. Default is 10 minutes. Floating point values are accepted. The minimum value is set to 0.05 (~3 seconds).

[rig] Rig specific configuration

```
[rig]
has_screen_distorsions=true
ncamBuggyAspect=false
screenHasNoDistos=true
stypeDisplayTestPattern=false
stypeDistorsionCorrectionFactor=1
```

- ncamBuggyAspect: please refer to NCam chapter.
- screenHasNoDistos: please refer to Screen (aka Parallax) rig chapter.
- if stypeDisplayTestPattern is true, then the Stype tests patterns are displayed in the viewer windows (they consist of 4 rectangles). Setting this value to true only makes sense for debugging purposes. The default is false.

[server] Communication configuration (see also [dataengine])

```
[server]
BusEnabled=false
DEEnabled=false
port=7100
port2=7101
```

- DEEnabled enables or disables the data engine communication.
- port is the port onto which cesium is delivering its data to 3rd party rendering engines. This communication is always enabled. Messages sent over this port are original messages.
- port2 is the port onto which cesium is delivering its data to 3rd party rendering engines in protobuf format. This communication is always enabled. This is the recommended and more recent version of the communication protocol.

[solver] configures the mathematical solver

```
[solver]
autoRefine=true
maxIteration=4000
numPass=1
relTolerance=1
verbose=0
usenlopt=false
```

- if autoRefine is true, then whenever a new sight is added/removed a calculation is ran.
- maxIteration defines the maximum number of iterations before a computing stage stops. The lower the faster, but if set too low you may miss the solution.
- use the nlopt library to solve
- numPass defines the number of computing stages; you can leave this value to default.. Default value is 1.
- relTolerance is the threshold to stop the calculations. The lower, the longer the calculation may be, but the better the found solution can be.
- verbose set the verbosity level of the solver.

[system]

```
[system]
halHasDebugMessages=false
```

- if halHasDebugMessages is true, then warning and error messages are logged. Setting this value to true makes sense for debugging purposes, it could be very very verbose. The default is false.

[video] configures the video subsystem

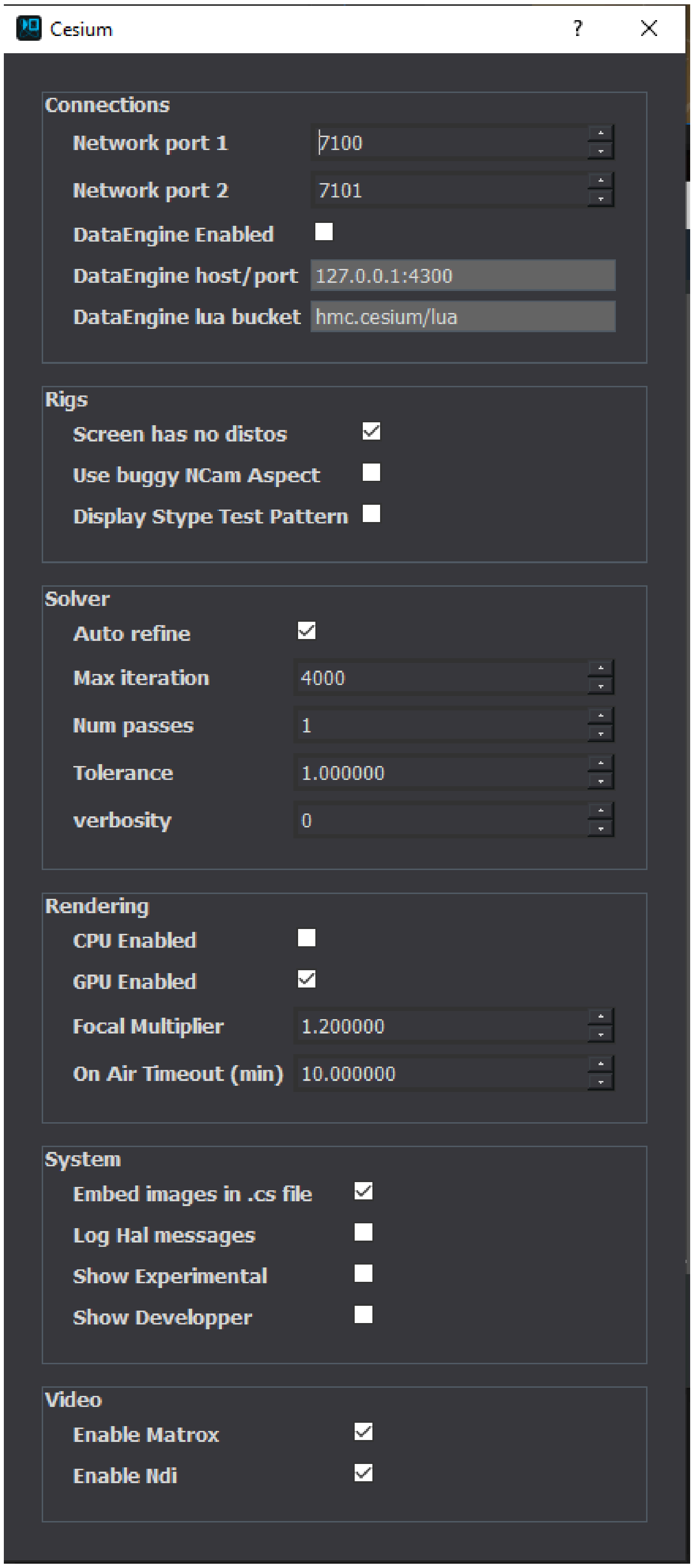
```
[video]
halNdiPixelFormat=YCrCb_422_8b
halNdiVideoFormat=1080I_2500
halconfig=${HOME}/.hmc/cesium/Hal.xml
halMatroxEnabled=true
halNdiEnabled=true
```

- halconfig is the filename of the Hal configuration file. See HAL Configuration file below.
- halNdiPixelFormat sets the pixel format used to grab images from NDI streams. YCrCb_422_8b is used to read RGB feeds. The other mode is RGBA_8b to read fill + key streams. Default is YCrCb_422_8b.
- halNdiVideoFormat=1080I_2500 sets the video format NDI is going to read. Available formats are listed in “List of recognized videoStandard” below. The default is 1080I_2500.
- halMatroxEnabled=true when set to false, all the Matrox inputs are disabled, whatever is set in the HAL.xml file. This is useful not to interfere with VSAR inputs for example.
- halNdiEnabled=true when set to false, all the NDI inputs are disabled, whatever is set in the HAL.xml file.

Preference Dialog

The preference Dialog, which is accessible through the menu bar “Tools>Preferences…”, allows you to set some configuration parameters. When you change a parameter in the Preferences Dialog, it is immediately saved in the configuration file.

The different parameters have been discussed in the previous section.


Cesium

Connections

Network port 1	7100
Network port 2	7101
DataEngine Enabled	☐
DataEngine host/port	127.0.0.1:4300
DataEngine lua bucket	hmc.cesium/lua

Rigs

Screen has no distos	✓
Use buggy NCam Aspect	☐
Display Stype Test Pattern	☐

Solver

Auto refine	✓
Max iteration	4000
Num passes	1
Tolerance	1.000000
verbosity	0

Rendering

CPU Enabled	☐
GPU Enabled	✓
Focal Multiplier	1.200000
On Air Timeout (min)	10.000000

System

Embed images in .cs file	✓
Log Hal messages	☐
Show Experimental	☐
Show Developper	☐

Video

Enable Matrox	✓
Enable Ndi	✓

Hal Configuration file

“Hal configuration file” syntax is compatible with the syntax of “VSAR Hal configuration file” with the following noticeable difference:

- “4k2si” attribute is not supported by Cesium because it starts with a non-letter character. Rename it to f4k2si (for example) if necessary.

This parameter was supported for Deltacast SDI card, Matrox SDI card does not support it.

4k mode could be possible prepare in Matrox DSX LE4 card : 4x1080p in 4 quadrants

we recommend to not use the parameter 4k2si in Hal.xml

Note that the syntax of “VSAR Hal configuration file” is not necessarily compatible with “Cesium Hal configuration file”.

File syntax

CESIUM HAL.XML CONFIGURATION FILE:

```
<halconfig>
 <board>
          <genlock standard="S259M_PAL" source="internal"/>
                    <!– Outputs →
                    <output standard="S274M_1080i_50" enabled="true" haskey="false"/>
                    <!– Inputs →
          <input standard="S274M_1080i_50" enabled="true" haskey="false" fifo="3" timeout="50"/>
          <input standard="S274M_1080i_50" enabled="true" haskey="false" fifo="3" timeout="50"/>
    </board>
</halconfig>
```

VSAR HAL.XML CONFIGURATION FILE:

```
<halconfig>
  <board>
    <genlock standard="S259M_PAL" source="internal"/>
    <!-- Inputs →
    <input standard="S274M_1080i_50" enabled="true" haskey="false" fifo="3" timeout="50"
useassync="false" />
    <input standard="S274M_1080i_50" enabled="true" haskey="false" fifo="3" timeout="50"
useassync="false" />
    <!-- Outputs →
    <output standard="S274M_1080i_50" enabled="true" haskey="false" hphase="0" vphase="0"
fifo="3" />
  </board>
</halconfig>
```

List of recognized videoStandard (aka standard):

```
// SD

"576I_2500", "S259M_PAL"         /** 720  x 576  25            Interlaced */

  -> Cesium SDI In : No Access

"486I_2997", "S259M_NTSC"        /** 720  x 486  30/1.001 Interlaced */

 -> Cesium SDI In : No Access

// HD

"720P_5000","S296M_720p_50"  /** 1280 x 720  50         Progressive */

"720P_5994", "S296M_720p_59" /** 1280 x 720  60/1.001 Progressive */

"720P_6000", "S296M_720p_60"  /** 1280 x 720  60        Progressive */

"1080I_2500”, "S274M_1080i_50"           /** 1920 x 1080 25        Interlaced */

"1080I_2997", "S274M_1080i_59"           /** 1920 x 1080 30/1.001 Interlaced */

"1080I_3000", "S274M_1080i_60"           /** 1920 x 1080 30        Interlaced */

"1080P_2500", "S274M_1080p_25"          /** 1920 x 1080 25        Progressive */

"1080P_2997", "S274M_1080p_29"          /** 1920 x 1080 30/1.001 Progressive */

"1080P_3000", "S274M_1080p_30"          /** 1920 x 1080 30        Progressive */

// 3G
```

```
            "1080P_5000",                    /** 1920 x 1080 50        Progressive */
             -> Cesium SDI In : No Access
            "1080P_5994",                    /** 1920 x 1080 60/1.001 Progressive */
             -> Cesium SDI In : No Access
            "1080P_6000",                    /** 1920 x 1080 60        Progressive */
             -> Cesium SDI In : No Access
            // 4K
            "2160P_2500", “S4K_2160p_25”  /** 3840 x 2160 25        Progressive */
-> Cesium SDI In : No Access
            "2160P_2997", “S4K_2160p_29”  /** 3840 x 2160 30/1.001 Progressive */
            -> Cesium SDI In : No Access
            "2160P_3000", “S4K_2160p_30”  /** 3840 x 2160 30        Progressive */
            -> Cesium SDI In : No Access
            "2160P_5000", “S4K_2160p_50”  /** 3840 x 2160 50        Progressive */
            -> Cesium SDI In : No Access
            "2160P_5994", “S4K_2160p_59”  /** 3840 x 2160 60/1.001 Progressive */
              -> Cesium SDI In : No Access
            "2160P_6000", “S4K_2160p_60”  /** 3840 x 2160 60        Progressive */
           -> Cesium SDI In : No Access
            // HD 720 lo-fps
            "720P_2500",                          /** 1280 x 720  25         Progressive */
           -> Cesium SDI In : No Access
            "720P_2997",                          /** 1280 x 720  30/1.001 Progressive */
           -> Cesium SDI In : No Access
            "720P_3000"                           /** 1280 x 720  30         Progressive */
            -> Cesium SDI In : No Access
```

VSAR Only:

```
// FILM FORMATS HD
"SFILM_720p_23",            /** Progressive */
"SFILM_720p_24",            /** Progressive */
"SFILM_1080p_23",           /** Progressive */
"SFILM_1080p_24",           /** Progressive */
-> VSAR SDI In/Out : PASS
-> Cesium, Haltest : Not Active
// FILM FORMATS 4K
"SFILM_2160p_23",           /** Progressive */
"SFILM_2160p_24",           /** Progressive */
-> VSAR SDI In/Out : PASS
-> Cesium, Haltest : Not Active
```

List of recognized genlock sources

```
"Internal", "internal"          // Internal genlock
"Analog", "analog"              // External analog genlock
"Input0"            // SDI input 0
"Input1"            // SDI input 1
"Input2"            // SDI input 2
"Input3"            // SDI input 3
"Input4"            // SDI input 4
"Input5"            // SDI input 5
"Input6"            // SDI input 6
"Input7"            // SDI input 7
```

Calibration

Calibration process consists of finding internal parameters of an articulated Rig. It can be used for example when:

- the user moves the pedestal of a pan/tilt system. In this case we need to find the new location of the pedestal
- the user installs a brand new camera system and wants to fine tune some internal parameters, such as small internal angles that can just be estimated. For example, when you set a camera on a pan/tilt system, there is a slight rotation of the camera with respect to the camera head.

There are 2 main ways of doing calibration that are mutually exclusive, both of them are using a set of known points (X,Y,Z coordinates) in the studio space called 'Targets'.

- Sights: consists of aiming at some targets with the camera system. This mode doesn't need to route the camera video signal to cesium. This mode is unable to find lens related parameters since it works only on the optical axis.This mode cannot be used offline.
- Trackers: powerful calibration mode which allows to fine tune some internal parameters, including lens parameters. Can be used offline. Needs camera video signal routed to cesium.

	Fine Tuning	Remote Support	Lens Parameters	Needs Camera Video In
Sights	No	Impossible	No	No
Trackers	Yes	Yes	Yes	Yes

In the following we will emphasize on Trackers based calibration, for Sight based calibration, refer to Sights.

Trackers based calibration case study

The calibration process is an iterative process where you first start by finding rough parameters that you will quickly fine tune. Most of the time you will have to:

- find rough initialisation estimation
- introduce unknowns little by little until you reach a satisfactory result.

Targets

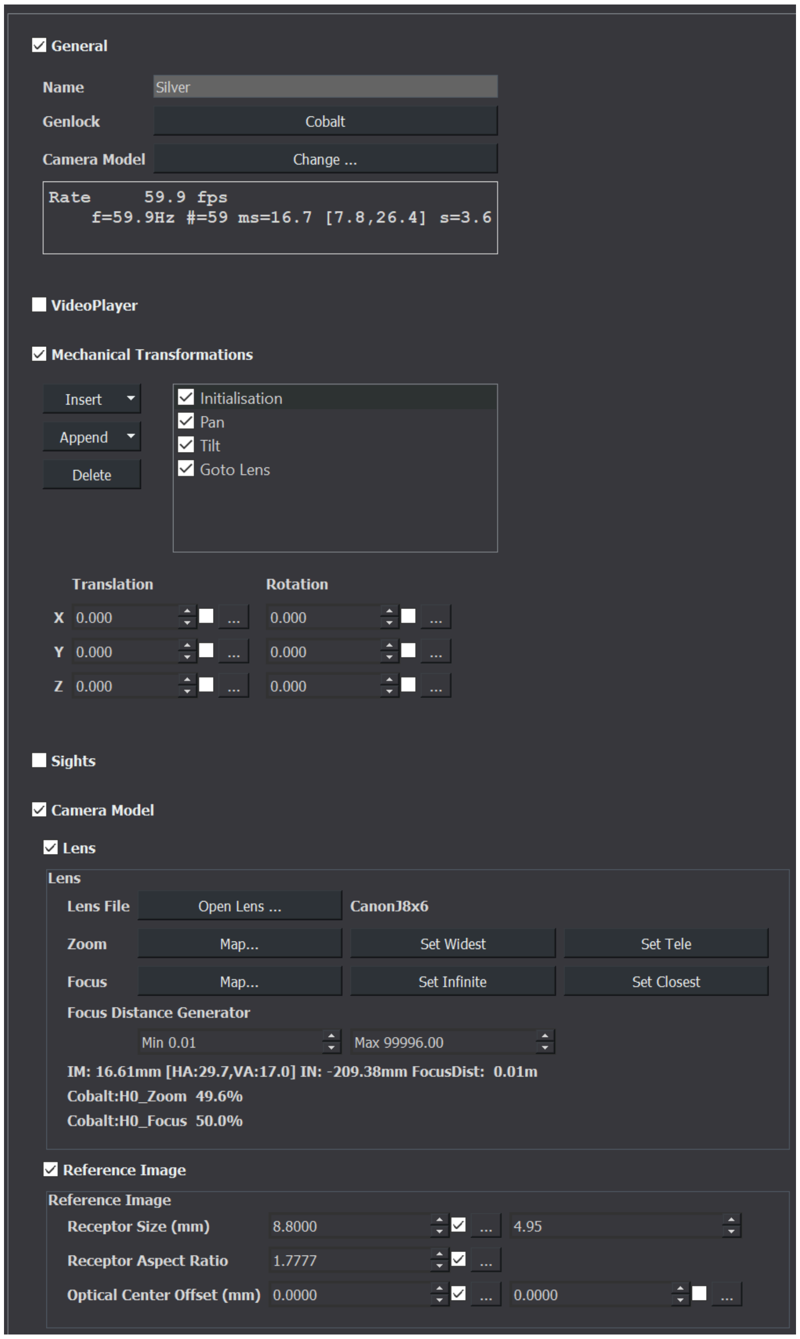
Let's suppose you have a valid set of targets. In our example we have 21 targets labeled 0 to 20.

Rig description

We are working with a rig consisting of:

- tracked pan/tilt head (namely Silver), with 4 stages: "Initialization", "Pan", "Tilt" and "Goto Lens"
- Chyron Camera model with a Canon lens

We suppose that all the sensors are properly mapped and connected to their respective transforms including lenses (genlock, Pan, Tilt, Zoom, Focus sensors are mapped and appropriate lenses are selected).



☑ General

Name	Silver
Genlock	Cobalt
Camera Model	Change ...

```
Rate      59.9 fps
    f=59.9Hz #=59 ms=16.7 [7.8,26.4] s=3.6
```

☐ VideoPlayer

☑ Mechanical Transformations

☑ Initialisation
☑ Pan
☑ Tilt
☑ Goto Lens

	Translation	Rotation
X	0.000	0.000
Y	0.000	0.000
Z	0.000	0.000

☐ Sights

☑ Camera Model

☑ Lens

Lens

Lens File: Open Lens ... CanonJ8x6

Focus Distance Generator: Min 0.01, Max 99996.00

IM: 16.61mm [HA:29.7,VA:17.0] IN: -209.38mm FocusDist: 0.01m
Cobalt:H0_Zoom 49.6%
Cobalt:H0_Focus 50.0%

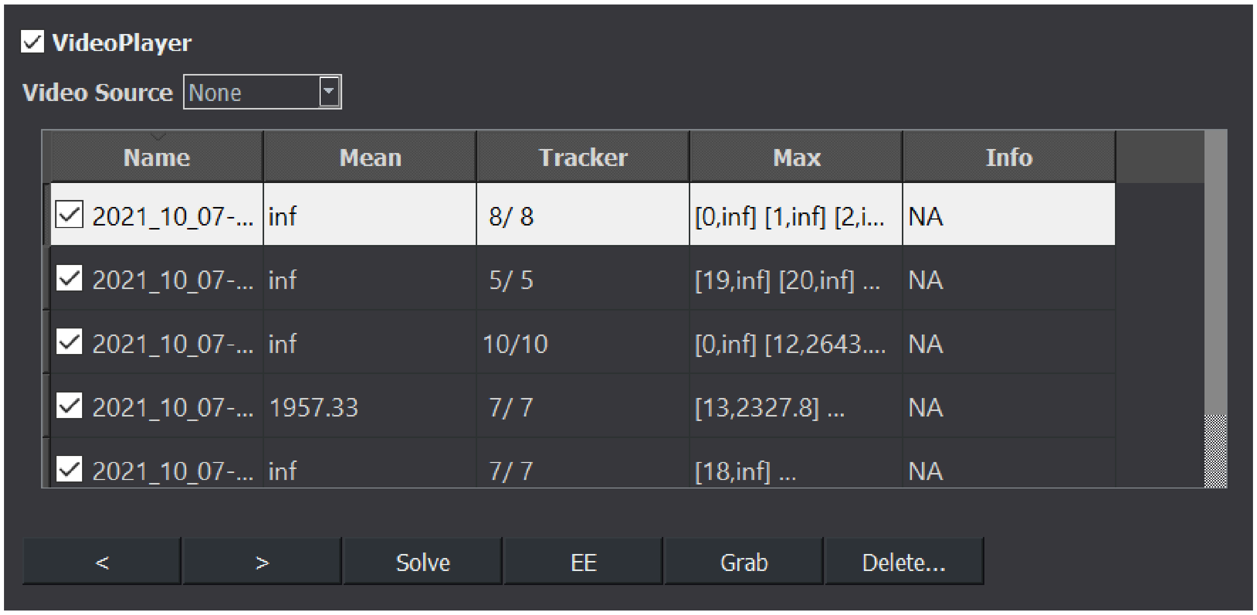
☑ Reference Image

Reference Image

Receptor Size (mm)	8.8000 ☑	4.95
Receptor Aspect Ratio	1.7777 ☑	
Optical Center Offset (mm)	0.0000 ☑	0.0000 ☐

Trackers picking

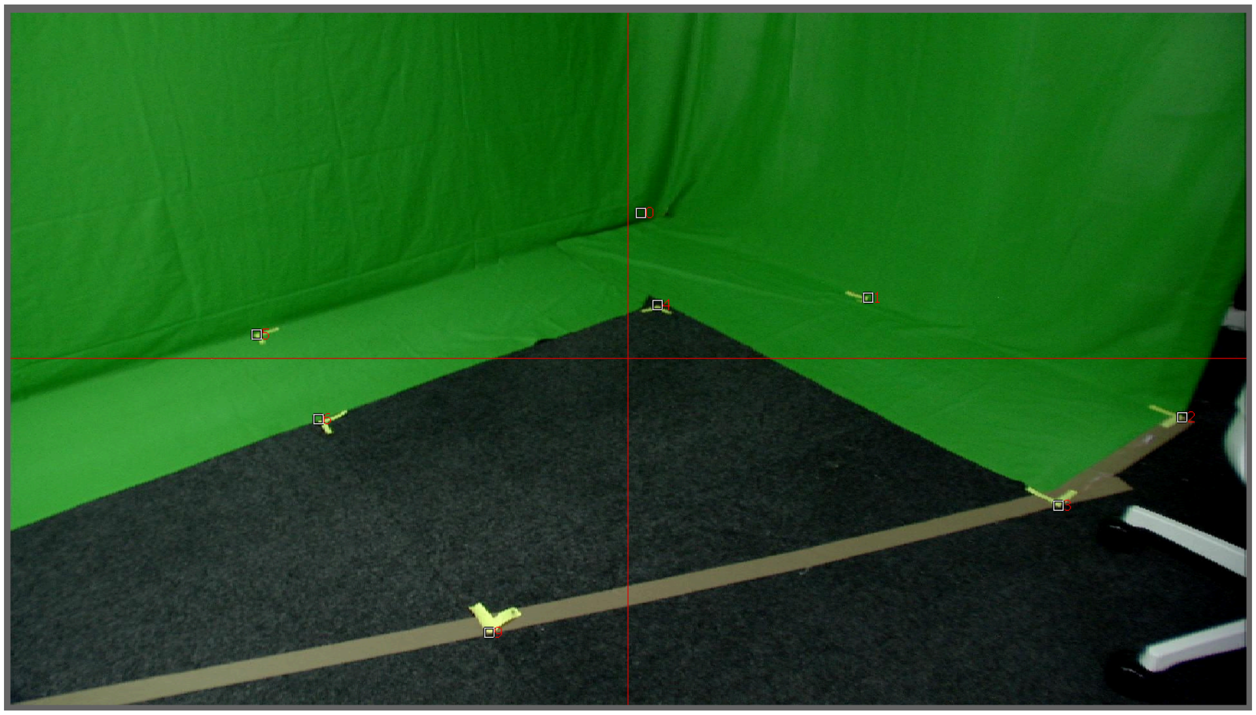
In this example, 5 images have been grabbed, each of them having from 65 to 10 trackers (see Tracker for information about trackers creation). Currently we don't care about Errors. Note that all the frames have been checked, and thus will be used in subsequent calculations.

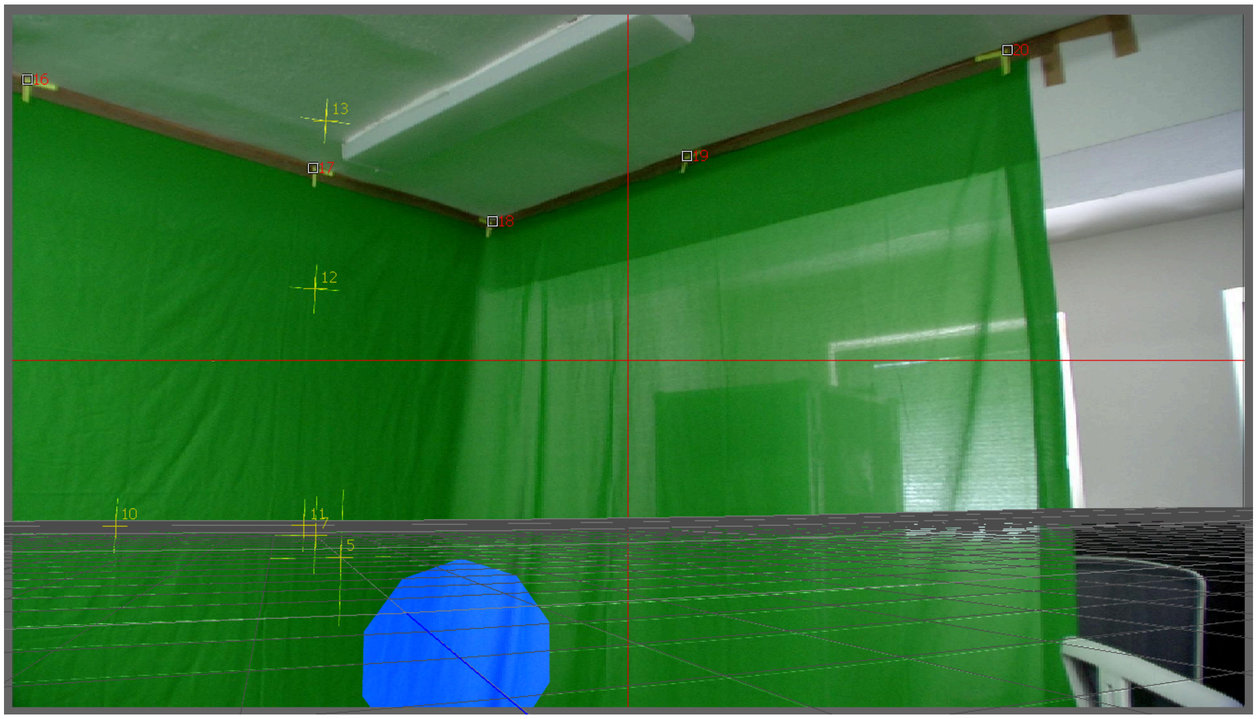


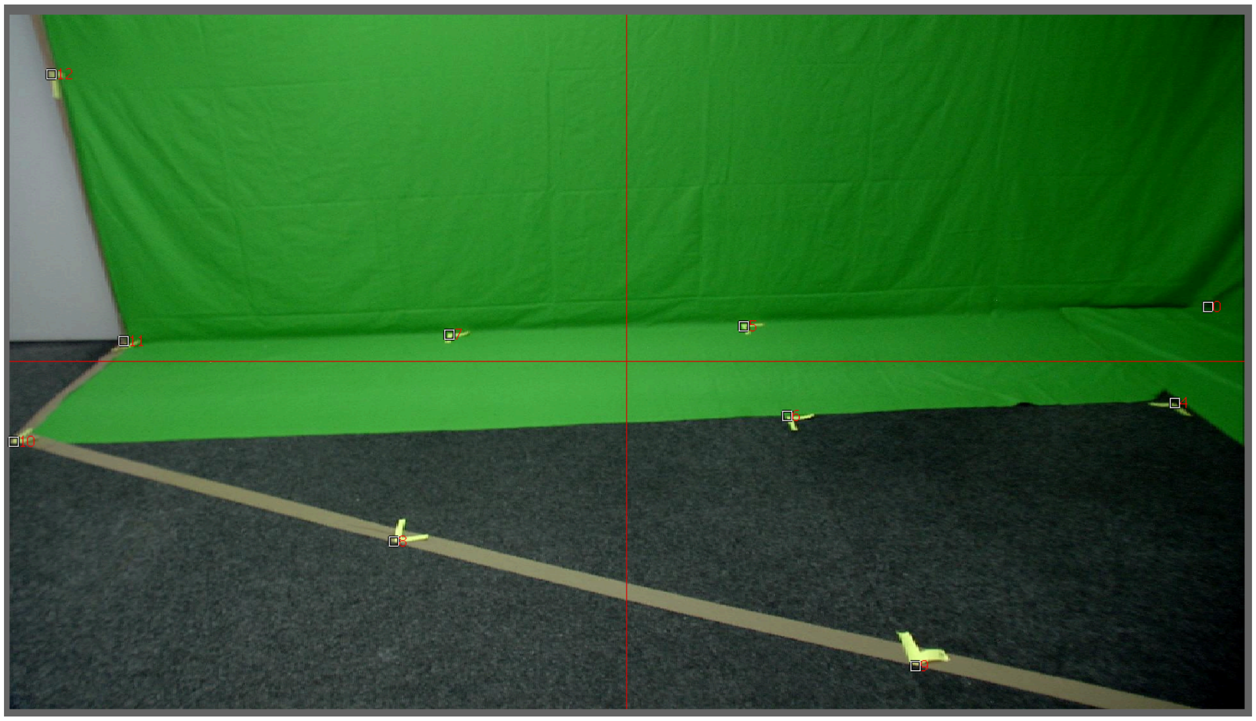
☑ VideoPlayer

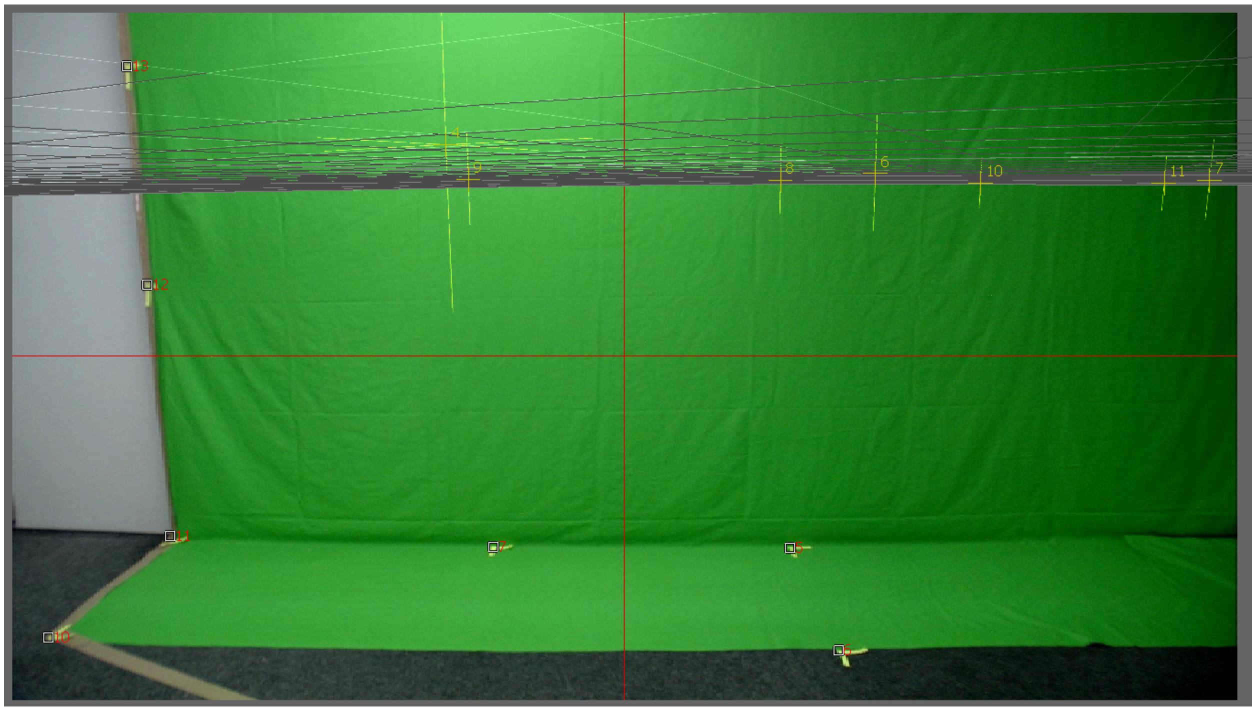
Video Source: None

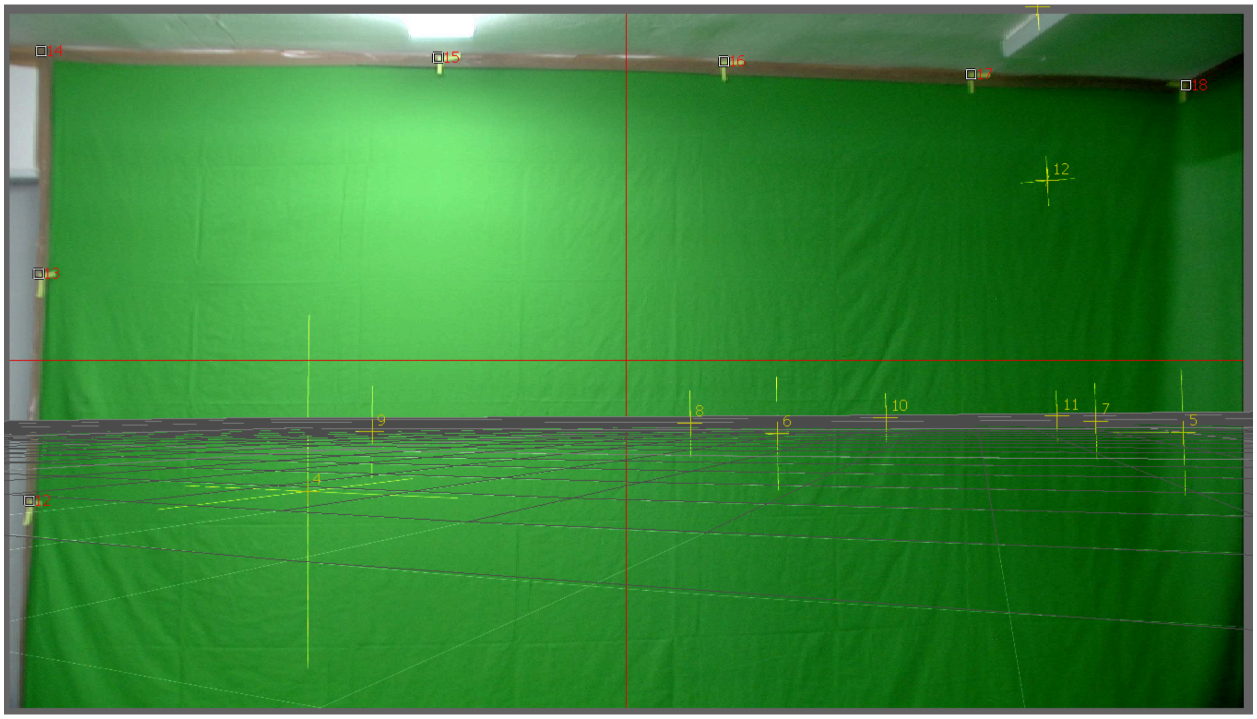
Name	Mean	Tracker	Max	Info
☑ 2021_10_07-...	inf	8/ 8	[0,inf] [1,inf] [2,i...	NA
☑ 2021_10_07-...	inf	5/ 5	[19,inf] [20,inf] ...	NA
☑ 2021_10_07-...	inf	10/10	[0,inf] [12,2643....	NA
☑ 2021_10_07-...	1957.33	7/ 7	[13,2327.8] ...	NA
☑ 2021_10_07-...	inf	7/ 7	[18,inf] ...	NA




16
13
17
18
19
20
12
10
11
7
5



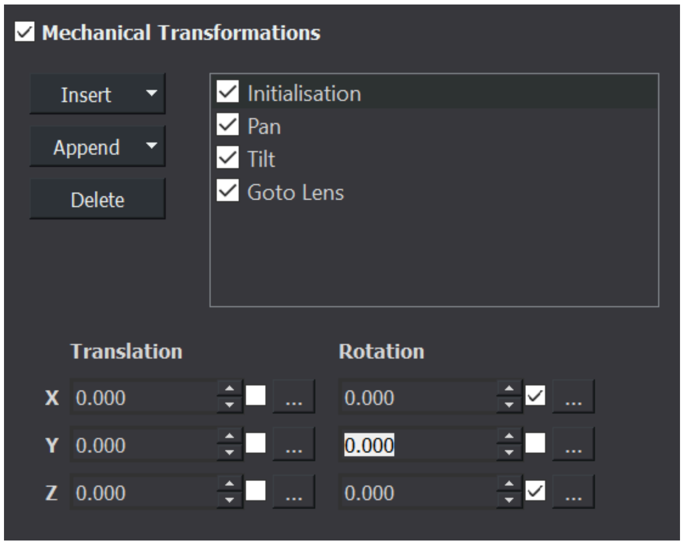

9
8
6
10
11
7
12


14
15
16
18
12
9
8
6
10
11
7
5
4

First step: find rough Initialisation

In our example, we have two stages “Initialisation” and “Goto Lens” where values can be computed.

Since we don’t know much about initialisation position, we can set all the values to 0, and uncheck all the parameters. You can also lock RX and RZ to 0 because you know that the tripod is leveled. Those RX and RZ will be unlocked later.

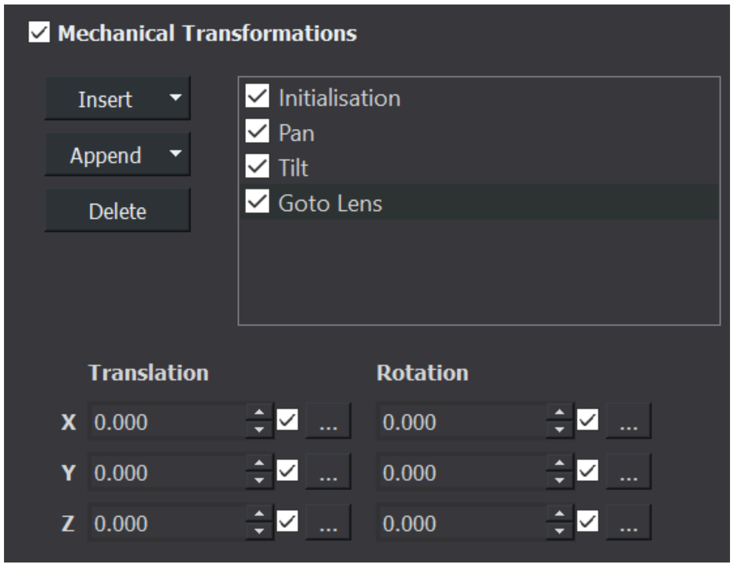


✓ Mechanical Transformations

- ✓ Initialisation
- ✓ Pan
- ✓ Tilt
- ✓ Goto Lens

	Translation		Rotation	
X	0.000	☐	0.000	✓
Y	0.000	☐	0.000	☐
Z	0.000	☐	0.000	✓

Regarding “Goto Lens” it’s important here to lock the values, and fill them with estimated values if you can. Here we set everything to 0 and locked.

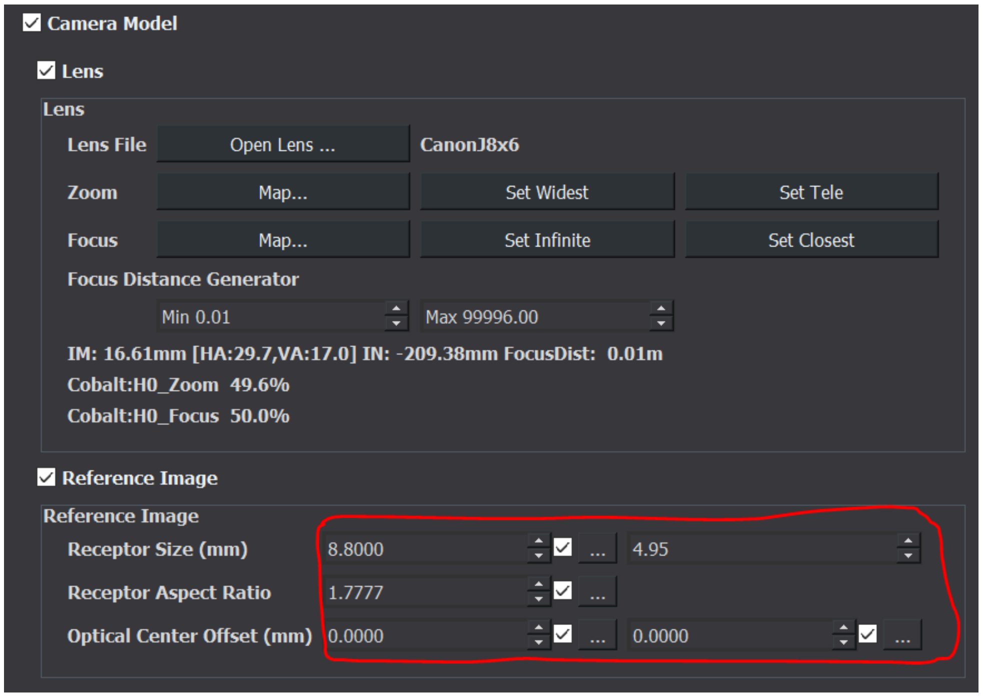


✓ Mechanical Transformations

- ✓ Initialisation
- ✓ Pan
- ✓ Tilt
- ✓ Goto Lens

	Translation		Rotation	
X	0.000	✓	0.000	✓
Y	0.000	✓	0.000	✓
Z	0.000	✓	0.000	✓

Check that all the unknowns in Camera Model are checked and set to reasonable values:

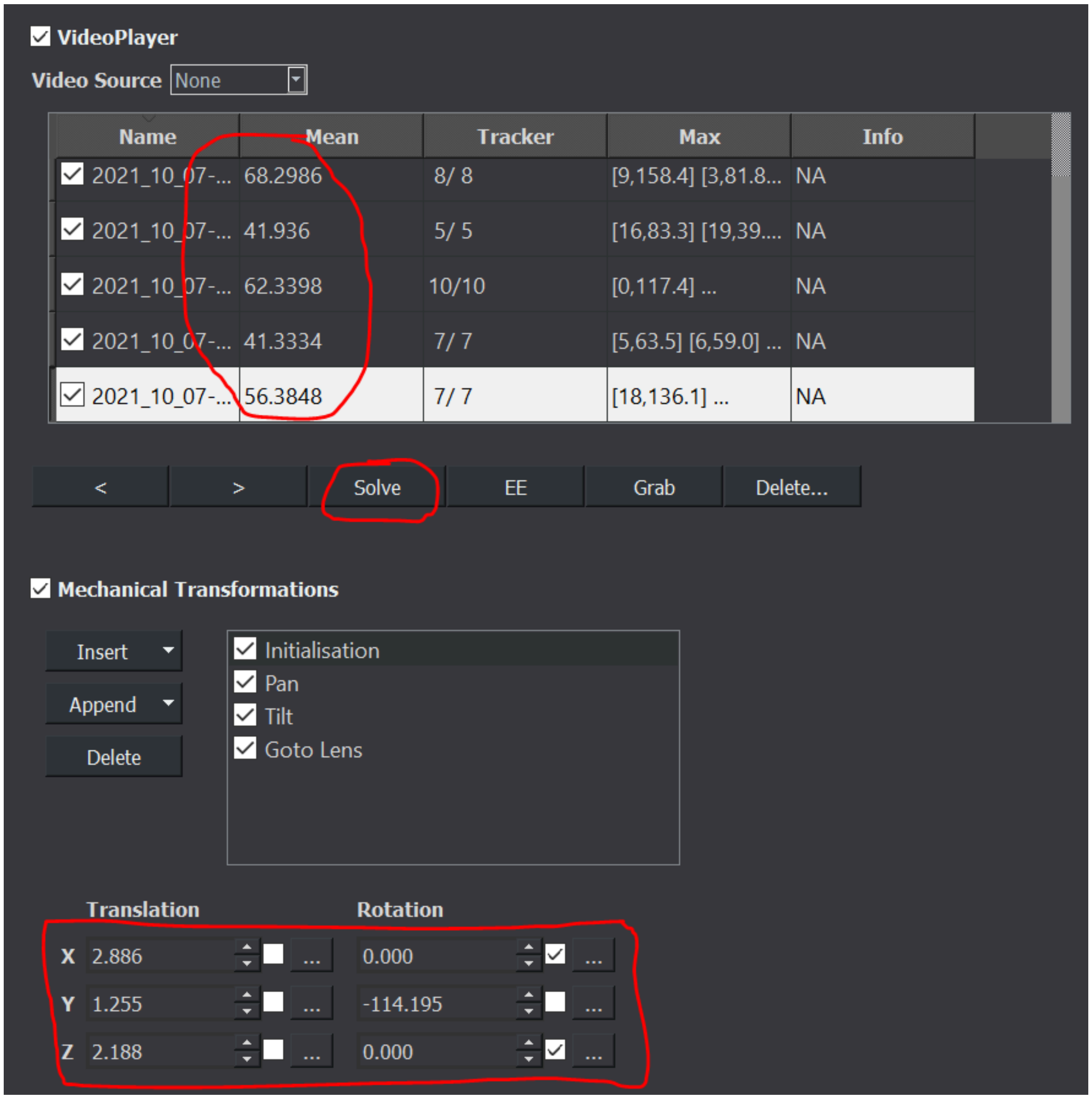


IM: 16.61mm [HA:29.7,VA:17.0] IN: -209.38mm FocusDist: 0.01m
Cobalt:H0_Zoom 49.6%
Cobalt:H0_Focus 50.0%

Receptor Size (mm)	8.8000 ✓	4.95
Receptor Aspect Ratio	1.7777 ✓	
Optical Center Offset (mm)	0.0000 ✓	0.0000 ✓

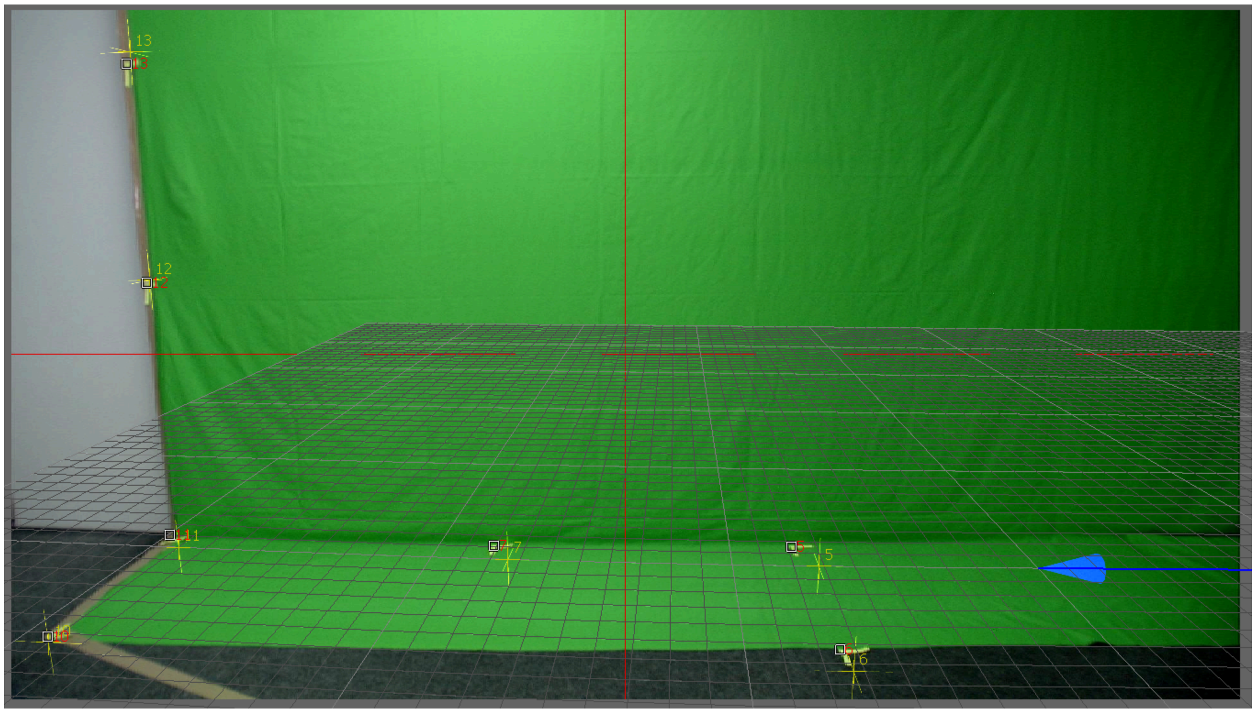
Click Solve:

- Errors drop down to 40/60 pixels (which is still big).
- The values of “Initialisation” except “RX” and “RZ” have been computed.
- The 2D trackers are getting closer to their 3D counterparts.
- Check that on your SDI output when you move the real camera, targets stay moreless attached to their real counterparts.



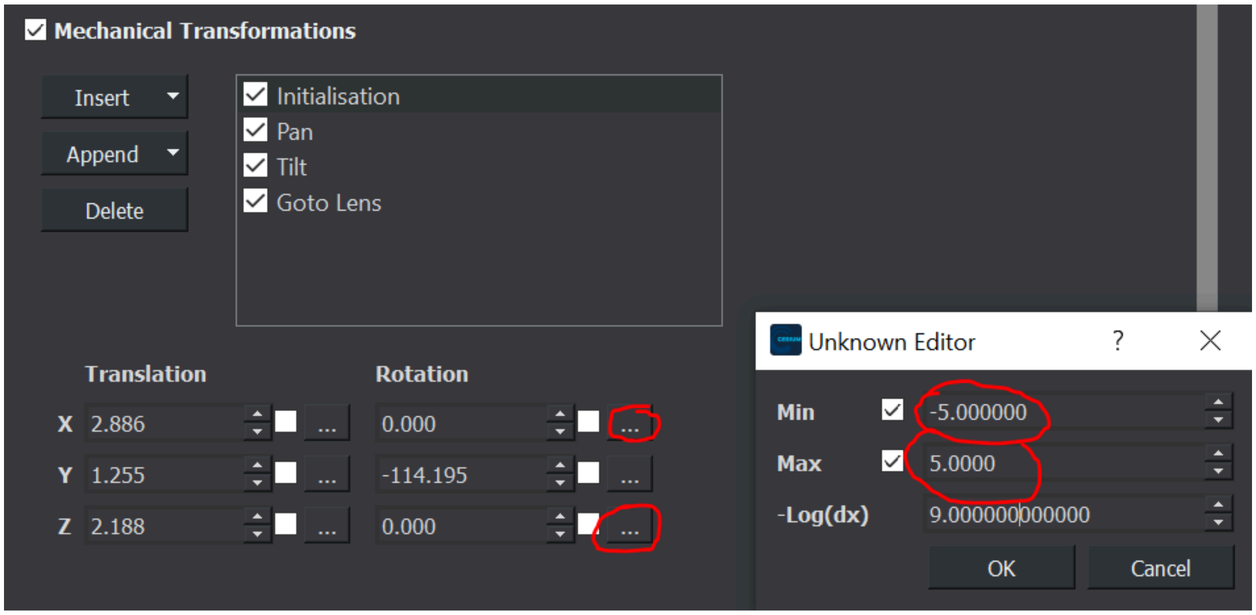
Name	Mean	Tracker	Max	Info
☑ 2021_10_07-...	68.2986	8/ 8	[9,158.4] [3,81.8...	NA
☑ 2021_10_07-...	41.936	5/ 5	[16,83.3] [19,39....	NA
☑ 2021_10_07-...	62.3398	10/10	[0,117.4] ...	NA
☑ 2021_10_07-...	41.3334	7/ 7	[5,63.5] [6,59.0] ...	NA
☑ 2021_10_07-...	56.3848	7/ 7	[18,136.1] ...	NA

	Translation		Rotation	
X	2.886		0.000	✓
Y	1.255		-114.195	
Z	2.188		0.000	✓



Initialisation Improvement

You can now uncheck the “RX” and “RZ” in Initialisation. Since we know those angles are quite small, you can help the optimization process by setting limits on these 2 values (this is not mandatory but can sometimes help a lot). Here we force those 2 angles to stay in the -5, + 5 degrees interval.



☑ Mechanical Transformations

- ☑ Initialisation
- ☑ Pan
- ☑ Tilt
- ☑ Goto Lens

	Translation	Rotation
X	2.886	0.000
Y	1.255	-114.195
Z	2.188	0.000

Unknown Editor

Min	☑	-5.000000
Max	☑	5.0000
-Log(dx)		9.000000000000

Click Solve:

- Error decreases
- RX and RZ have changed.

Note that:

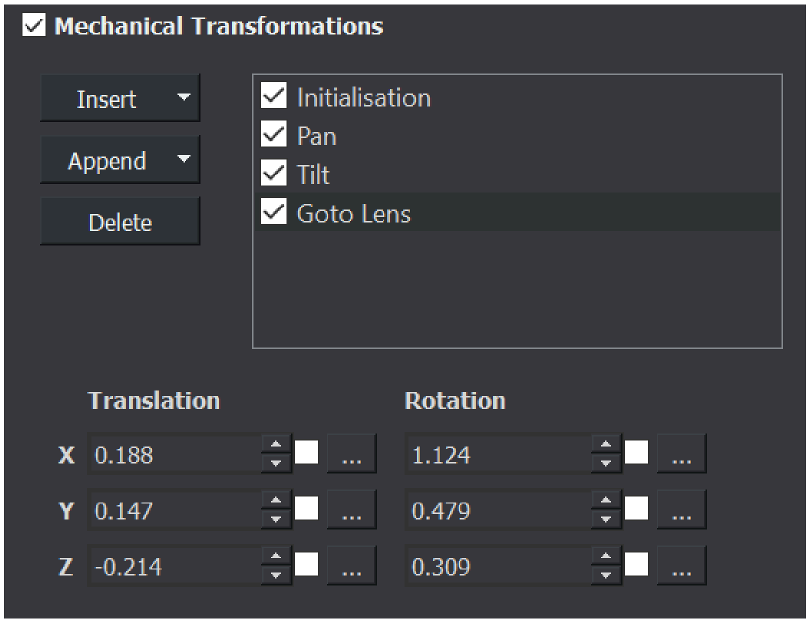
- you can click "Solve" as many times as you want, each time values will be optimized if possible.

Fine tuning

Fine tuning consists of adding new parameters to the optimization process. Not every parameter can be added without further thinking of their meaning. If you do so, the optimization process will become very unstable.

Goto Lens

You can uncheck some "Goto Lens" parameters. Here we just relax all the parameters, and click "Solve" several times. Error is now about 15/30, which becomes much better.



☑ Mechanical Transformations

☑ Initialisation
☑ Pan
☑ Tilt
☑ Goto Lens

	Translation	Rotation
X	0.188	1.124
Y	0.147	0.479
Z	-0.214	0.309

Adding Tilt0 stage

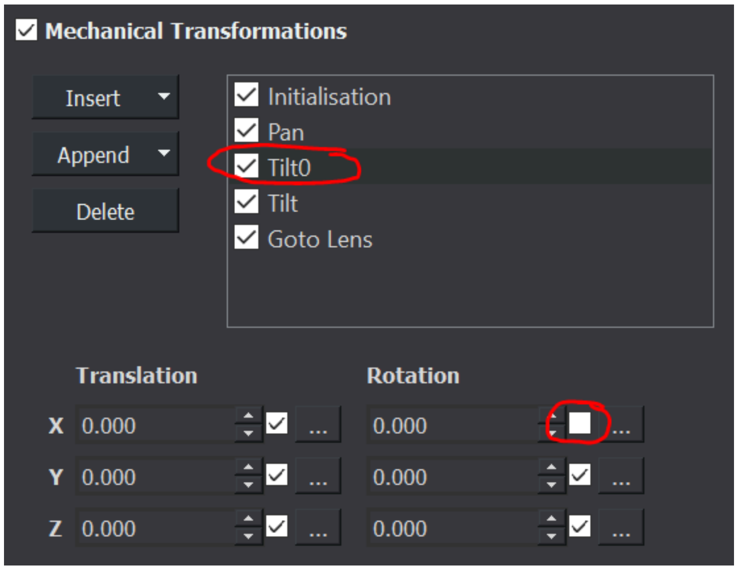
For the silver head, the Tilt reference (zero position), is done when the camera is perpendicular to the pan axis. This is done whether by hand or using a level. In both cases it is not very precise, we can ask the optimisation process to find the value for us.

Important note: if you want the optimization process to reliably compute parameters, you need to provide it with enough trackers and grabbed images. It's probably worthless to compute Tilt0 with a single image.

To add the Tilt0 stage, select the Pan stage, and "Append" a "Fixed Transformation". Rename this stage "Tilt0". Since Tilt is a rotation around the X axis, we uncheck the "RX" axis of Tilt0.

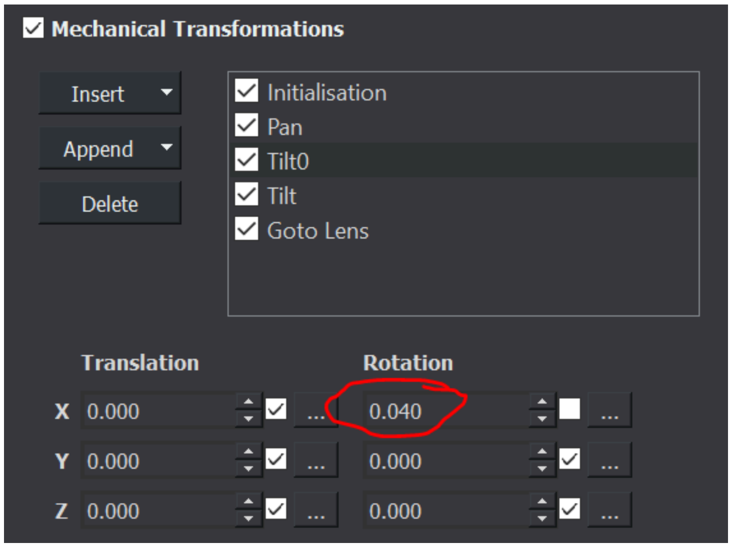
Click Solve, a new value of Tilt0 is computed, and the error drops down a bit.

Note that whenever you click "Solve" all the unknown parameters are computed. When you click "Solve" for Tilt0, not only Tilt0 will be computed but also "Goto Lens" and "Initialisation".



✓ Initialisation
✓ Pan
✓ Tilt0
✓ Tilt
✓ Goto Lens

	Translation		Rotation	
X	0.000	✓	0.000	☐
Y	0.000	✓	0.000	✓
Z	0.000	✓	0.000	✓


☑ Mechanical Transformations

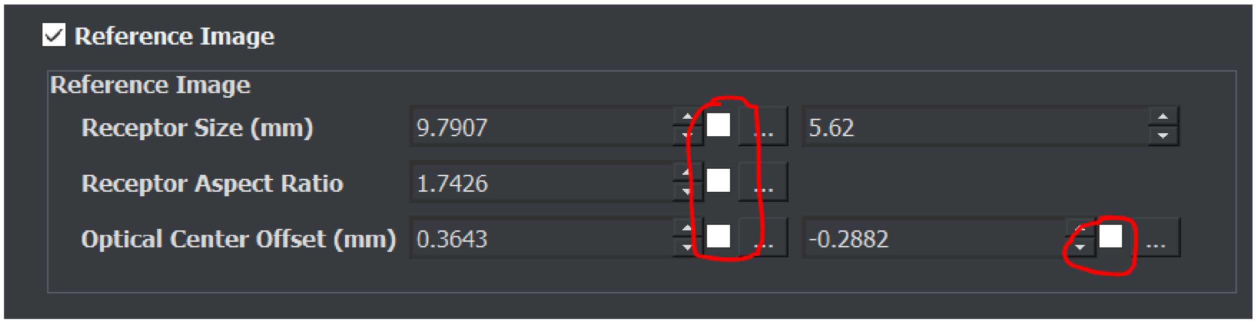
Insert, Append, Delete

☑ Initialisation
☑ Pan
☑ Tilt0
☑ Tilt
☑ Goto Lens

	Translation		Rotation	
X	0.000	✓	0.040	☐
Y	0.000	✓	0.000	✓
Z	0.000	✓	0.000	✓

Adding Lens Parameters

You can then add some lens parameters. It's not necessary to add all of them all the time. Here is the result after having clicked "Solve" a couple of times, the error is now about 10/15 pixels which is not so bad.


☑ Reference Image

Reference Image

Receptor Size (mm)	9.7907	☐	5.62	
Receptor Aspect Ratio	1.7426	☐		
Optical Center Offset (mm)	0.3643	☐	-0.2882	☐

Vendors specific notes

Sony BRC camera

SONY BRC X-400, X-1000 is supported via the Vinten D1 IP driver.

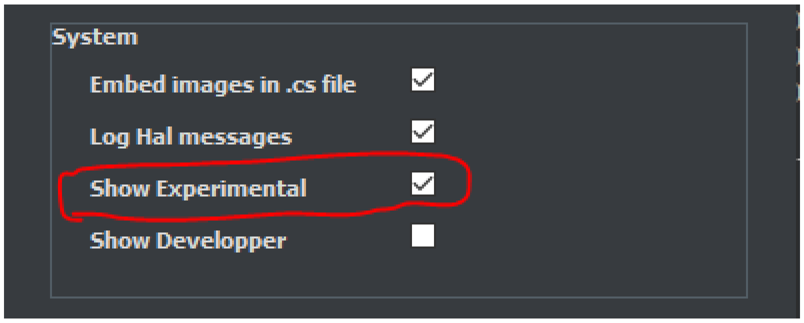
You need to configure the BRC so that it exports its tracking data to your IP address.

The default port is 40000.

If you have several BRC cameras connected to a single Cesium, then each BRC should export on its own port.

Experimental

Under Tools → Preferences... → System → Show Experimental, enable experimental



System

Embed images in .cs file ✓

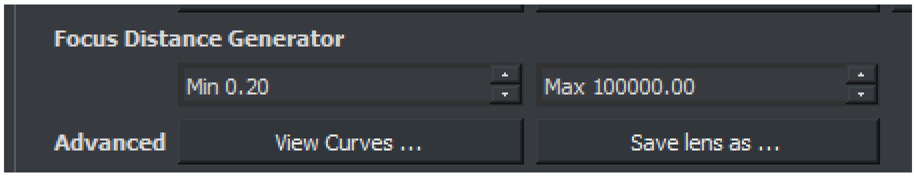
Log Hal messages ✓

Show Experimental ✓

Show Developper ☐

View Curves

allows to visualize the calibration curves of the lens.

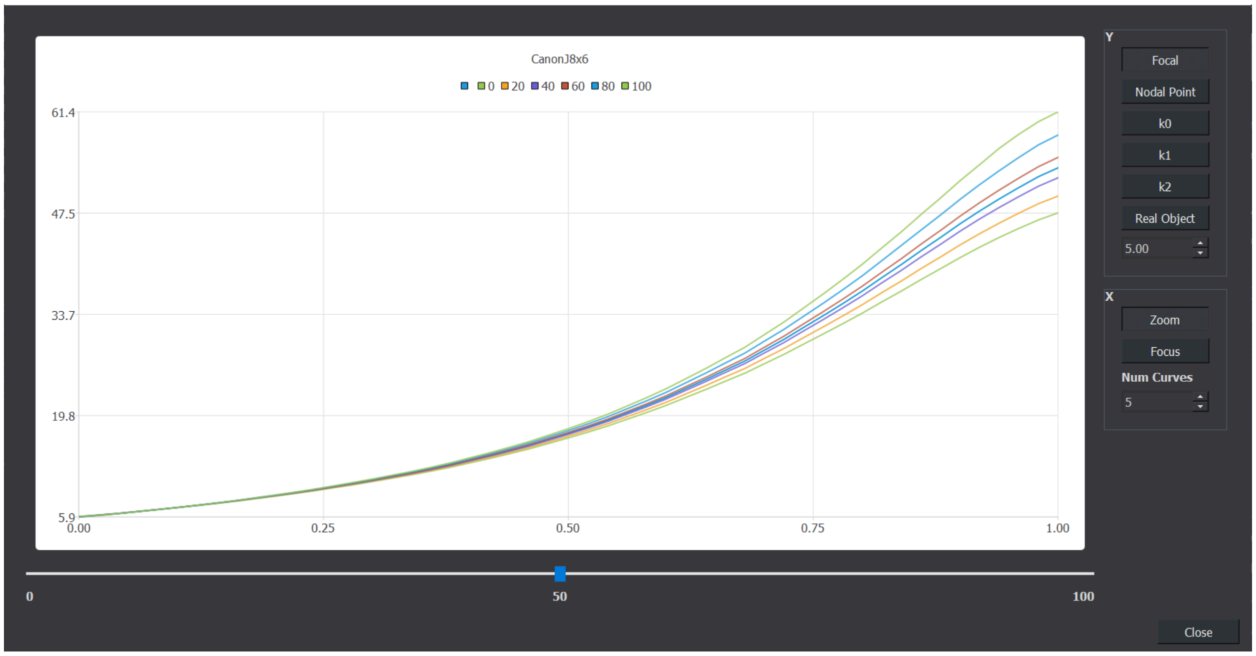


Focus Distance Generator

Min 0.20

Max 100000.00

It can be found in the Advanced section of Lens.



This tool allow to vertically visualize one of:

- Focal: Focal length in mm
- Nodal Point Shift: displacement of the nodal point along the optical axis in mm
- K0,K1,K2: the distortions parameters (mm-2,mm-4,mm-6,)
- Real Object: display the size on the ccd of a real object located at 10m from the camera. The actual size of the real object is set with the lower spinbox (here 5.0m). Caution: values above 5. mm are usually (not always) meaningless. In fact the maximum acceptable value can be taken as the half diagonal of the receptor size.

The horizontal axis can be on of:

- Zoom: normalized zoom encoder in [0,1]
- Focus: normalized focus encoder in [0,1]

Open Lens

when enabled it adds formats to the existing open lens format of .lns in form of .xml and .csv

this is found in the Rig window (if created) under the Camera Model → Lens section then “Open Lens …”

.xml format

xml format support ph format witch is Paraboloid Hyperbolic and polynomial witch is

K0,K1,K2 polynomial

file start with

```
<!DOCTYPE hmc_lens>

<lens>

        …

</lens>
```

First required section is asset which specifies basic data

```
<asset>

  <date>Thu Feb 1 14:38:33 2024</date>

  <name>Canon 14x4.3</name>

  <description></description>

  <model>ph</model>

 </asset>
```

Second section can be ph or polynomial

```
<ph>
…
</ph>
or
<polynomial>
…
</polynomial>
```

For ph

<zoom size="10" min="0" max="100">0 11.1111 22.2222 33.3333 44.4444 55.5556 66.6667 77.7778 88.8889 100 </zoom>

specifies a zoom levels in sensor values

<focus size="6" min="0" max="100">0 19.9999 40 60 80.0001 100 </focus>

specifies a focus levels in sensor values

<rsrc size="25" min="0.22" max="5.5">0.22 0.44 0.66 0.88 1.1 1.32 1.54 1.76 1.98 2.2 2.42 2.64 2.86 3.08 3.3 3.52 3.74 3.96 4.18 4.4 4.62 4.84 5.06 5.28 5.5 </rsrc>

specifies circle undistorted radii coming from screen center out, the size is related to the receptor size with calculation "length(ReceptorSize * (UV + offset))"

```
 <im>
     <im_ size="6" zi="0" min="19.654399871826172" max="19.602199554443359">19.6544 19.6382
19.608 19.5976 19.5827 19.6022 </im_>
     <im_ size="6" zi="1" min="23.55109977722168" max="21.727800369262695">23.5511 23.5803
23.5869 23.6242 23.6493 21.7278 </im_>
    <im_> …  </im_>
    …
</im>
```

paraxial focal distance in array based on the focus, meaning size = is same as the <focus> tag elements and the number of rows is based on <zoom> tag elements indexed with zi =

```
 <in>
   <in_ size="6" zi="0" min="-288.39801025390625" max="-288.39801025390625">-288.398 -288.398
-288.398 -288.398 -288.398 -288.398 </in_>
   <in_ size="6" zi="1" min="-288.39801025390625" max="-288.39801025390625">-288.398 -288.398
-288.398 -288.398 -288.398 -288.398 </in_>
   <in_>... </in_>
   …
 </in>
```

nodal point shift in array based on the focus, meaning size = is same as the <focus> tag elements and the number of rows is based on <zoom> tag elements indexed with zi =

```
 <rdist>
```

```
  <rd_ size="25" zi="0" fi="0" min="0.219" max="5.048">0.219969 0.439751 0.659159 0.878009
1.09612 1.3133 1.52938 1.74417 1.95751 2.16922 2.37913 2.58708 2.7929 2.99643 3.19752
3.39601 3.59174 3.78458 3.97439 4.16104 4.34448 4.52468 4.7018 4.87622 5.04882 </rd_>
  <rd_ size="25" zi="0" fi="1" min="0.219" max="5.075">0.21997 0.43976 0.659192 0.878085 1.09626
1.31356 1.52979 1.74479 1.95839 2.17044 2.38077 2.58923 2.79567 2.99994 3.20189 3.4014
3.59833 3.79256 3.98397 4.17251 4.35812 4.54087 4.72095 4.89886 5.07557 </rd_>
    <rd_> … </rd_>
     …
 </rdist>
```

distorted radii in 2D array. The number of <rd_> elements is based on the zoom elements times the focus elements, zoom is indexed with zi= and focus is indexed with fi=. The number of elements in <rd_> is based on the <rsrc> tag. where the resulting "UVd = (Pc * (rd / r) / ReceptorSize) - offset" where "Pc = ReceptorSize * (UV + offset)" and "r = length(Pc)", while rd is larped value depending on the ViewportUV (UV) (value 0 to 1 from the center of the screen)

For polynomial

currently not supported

Last section is
<focus_info dmax="100000" dmin="0.2" model="Hyperbolic"/>
defines maximum and minimum focus distance

example file:

```
<!DOCTYPE hmc_lens>
<lens>
  <asset>
    <date>Mon Feb 5 15:30:14 2024</date>
    <name>OneOneLens</name>
    <description></description>
    <model>ph</model>
  </asset>
```

```
 <ph>
  <zoom size="2" min="0" max="100">0 100 </zoom>
  <focus size="2" min="0" max="100">0 100 </focus>
  <rsrc size="2" min="0" max="5.5">0 5.5 </rsrc>
  <im>
   <im_ size="2" min="0" zi="0" max="0">0 0 </im_>
   <im_ size="2" min="100" zi="1" max="100">100 100 </im_>
  </im>
  <in>
   <in_ size="2" min="0" zi="0" max="0">0 0 </in_>
   <in_ size="2" min="0" zi="1" max="0">0 0 </in_>
  </in>
  <rdist>
   <rd_ size="2" fi="0" min="0" zi="0" max="5.5">0 5.5 </rd_>
   <rd_ size="2" fi="1" min="0" zi="0" max="5.5">0 5.5 </rd_>
   <rd_ size="2" fi="0" min="0" zi="1" max="5.5">0 5.5 </rd_>
   <rd_ size="2" fi="1" min="0" zi="1" max="5.5">0 5.5 </rd_>
  </rdist>
 </ph>
 <focus_info dmin="0.2" dmax="100000" model="Hyperbolic"/>
</lens>
```

.csv format

This lens file format supports only a very specific format structure that needs to be kept or it will result in the lens not loading correctly.

CJ24ex7.5 B [Name]	65535 [Zoom sensor range]	65535 [Focus sensor range]	Image size [Static text]	23.6 [Image size H]	13.3 [Image size W]

First row

- default Zoom/Focus sensor range is 100 meaning a range from 0-100

- [Zoom sensor range] is optional
- [Focus sensor range] is optional

for [Image size H] to be set both [Zoom sensor range] and [Focus sensor range] needs to be set and [static text] needs to be "Image size"

- [Image size H] is optional
- [Image size W] is optional - Not implemented (setting this does nothing)

Second row

[Empty space]	Zoom(%) [Static text]	0-100 (0,2,4...100) [Zoom range array]

[Static text] needs to be this specific text "Zoom(%)"

note that amount of columns in [Zoom range array] should be same as the rows under

Third row

[Empty space]	Principal Point(mm) [Static text]	55,56,58...85 [Principal Point array]

[Static text] needs to be this specific text "Principal Point(mm)"

Fourth row

Focus(%) [Static text]	obj(mm)/f(mm) [Static text]	7.2,7.4...23 [Focal Length array]

[Static text] needs to be this specific text "obj(mm)/f(mm)"

Fifth row and so on

0-100 (0,2,4...100) ↓	INF...-800 ↓	1,1,..

[Focus range array]	[Distance from object]	0.99,0.93.. ... [AFOV multiplier 2D array]

[Focus range array] values go down in single column form 0 at first row and 100 at last row

[Distance from object] values go down in single column - Not used (setting this does nothing)

[AFOV multiplier 2D array] Values in row should be in the Zoom(%) columns, and in columns should be number of rows in Focus(%)

Save Lens

Allows to save selected lens to file

Troubleshooting

Virtual elements are not moving with respect to the camera in my 3D rendering engine:

Check in Cesium viewer that the grids and/or target are moving properly for the SAME camera.

- If not, check the configuration of your 3D rendering engine. Does it read its Cesium data on the correct IP/port, is it properly configured to display this Cesium camera?
- If yes, go to the next question.

Virtual elements (grids/targets) are not moving with respect to the camera in Cesium viewer:

Possible causes could be:

- Driver does not deliver data: check that the driver to which the rig is getting data from delivers data (numdata should steadily increase). Check that those values make sense by displaying the 'cooked' data. You can quickly check that angles are given in degree and distances/offsets are given in meter.
- Rig is not properly genlocked to the driver: check that the driver/rig to which is genlocked the rig actually delivers data (numdata should steadily increase)
- Rig is not properly calibrate and/or lens is not properly calibrated and the camera is looking to an 'empty space': try to move the camera so that you can see a grid, or a target. Also check the focal angles, if you have a very narrow focal angle, it will be very difficult to aim a target or a piece of a grid (it's as if you were using a telescope).

Virtual elements are moving but are not properly located when the camera moves:

If the elements are properly positioned when the camera is still, you have to tune the delays between the real and the virtual camera. This entirely depends on your video configuration. You could act on:

- Hardware video delay
- Delays in Cesium
- Delays in your 3D software

Virtual elements are moving but are not properly located when the camera does not move:

Possible causes could be:

- Driver delivers wrong data: check in the corresponding driver that the delivered data seem correct. Also check in the rig that the Pan/Tilt/… values seem ok (distances are meters, angles are degree).
- Wrong calibration: check your mechanical and optical calibration. It may happen that the mechanical calibration is OK while the optical one is wrong; in this case, you should have a moreless proper calibration at the center of the image (close to the optical center). Note also that with some hardware systems, you may lose calibration when the camera rig is powered off.

Nothing happens when you click on Solve in Rig

- Make sure that in Tools->Preferences->Solver ->Max iterations, and Num Passes are positive integer numbers.

Known issues

Some SDI input formats are not supported marked with

"-> Cesium SDI In : No Access"

Crashes (on VSAR disconnect, changing camera model, deleting rig)

ABOUT US

Chyron is ushering in the next generation of storytelling in the digital age. Founded in 1966, the company pioneered broadcast titling and graphics systems. With a strong foundation built on over 50 years of innovation and efficiency, the name Chyron is synonymous with broadcast graphics. Chyron continues that legacy as a global leader focused on customer-centric broadcast solutions. Today, the company offers production professionals the industry's most comprehensive software portfolio for designing, sharing, and playing live graphics to air with ease. Chyron products are increasingly deployed to empower OTA & OTT workflows and deliver richer, more immersive experiences for audiences and sports fans in the arena, at home, or on the go.

CONTACT SALES

EMEA • North America • Latin America • Asia/Pacific
+1.631.845.2000 • sales@chyron.com