Blender Virtual Prod Addon Documentation

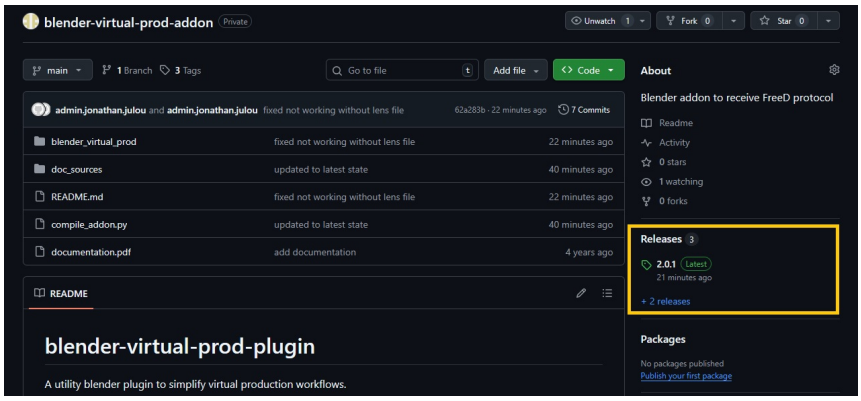
11/02/2026

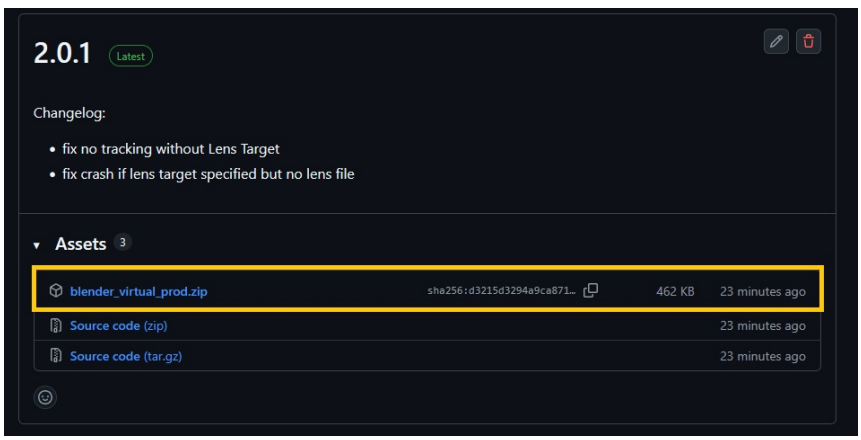
Contents

I. Installation

I.1. Github page

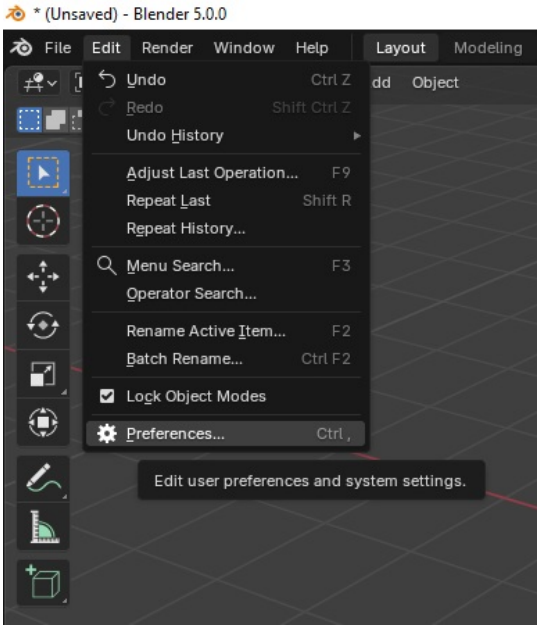
Go to https://github.com/jonathanjulou/blender-virtual-prod-addon. Go to the **Releases** page and download the latest stable release:



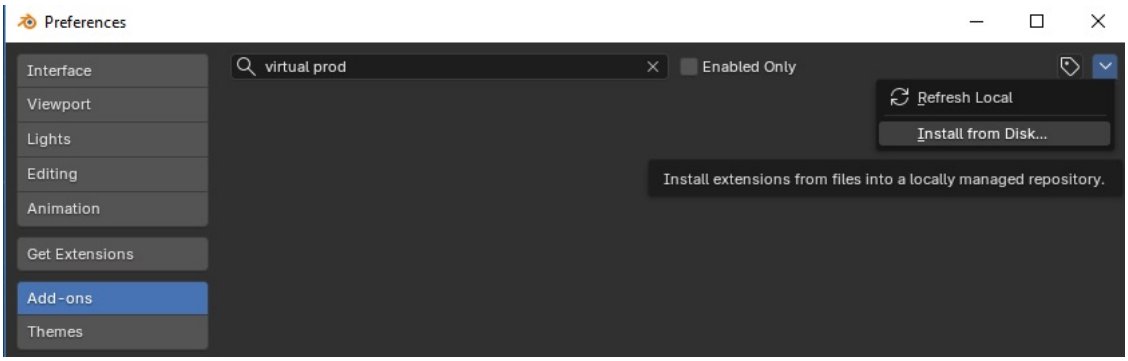


I.2. Install package in Blender

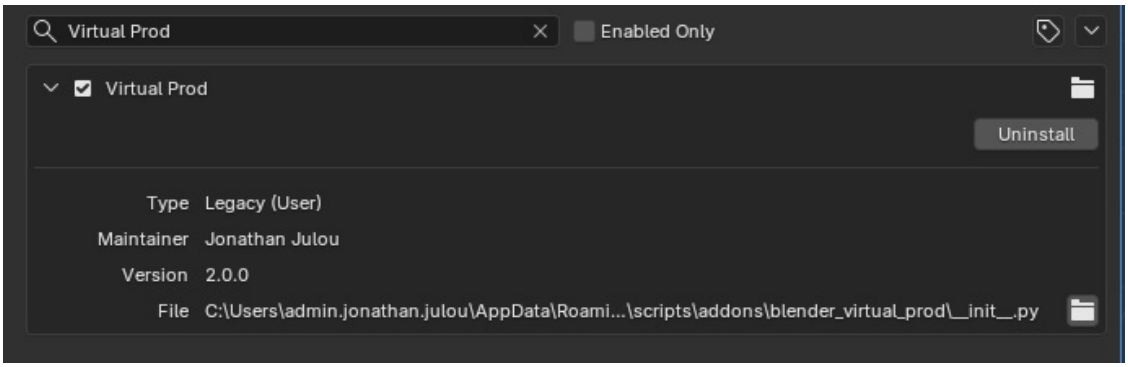
Open Blender and navigate to **Edit** → **Preferences**:



Open the **Add-Ons** tab, and click the little down arrow at the top-right. Select **Install from Disk...**



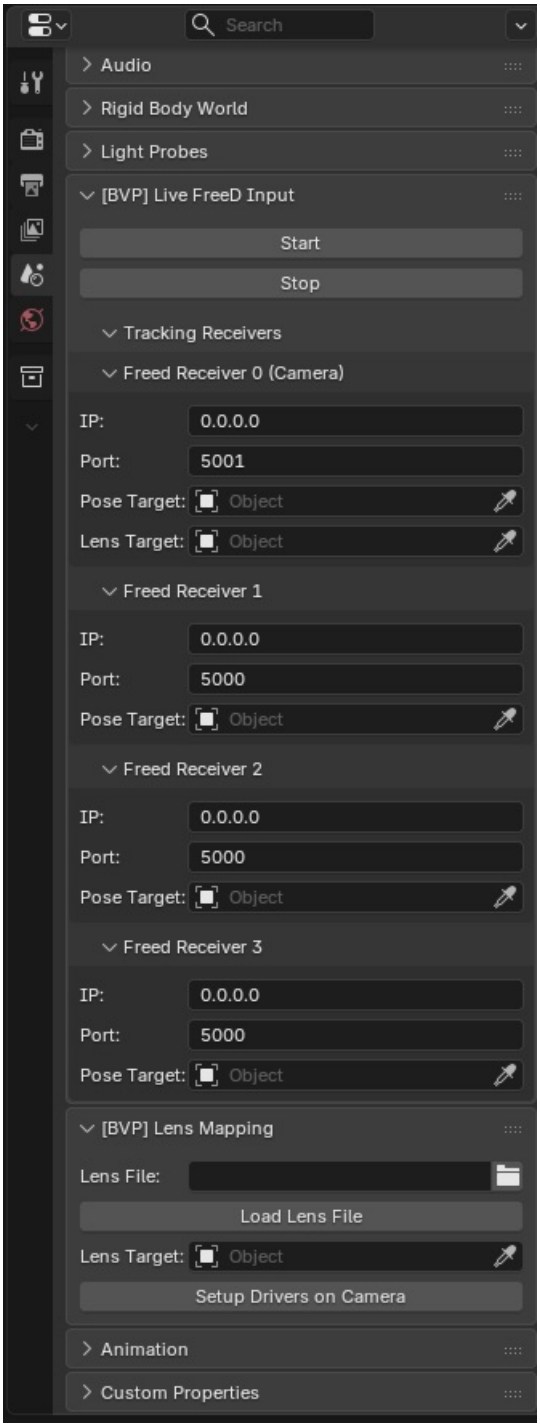
Select the .zip file downloaded from the Github release. This should appear:



The plugin is now correctly installed in Blender.

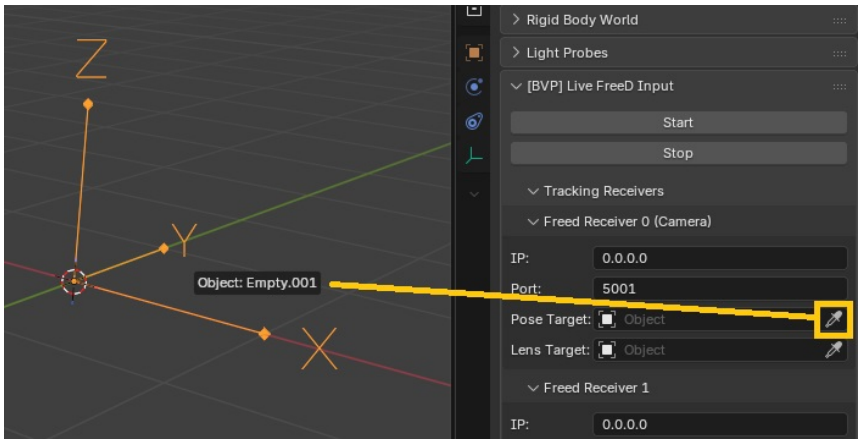
II. Live FreeD input

The Live FreeD Input part of the addon is found in the **Scene** tab:

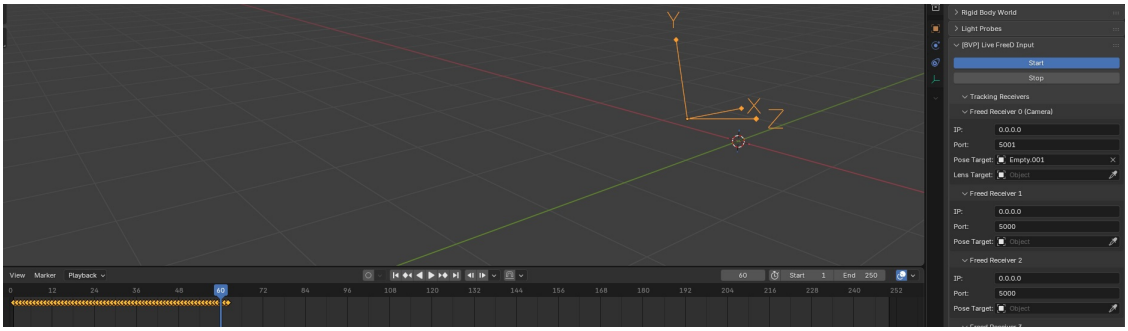


II.1. Basic pose tracking

You can apply the tracking data to any Blender object. Here we use an Empty. Select it in one of the 4 receivers (the main receiver, dedicated for camera, is Receiver 0):

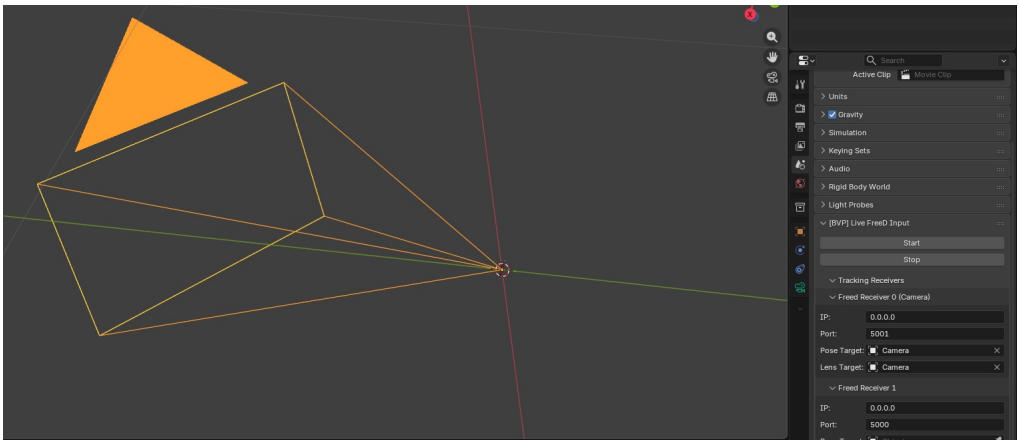


Press **Start**. The object will start moving according to the FreeD data. For each packet received a keyframe is written:

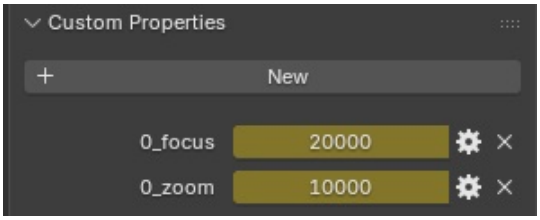


II.2. Basic lens tracking

Now we target a Blender Camera object as target for both pose and lens:

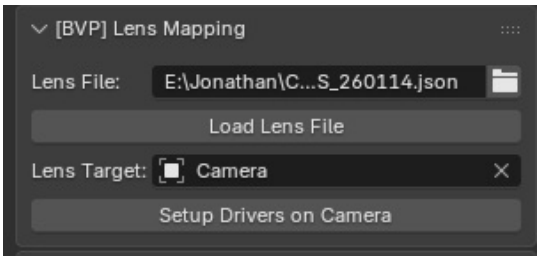


Now FreeD Zoom and Focus data is written as well as Custom Properties in the camera. Note that FreeD does not carry meaningful information like focal length or focus distance, so they still get the default value:



II.3. Using a lens file

It is possible to get useful camera intrinsics using a lens file. For this we will go to the **[BVP] Lens Mapping** tab:

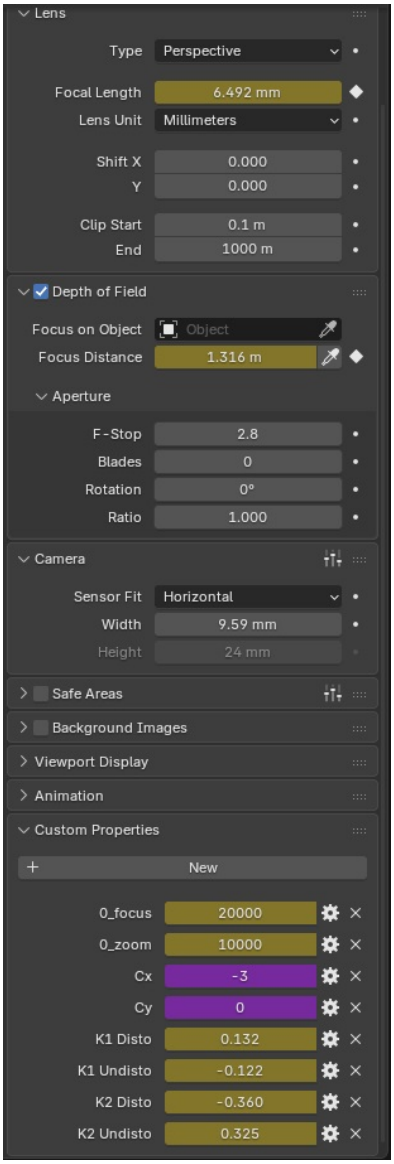


Select a **OpenLensFile** JSON file as lens file (standardized output from Miraxyz calibration software, currently this is the only mapping format supported), and press **Load Lens File**.

Then select the Camera object as **Lens Target** and press **Setup Drivers on Camera** to create the drivers for focal length and focus distance.

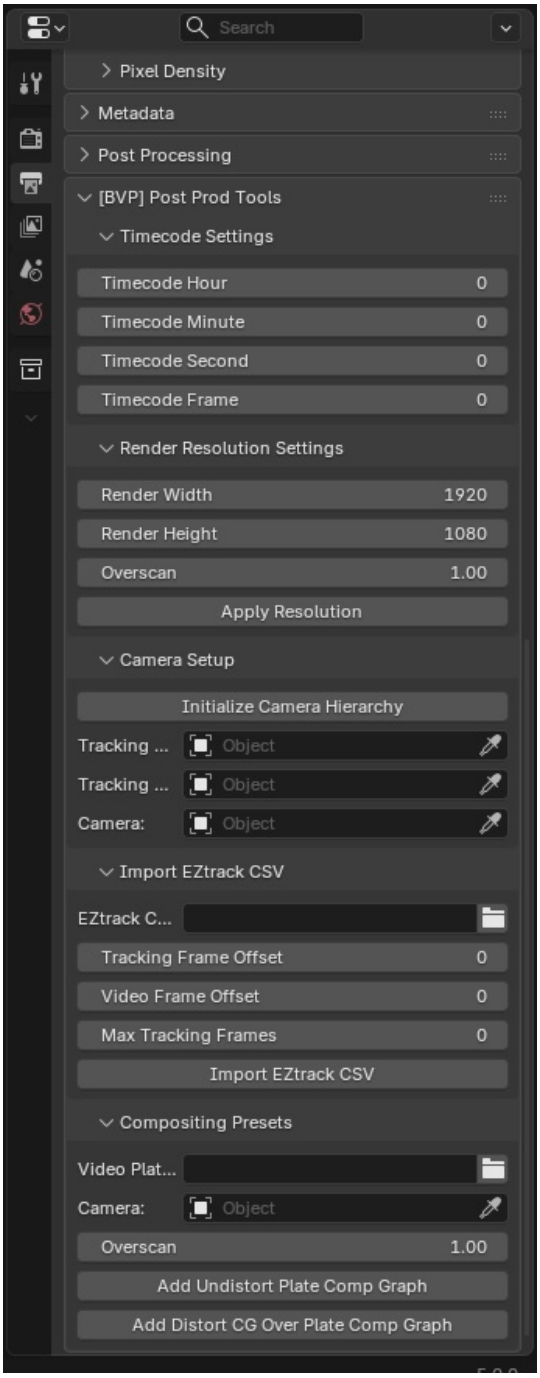
Loading the lens file declares functions that can be used from Blender drivers to convert raw zoom and focus values into calibrated data. This is more robust than doing the conversion directly in Python when not using the auto-keyframing, and allows using the lens file mapping without the live Freed input, by manually moving and keyframing the Zoom and Focus Custom Data.

Now the camera has usable parameters:



III. Post Prod Tools

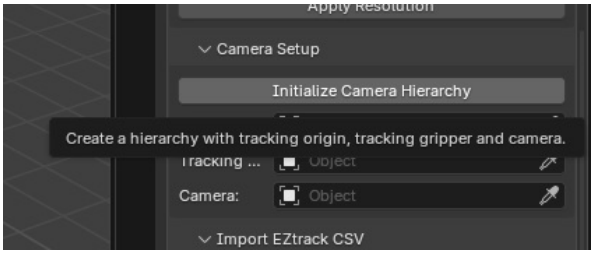
The Post Prod Tools part of the addon is found in the **Output** tab:



These tools are mostly intended to handle CSV recordings of tracking data (based on EZtrack format), but also provides useful tools for handling compositing with distortion and overscan in Blender, as well as setting up the camera hierarchy quickly.

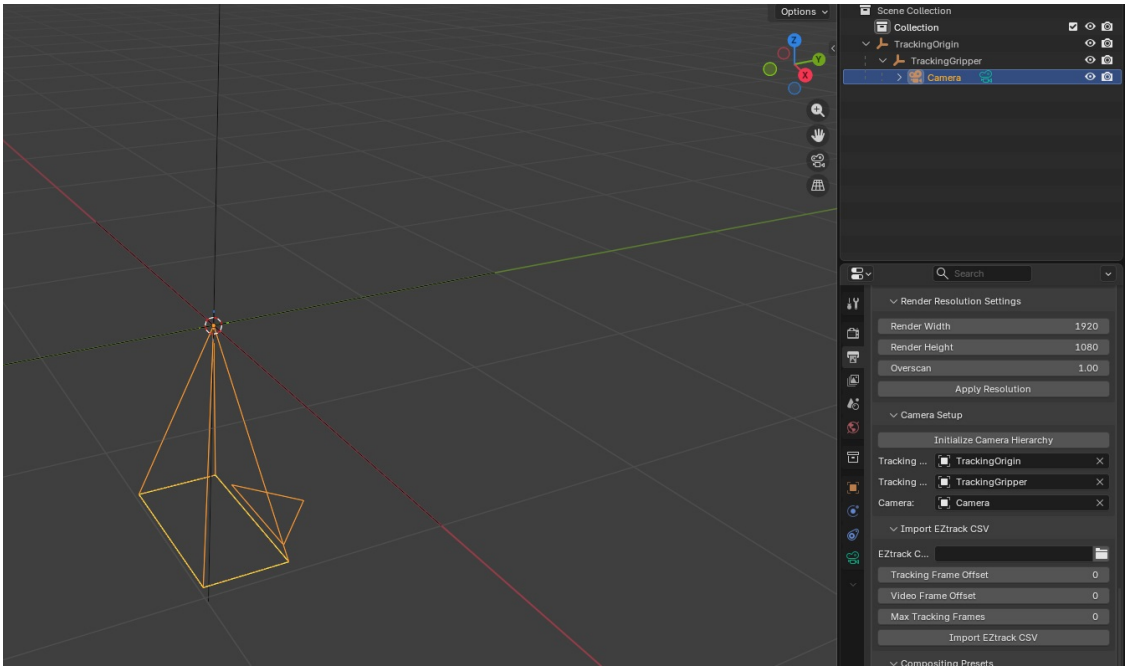
III.1. Camera Setup

Press **Initialize Camera Hierarchy**:



This creates three objects:

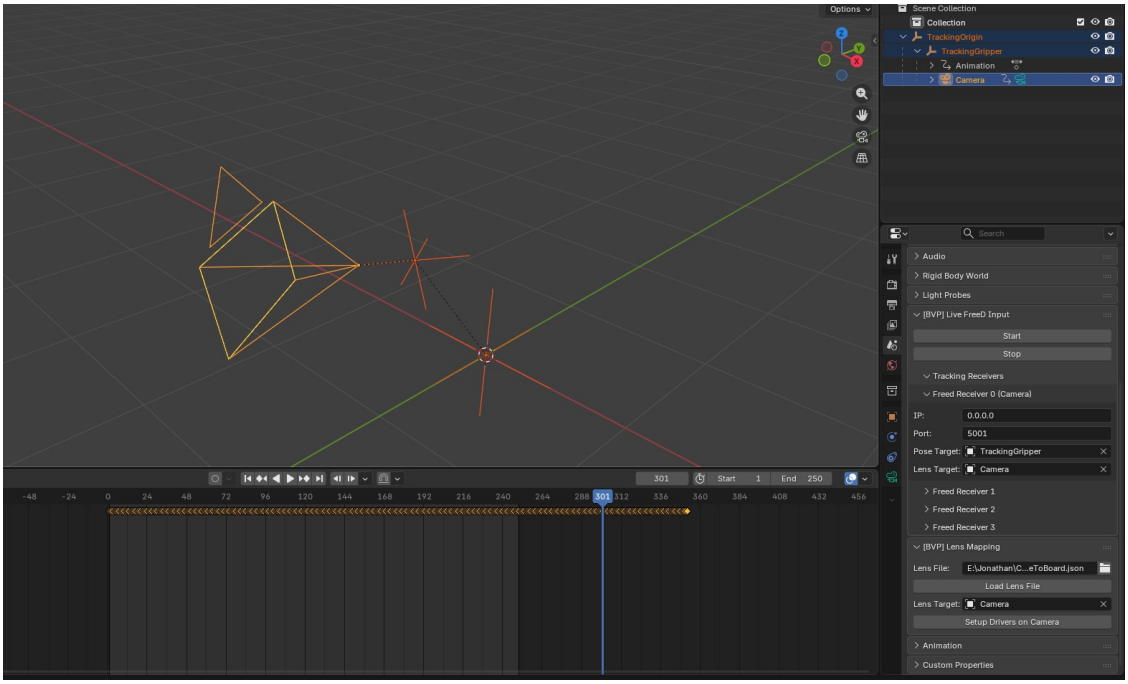
- Tracking Origin: The origin of the tracking system.
- Tracking Gripper: The position and rotation given by the FreeD stream, can be seen as the tracker, or the last degree of freedom of a tracking kinematic chain.
- Camera: The camera. This represents the position of the center of the sensor of the real camera.



This hierarchy is interesting, as it allows to refine both the global position in the 3D scene by moving the origin, and the camera alignment by moving the camera under the gripper object after the fact.

III.1.1 Using camera setup in live freed mode

Set the **Pose Target** to **TrackingGripper** and the **Lens Target** to **Camera**:



This way lens data is applied on the camera and positional data on the gripper.

Note how the camera is placed forward of the gripper object. This is because the lens data applies the **Entrance Pupil Distance** (EPD, or nodal offset) as a driver on the camera's Z axis, bringing it forward according to the current zoom level and lens file: