Blender Virtual Prod Addon Documentation

11/02/2026

Contents

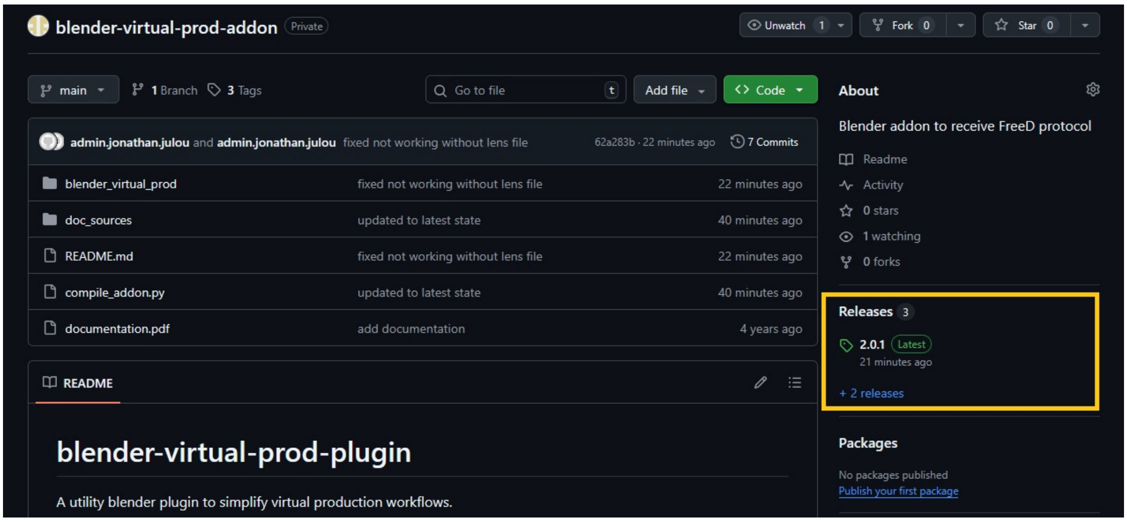
I Installation **1**
I.1 Github page 1
I.2 Install package in Blender 2

II Live FreeD input **3**
II.1 Basic pose tracking 4
II.2 Basic lens tracking 4
II.3 Using a lens file 5

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:



admin.jonathan.julou and admin.jonathan.julou fixed not working without lens file 62a283b · 22 minutes ago 7 Commits

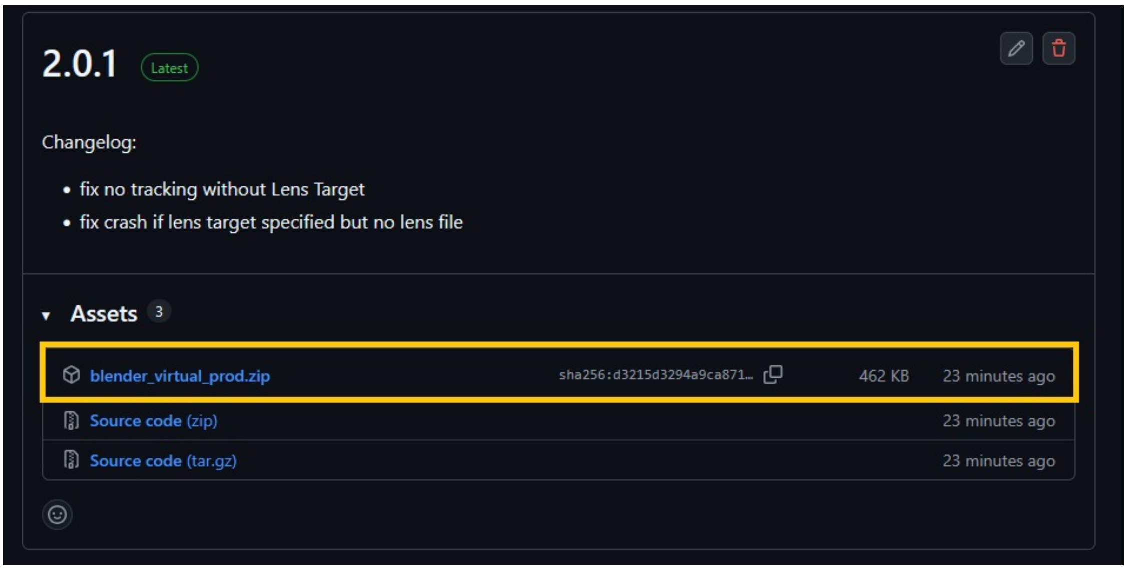
blender_virtual_prod	fixed not working without lens file	22 minutes ago
doc_sources	updated to latest state	40 minutes ago
README.md	fixed not working without lens file	22 minutes ago
compile_addon.py	updated to latest state	40 minutes ago
documentation.pdf	add documentation	4 years ago

blender-virtual-prod-plugin

A utility blender plugin to simplify virtual production workflows.

About

Blender addon to receive FreeD protocol



2.0.1

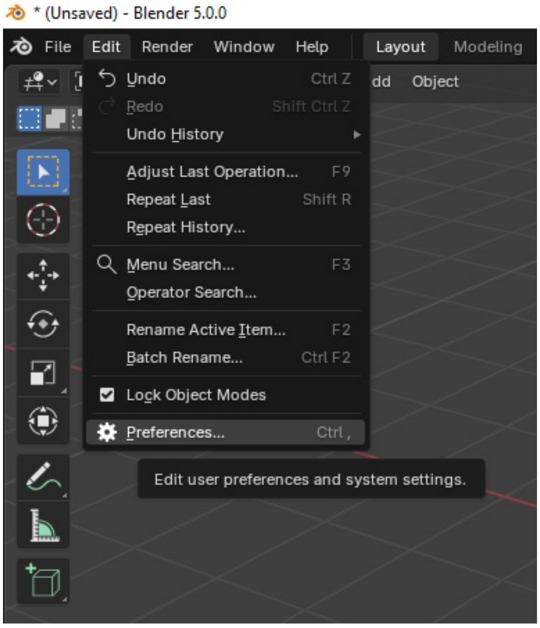
Changelog:

- fix no tracking without Lens Target
- fix crash if lens target specified but no lens file

blender_virtual_prod.zip	sha256:d3215d3294a9ca871...	462 KB	23 minutes ago
Source code (zip)			23 minutes ago
Source code (tar.gz)			23 minutes ago

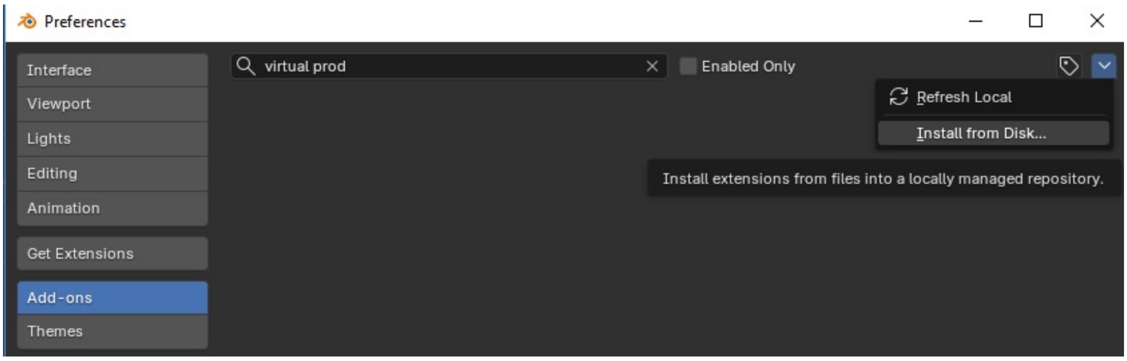
I.2. Install package in Blender

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



Edit user preferences and system settings.

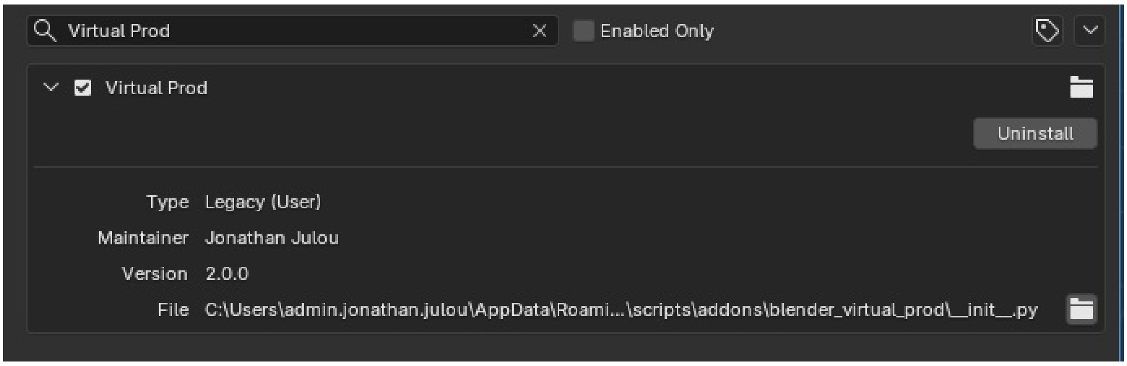
Open the **Add-Ons** tab, and click the little down arrow at the top-right. Select **Install from Disk...**



virtual prod

Install extensions from files into a locally managed repository.

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



Virtual Prod

☑ Virtual Prod

Type Legacy (User)

Maintainer Jonathan Julou

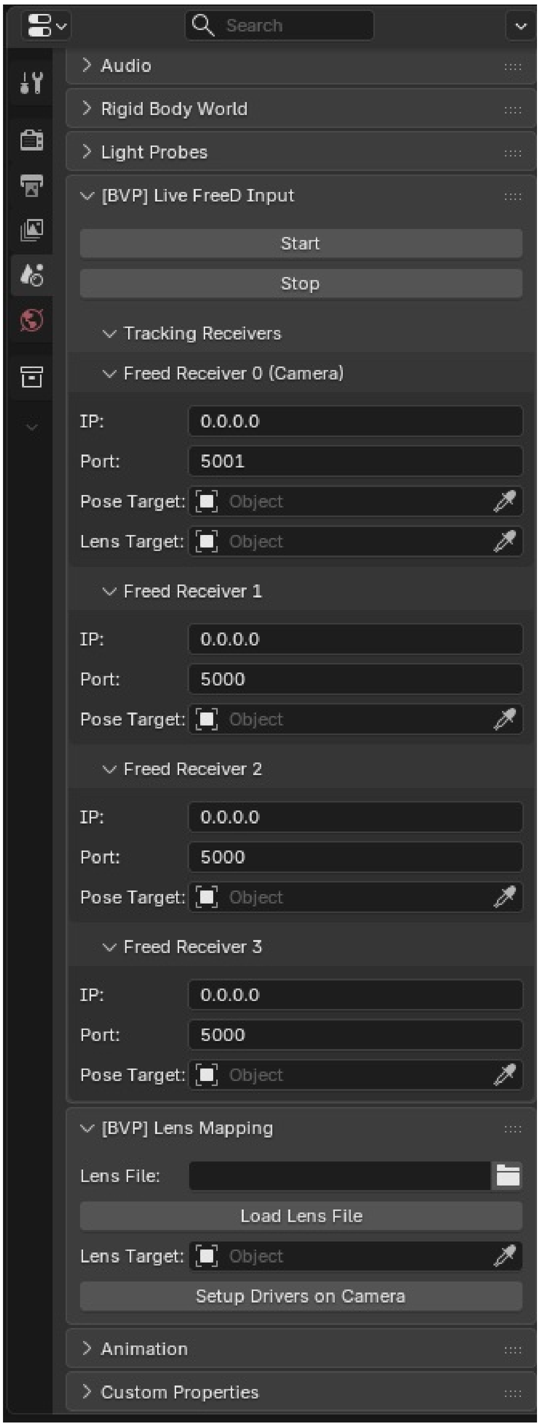
Version 2.0.0

File C:\Users\admin.jonathan.julou\AppData\Roami...\scripts\addons\blender_virtual_prod\__init__.py

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:



Search

> Audio

> Rigid Body World

> Light Probes

∨ [BVP] Live FreeD Input

Start

Stop

∨ Tracking Receivers

∨ Freed Receiver 0 (Camera)

IP: 0.0.0.0

Port: 5001

Pose Target: Object

Lens Target: Object

∨ Freed Receiver 1

IP: 0.0.0.0

Port: 5000

Pose Target: Object

∨ Freed Receiver 2

IP: 0.0.0.0

Port: 5000

Pose Target: Object

∨ Freed Receiver 3

IP: 0.0.0.0

Port: 5000

Pose Target: Object

∨ [BVP] Lens Mapping

Lens File:

Load Lens File

Lens Target: Object

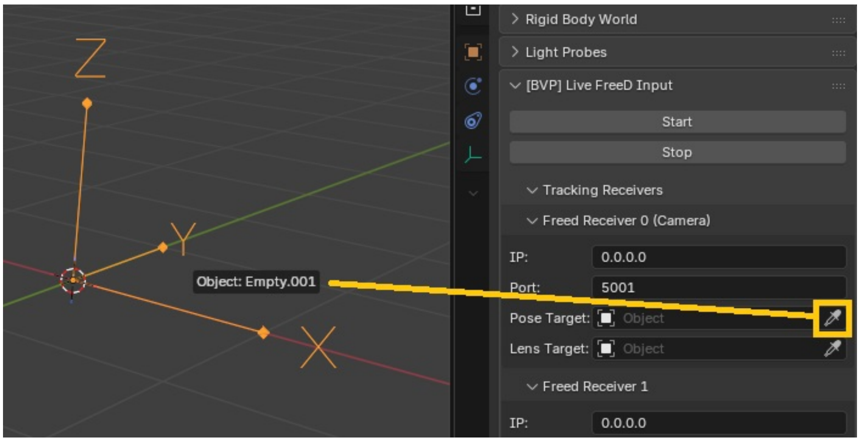
Setup Drivers on Camera

> Animation

> Custom Properties

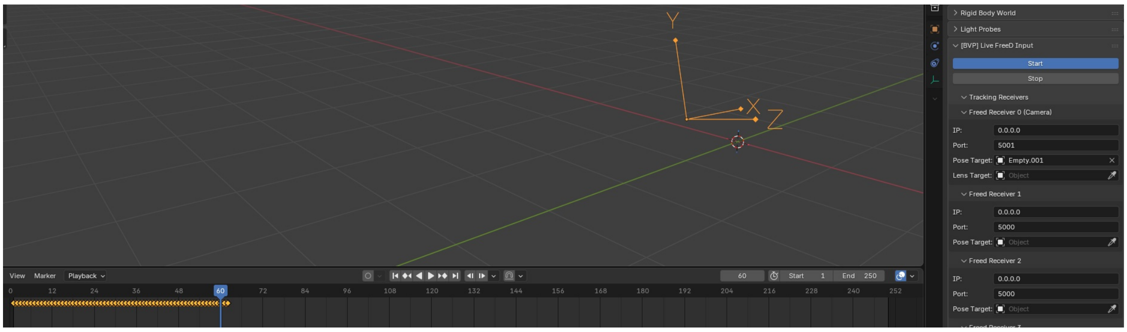
II.1. Basic pose tracking

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):



IP: 0.0.0.0
Port: 5001
Pose Target: Object
Lens Target: Object
IP: 0.0.0.0

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

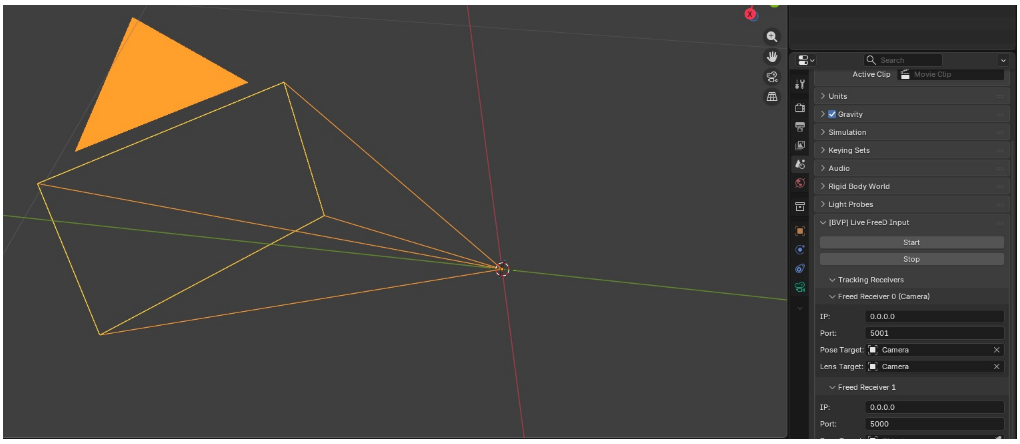


60 Start 1 End 250

IP: 0.0.0.0
Port: 5001
Pose Target: Empty.001
Lens Target: Object
IP: 0.0.0.0
Port: 5000
Pose Target: Object
IP: 0.0.0.0
Port: 5000
Pose Target: Object

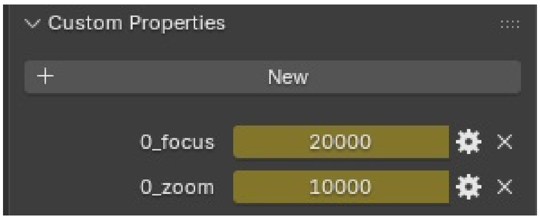
II.2. Basic lens tracking

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



IP: 0.0.0.0
Port: 5001
Pose Target: Camera
Lens Target: Camera
IP: 0.0.0.0
Port: 5000

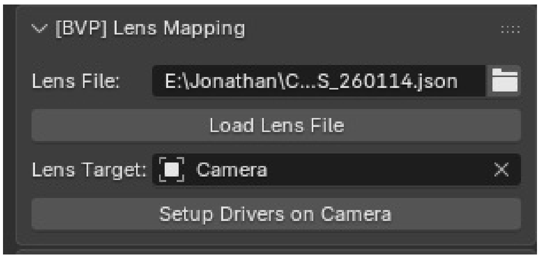
Now FreeD Zoom and Focus data is written as well as Custom Properties in the camera. Note that FreeD does not carry meaningful information like focal length or focus distance, so they still get the default value:



0_focus 20000
0_zoom 10000

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:



[BVP] Lens Mapping

Lens File: E:\Jonathan\C...S_260114.json

Load Lens File

Lens Target: Camera

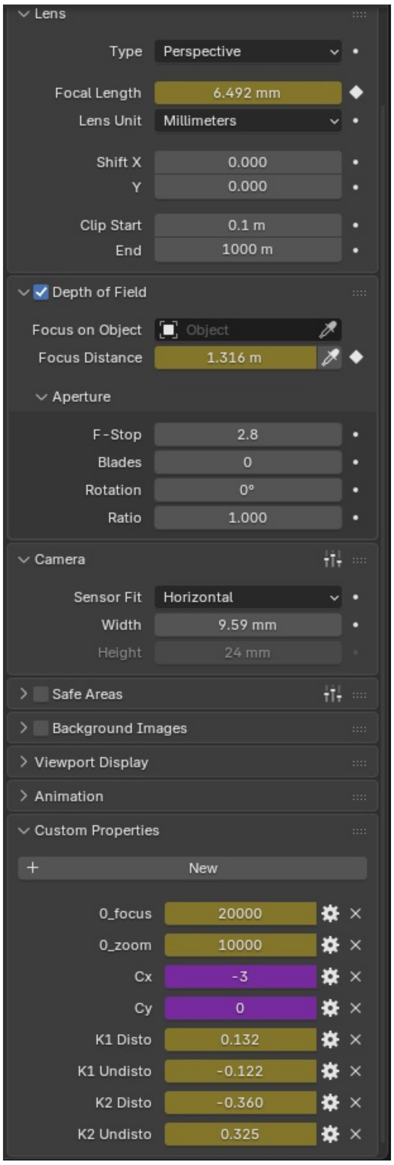
Setup Drivers on Camera

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:



Lens

Type: Perspective

Focal Length: 6.492 mm

Lens Unit: Millimeters

Shift X: 0.000

Y: 0.000

Clip Start: 0.1 m

End: 1000 m

☑ Depth of Field

Focus on Object: Object

Focus Distance: 1.316 m

Aperture

F-Stop: 2.8

Blades: 0

Rotation: 0°

Ratio: 1.000

Camera

Sensor Fit: Horizontal

Width: 9.59 mm

Height: 24 mm

Custom Properties

0_focus: 20000

0_zoom: 10000

Cx: -3

Cy: 0

K1 Disto: 0.132

K1 Undisto: -0.122

K2 Disto: -0.360

K2 Undisto: 0.325