Camera Projection Painter

Vladlen Kuzmin (ssh4), Ivan Perevala (ivpe)

Feb 01, 2024

GENERAL

Add-on for Blender for working with photo scans and photogrammetry.

Project on GitHub

Project Releases

CHAPTER

ONE

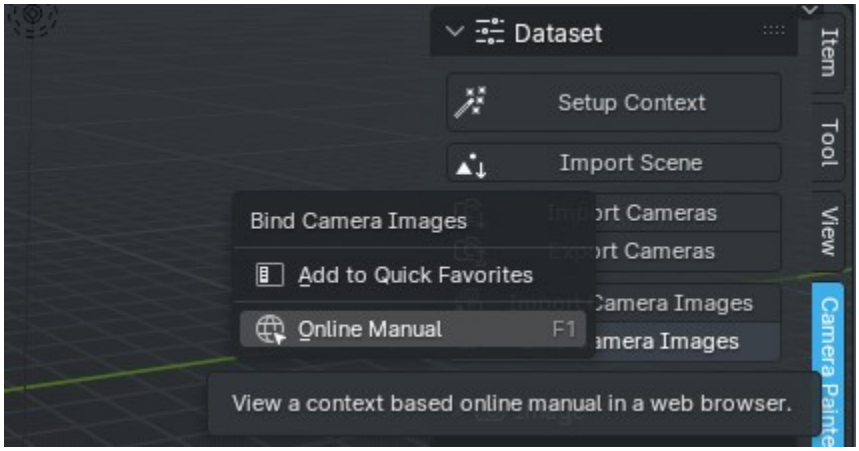
TABLE OF CONTENT

1.1 About

1.1.1 How To Use This Documentation

This user manual is an integral part of the add-on itself and contains a description of operators and their work, simple lessons that will help you start using the add-on, and automatically generated descriptions of properties.

Everything is documented - that is, you can simply hover over any add-on element (operator or property) while working, press the help key (`F1` in Blender's standard layout), or call the documentation from the context menu:



and get information about this element.

Note: For some properties, you can find more information here, as (as of version 3.6) Blender tooltips are limited to 400 ASCII characters. That is, for the localized version with utf-8 characters, it is even less. This is due to Blender performance, but it's hard to fit technical information into little more than a Twitter post (R.i.P.), so the full version of the descriptions for them can be found here.

1.1.2 History

Development began in 2019. Vlad wrote that sometimes there is a need to improve certain areas automatically textures generated by photogrammetric programs and currently he uses photoshop for this, but it is not convenient, but a more acceptable option at that time is Substance Painter, it just has a limit for the size of textures, so for him goals simply do not fit. We discussed for a long time how best to solve the dilemma and decided to implement it with help Blender drawing mode. It is open source, has a good community of users and is developing quite quickly.

In November 2019, we released the first release of the addon, then it was still called "Scan Paint" and in fact, it was quite a simple implementation that used object modifiers for object projection, but the basic concepts were preserved. The main disadvantage of this addon was that it was necessary to export images with distortion, which significantly increased the number of temporary files on the disk and created additional inconvenience.

In October 2020, Camera Projection Painter 0.1.5 was released, which was implemented in a radically different way, namely - its main part was written in C++, as a module for Python. This created a significant advantage - was added automatic distortion of the projected object, faster generation of preview images, faster search object and image name matches, but this also created a big problem with the distribution and update of the addon. The problem was that once Blender started using updated versions of the Python ABI, the module was no longer available use. Also, the problem was that Blender is distributed on different platforms - that is, it should be assemble the module for each platform separately.

In any case, version 0.1.6 lasted quite a long time (3 years), although it required using an older version Blender.

On 01.03.2024, the first release was released that used only the standard Blender API. It didn't change the workflow too much, however brought support for more camera data files, double-precision floating point data storage, camera selection in texture drawing mode by simple selection instead of gizmo.

1.2 Installation and Updates

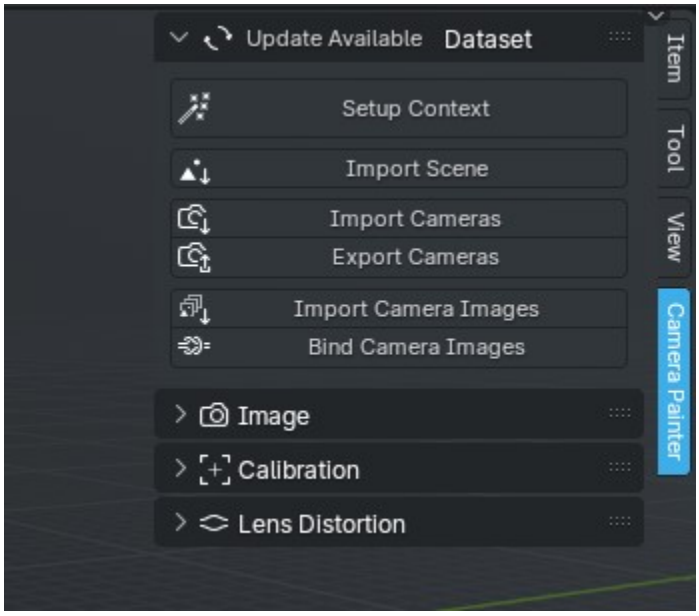
1.2.1 Installation

You can install an addon in a standard way. There is nothing complicated here

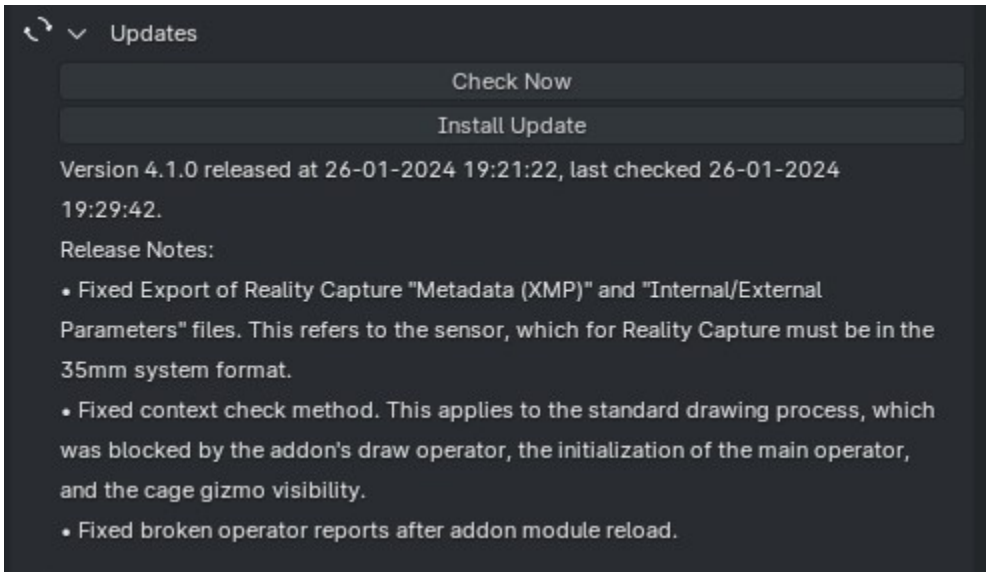
1.2.2 Updates

Everything is a little more complicated here, but not much. Obviously, most users only installing an addon once, after updating the Blender, and then when the current version stops working on newer or from some sources learns about new critical features. This addon has its own update system for users who do not use Git.

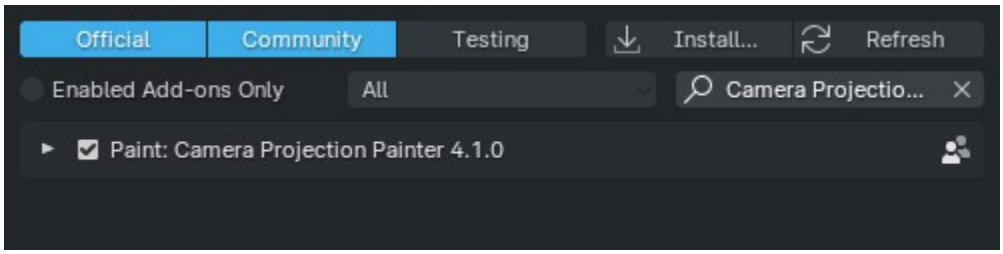
The updates are checked automatically when you open the Blender (don't worry, this is a background process, so it does not interfere with work). If the system detects an update, a message will appear in the user interface:



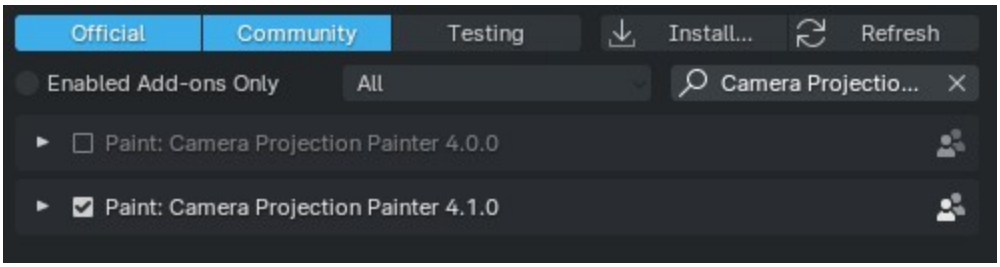
If you click on it then information about the update would appear:



The upgrade is also in the standard method, but automatically:



Other update systems modify current files, instead the previous version will simply be disabled but will not be changed or deleted any files:



That is, if for some reason you want to return to the previous version, you must turn off the updated and only then - turn on the previous.

1.3 Addon Basics

1.3.1 Data Blocks

The types of data blocks with which the addon works are described here. When it comes to the concept and definition of data types, in Blender, unlike photogrammetric software, everything is a little more complicated. We have an image on disk - it was used to reconstruct the scene, and maybe opened in the current `.blend` file (or maybe even packed into the file as an archive). There are also files exported from photogrammetric software with data from the cameras from which these shots were taken - there are data on the position of the cameras in space, the length of the lens, the type and parameters of lens distortion, etc.

In order to tie everything together to work in Blender, using standard data blocks, the following hierarchy is used:

```
Scene
└── Object of type "Camera"
    └── Camera (Data block)
        └── Image
```

It is clear that we have several camera objects in the scene, but each of them can have different or the same camera data block - this also applies to just Blender, without this addon. The addon adds the image associated with the camera data block to the standard hierarchy.

So, which data block does Blender store which information?

Object

Data about extrinsic parameters. This is the position and orientation of the camera in space.

Camera (Data Block)

Data about *intrinsic* camera parameters. This is information about the lens, as well as some additional parameters for import-export from third-party software.

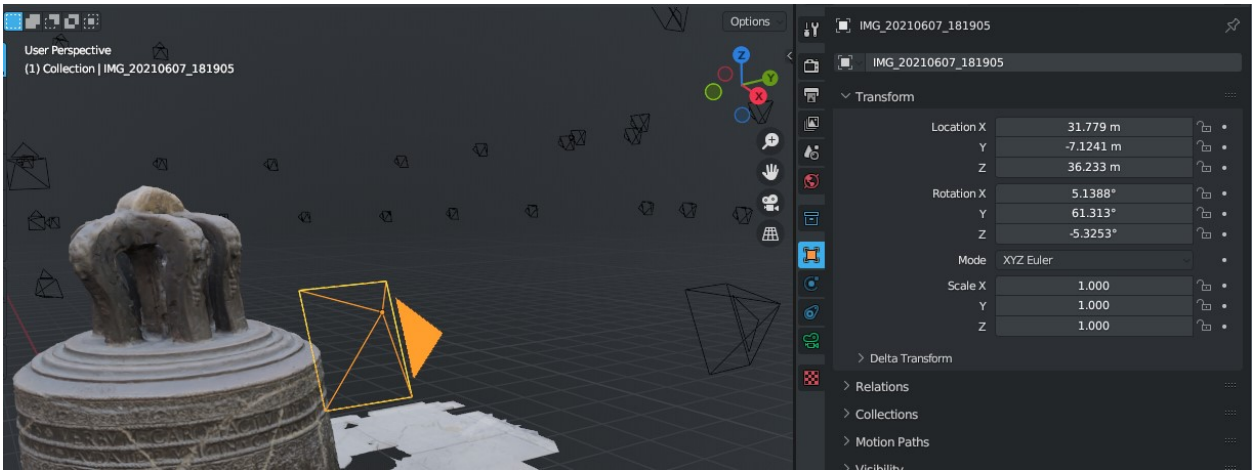
Image

Pixel data (obviously). It is important here that for Blender an image is just a block of data, of which we are primarily interested in its name and file path.

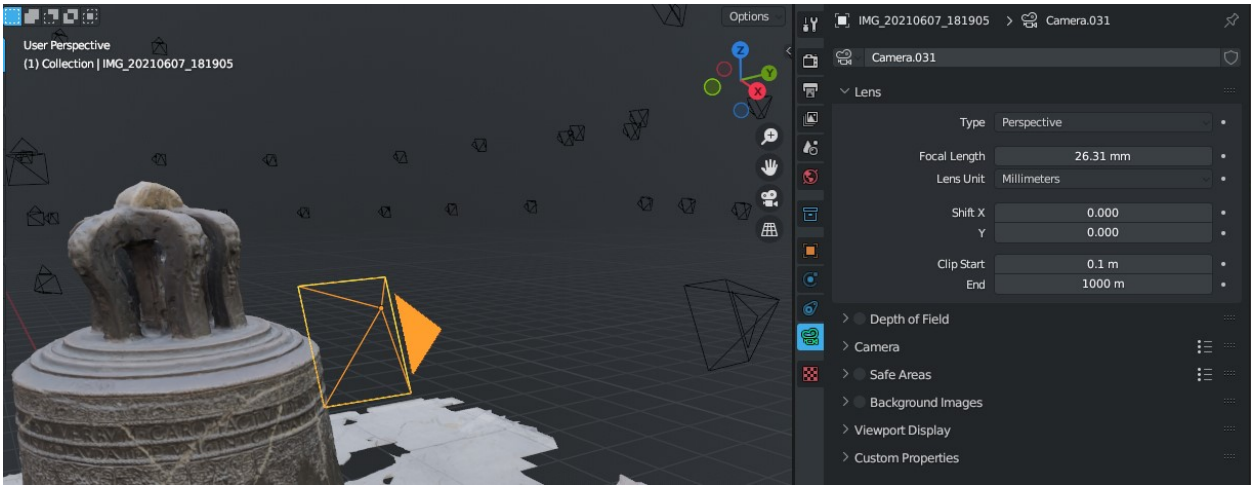
1.3.2 Names Comparison

Now that there is an understanding of the basic concepts, let's move on to the topic of name comparison. As is clear from the text above, not one block of data is used for one image, as in photogrammetric software, but three. And each of them has its own name.

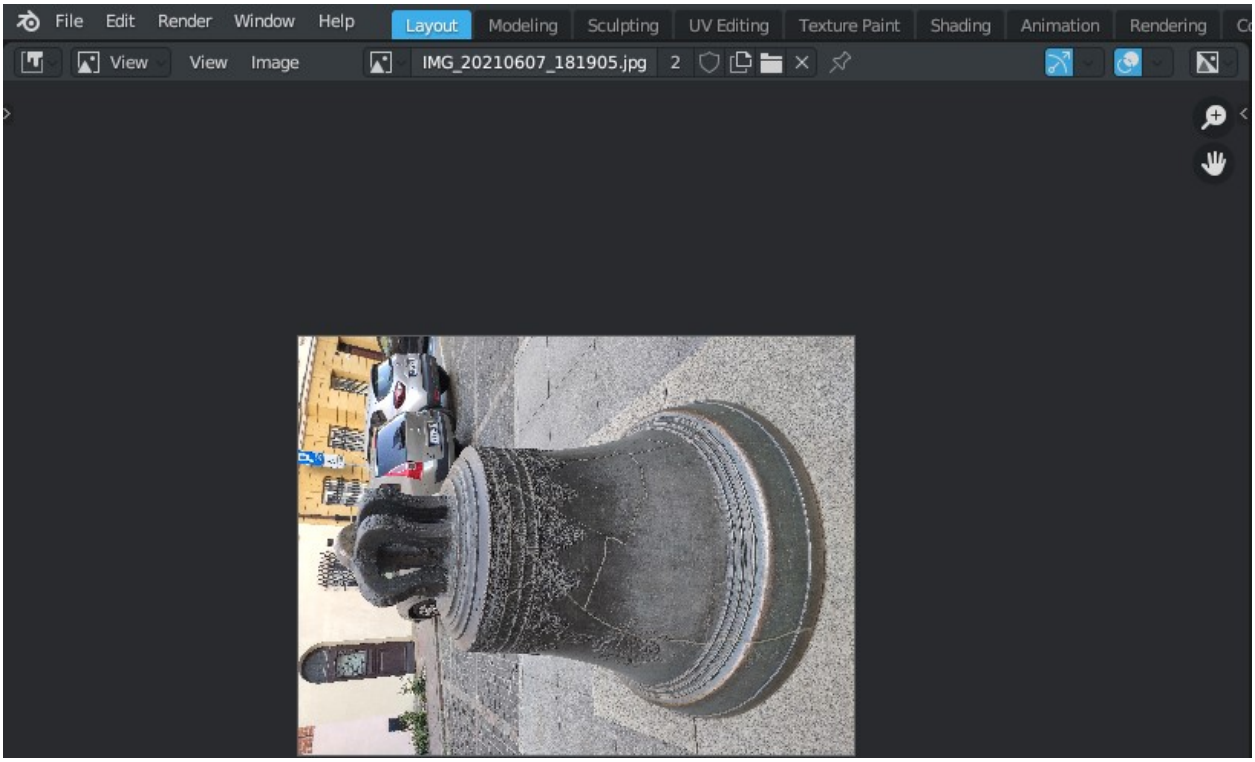
- *Camera objects* that have been imported, for example, via the `FBX` format, will most often have image filenames, but possibly no file extension. The objects that were created using the *import camera operator* will not have a file extension in the name.



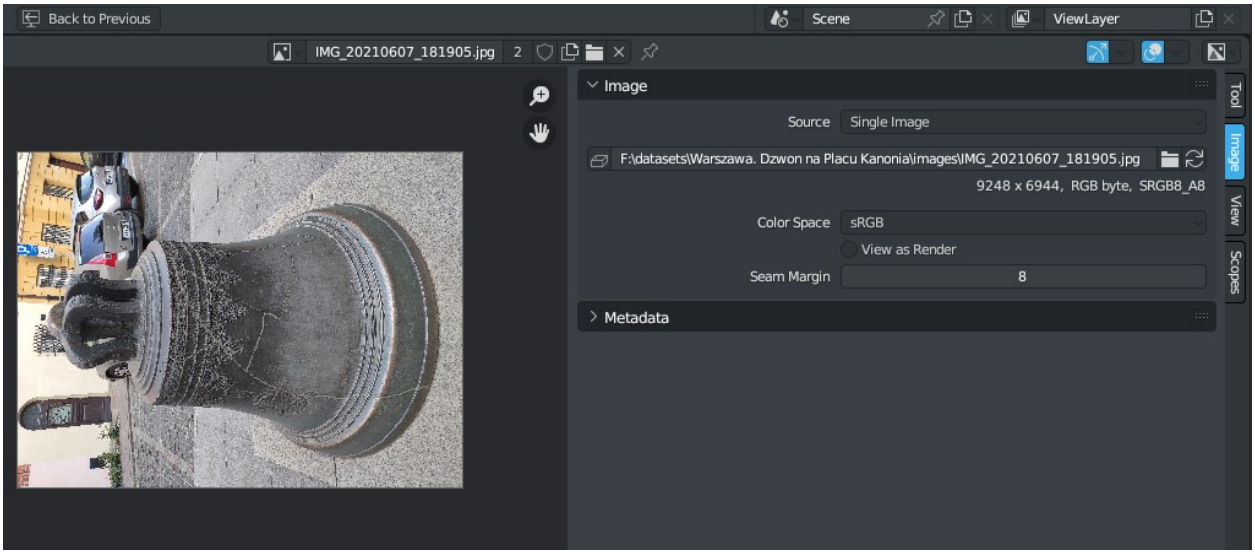
- *Camera data blocks* most often have standard names (*Camera*, *Camera.001*, ...) after importing the scene, and probably - after *importing camera data*, but for some file formats, such as `FBX` exported from Reality Capture, it will be the name of the image without the file extension.



- *Images* are most often named the same as image files, but there are quite possible situations when the image data block in Blender has a completely different name (for example, if you open one image or create a generated one and then specify the path to another for the same data block).



- *The image file path* always points to a specific file on disk, so obviously the name matches exactly. But this path may not be on disk because the image may have been packed into `*.blend` and the file may have been moved or deleted.



Regarding the different types of exported camera data files, there can also be different camera names. They can be with or without a file extension, they can be with a changed character register. For example, Reality Capture Metadata (XMP) files are named the same as images, but obviously have a different file extension because they are `xml` documents rather than images.

All this information is necessary to understand how the add-on automation works. This applies to *importing and binding cameras and images*, *importing camera data*.

1.4 Quick Start

Standard and recommended methods of working with the add-on are described here. Of course, no one forces you to use them - here you can experiment at your own discretion.

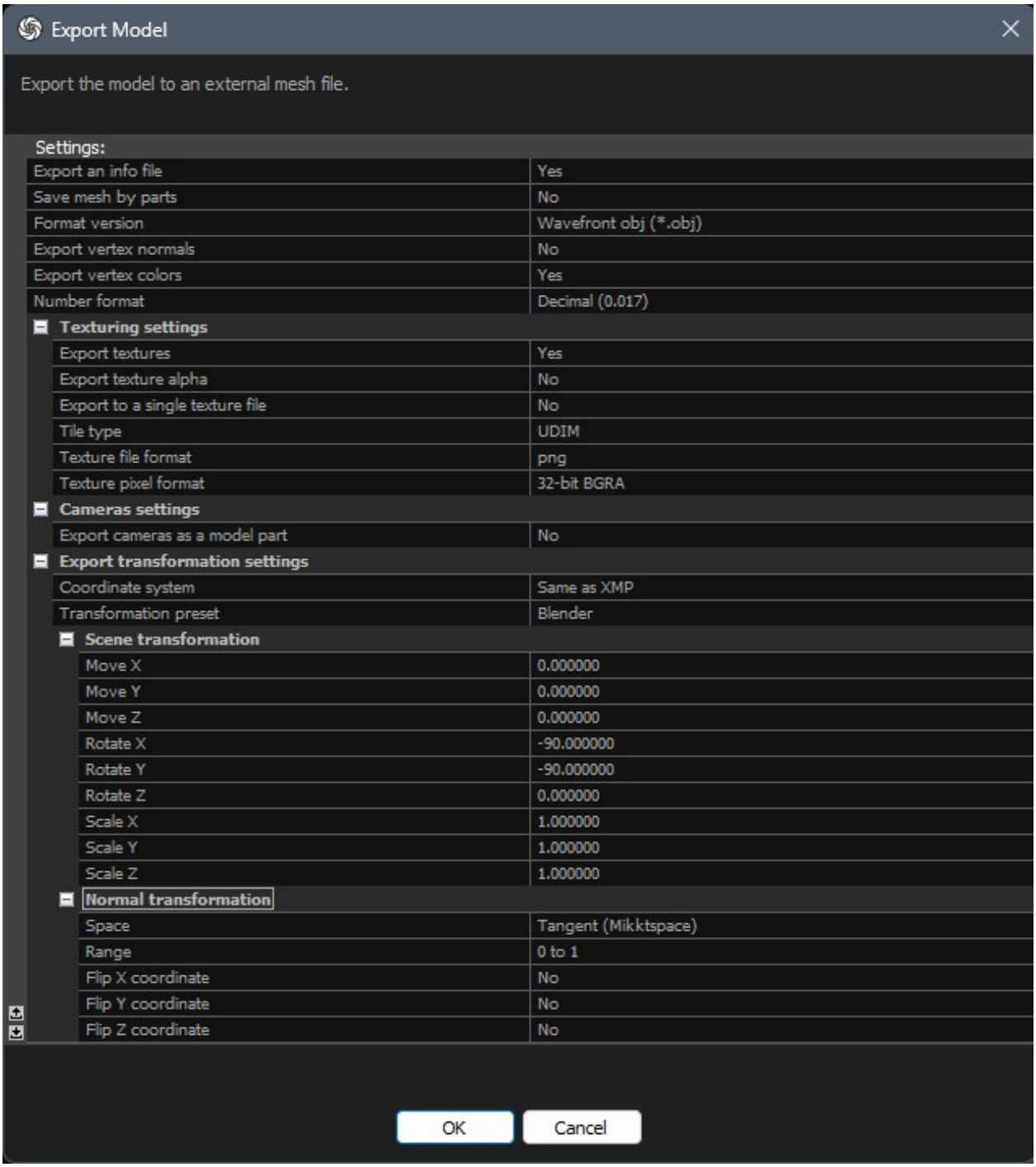
Reality Capture

- **Export Object Wire**

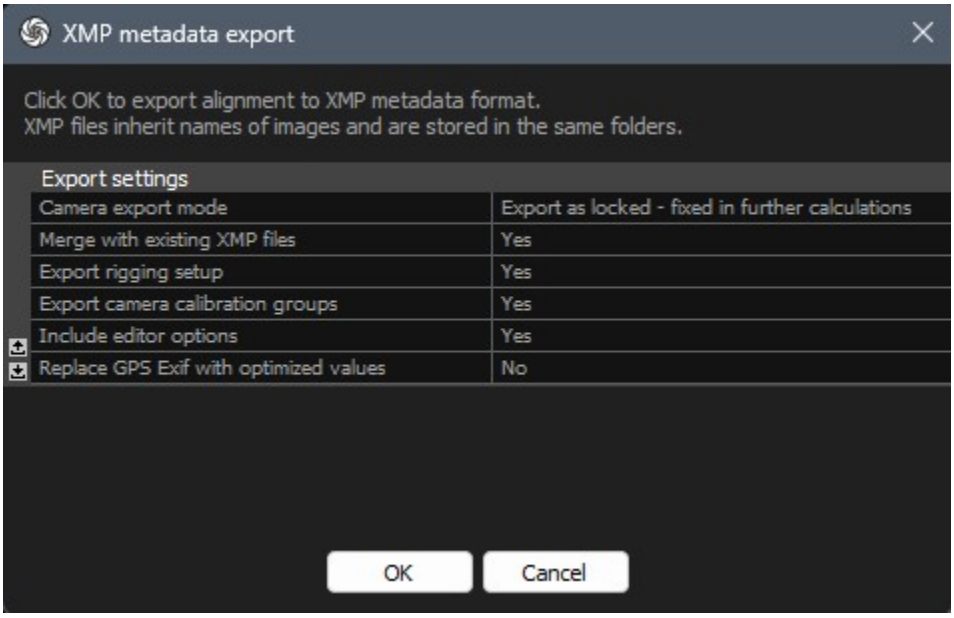
In order to simplify the task when importing the model to Blender, we export the model as a Wavefront (.obj) file. In newer versions of Blender (3.2+), this type of file is imported quickly and contains only the data needed to work with the addon. The important options here are:

- `Coordinate System` - `Same as XMP`
- `Transformation Preset` - `Blender`

These will be used *as presets for the* import from Reality Capture when *importing the scene*.

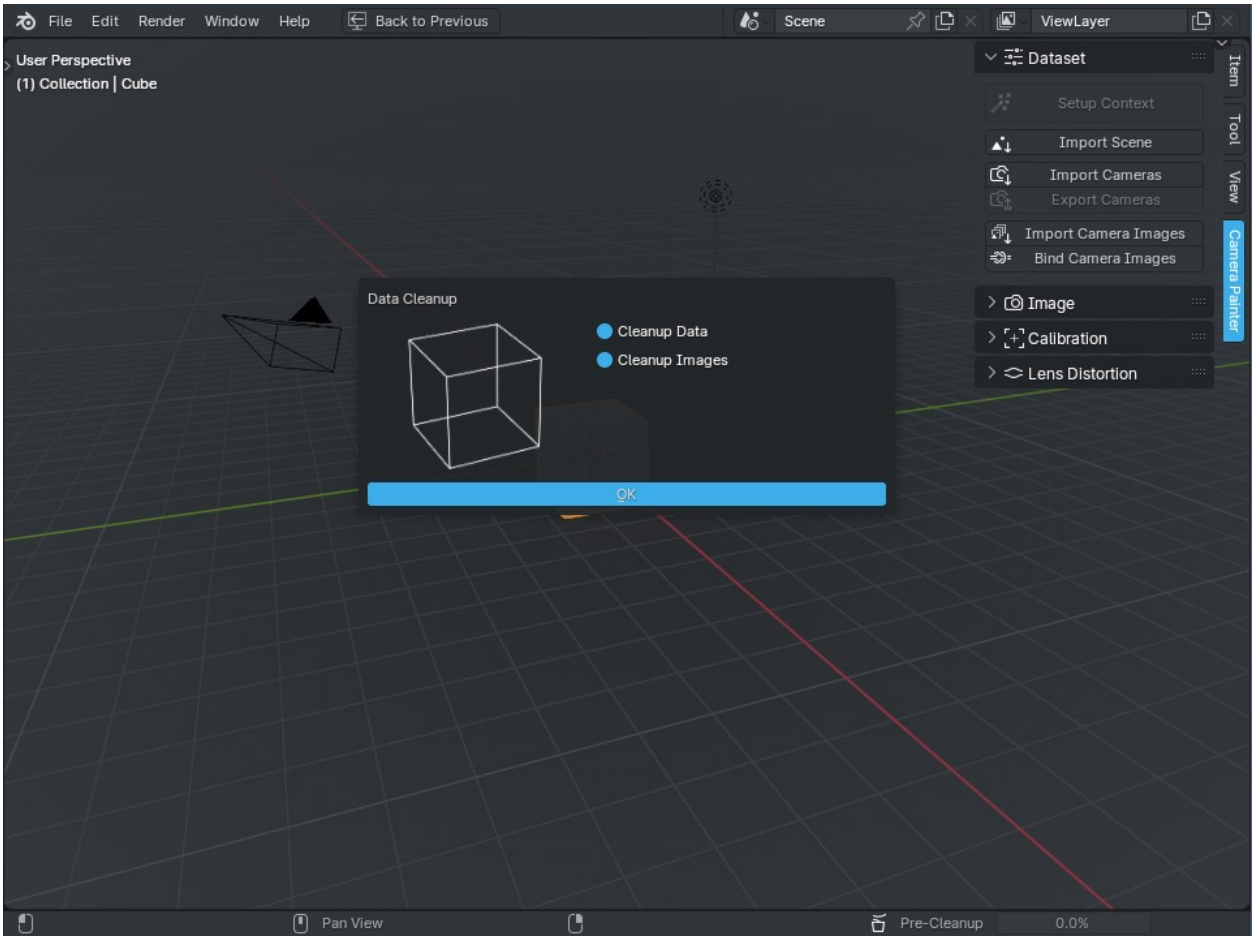


- **Export Camera Data**

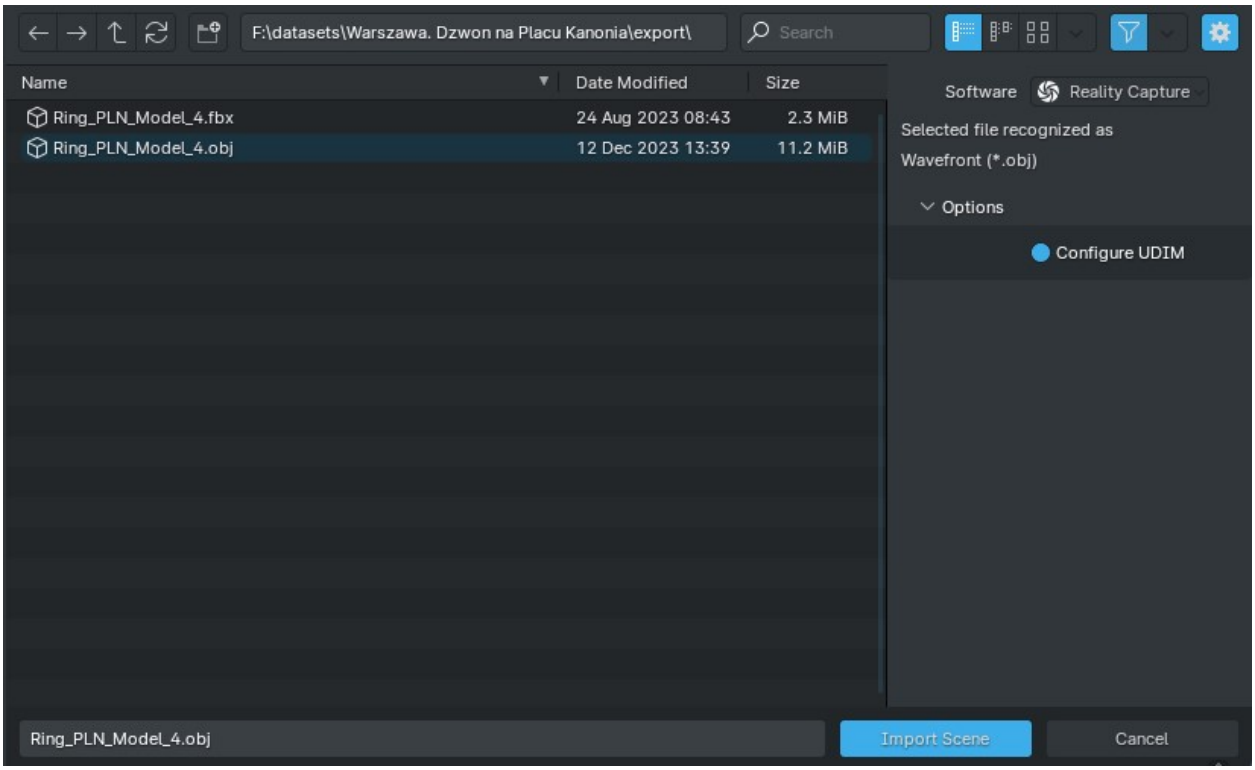


- **Setup Context**

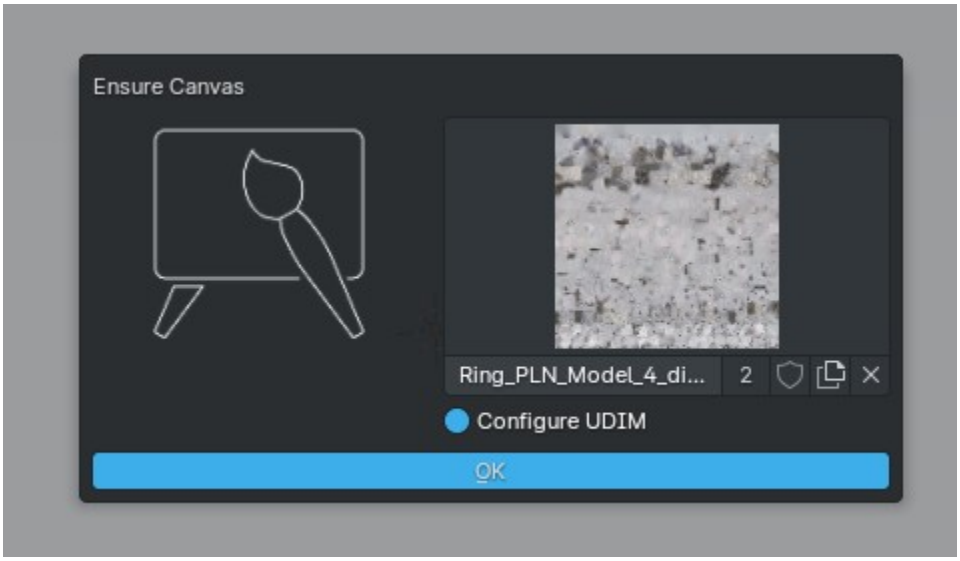
Next, you need to import this data. Start Blender and run *setup context*. There is nothing in the standard scene that will help us in our work, so this data can be deleted. This should also be done if, for example, you have finished adjusting one scene and want to move on to the next dataset. Therefore, the first question will be whether to clear the existing data:



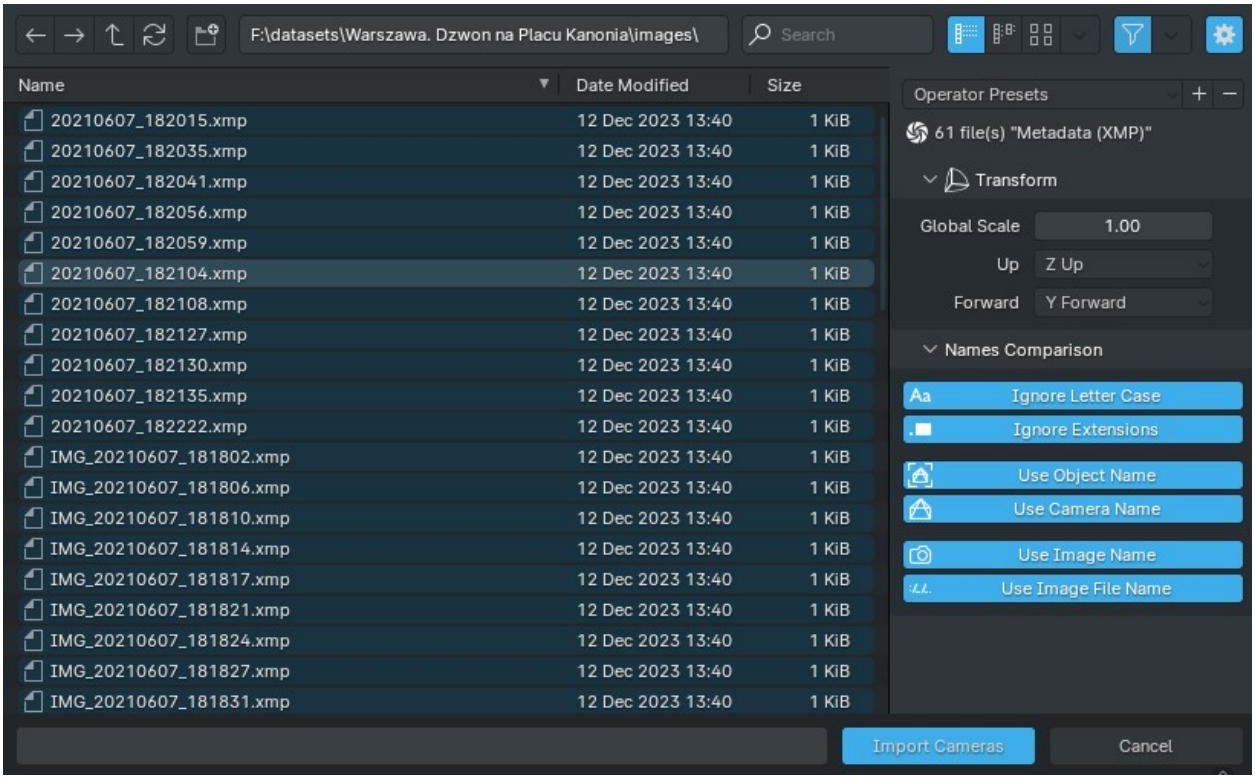
Next, select the file containing the object wire and import it.



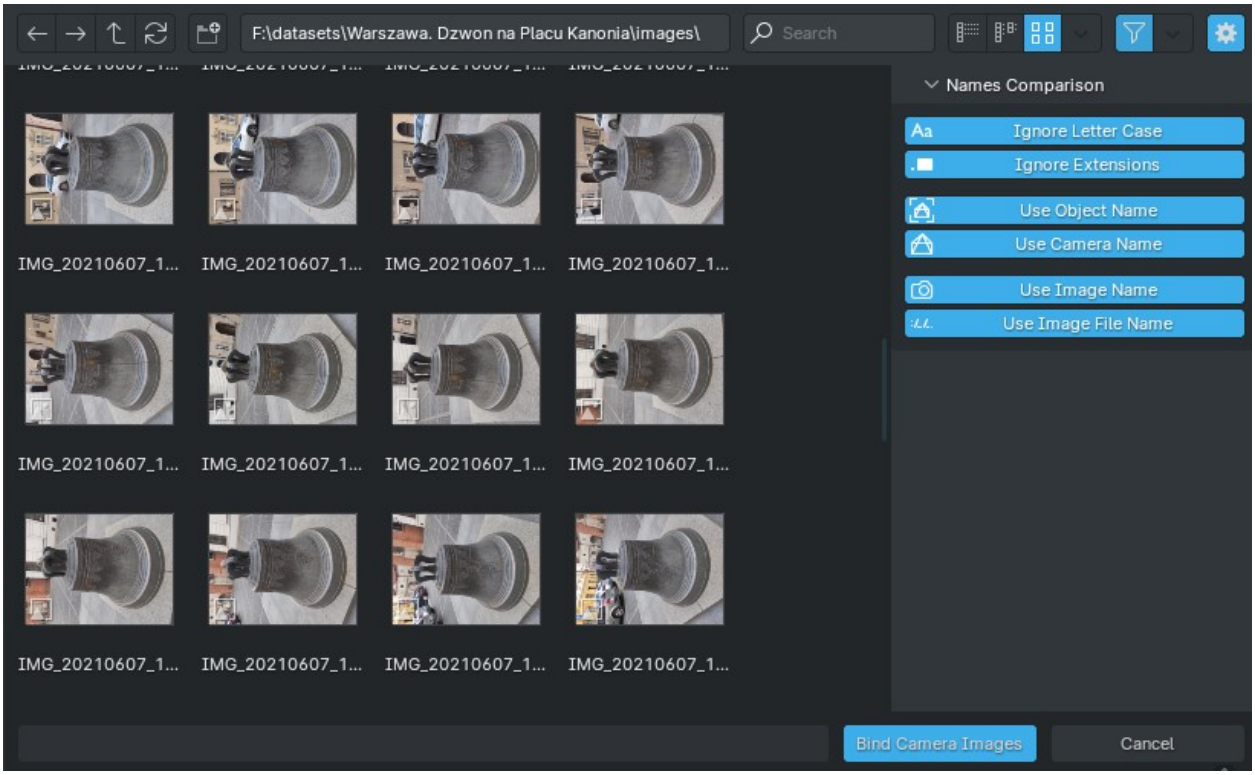
Then you need to choose the texture that we will adjust.



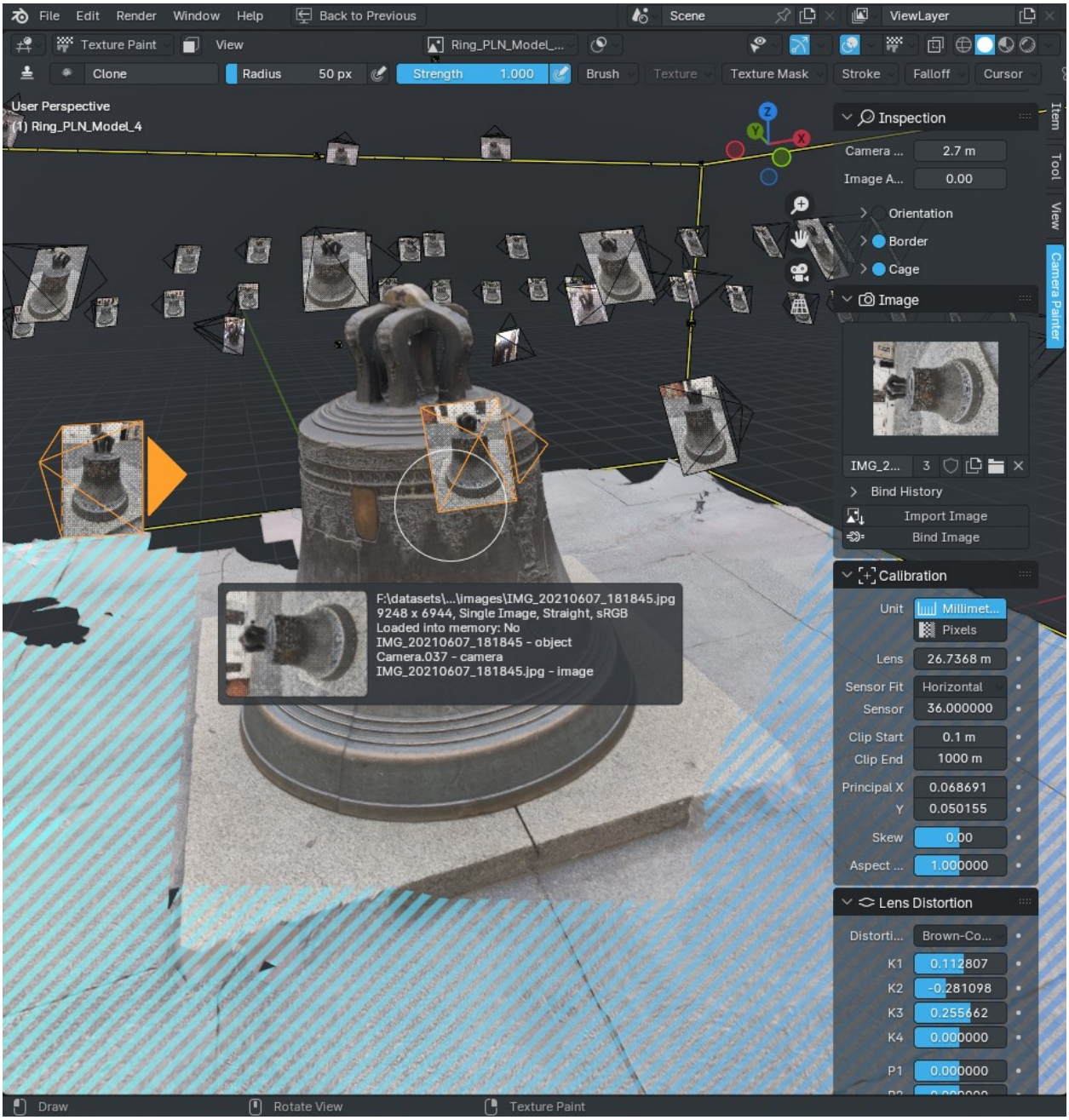
Next, we import camera data. This will allow you to use lens distortion without exporting distortion corrected images.



The next step will be linking the images to the camera objects.



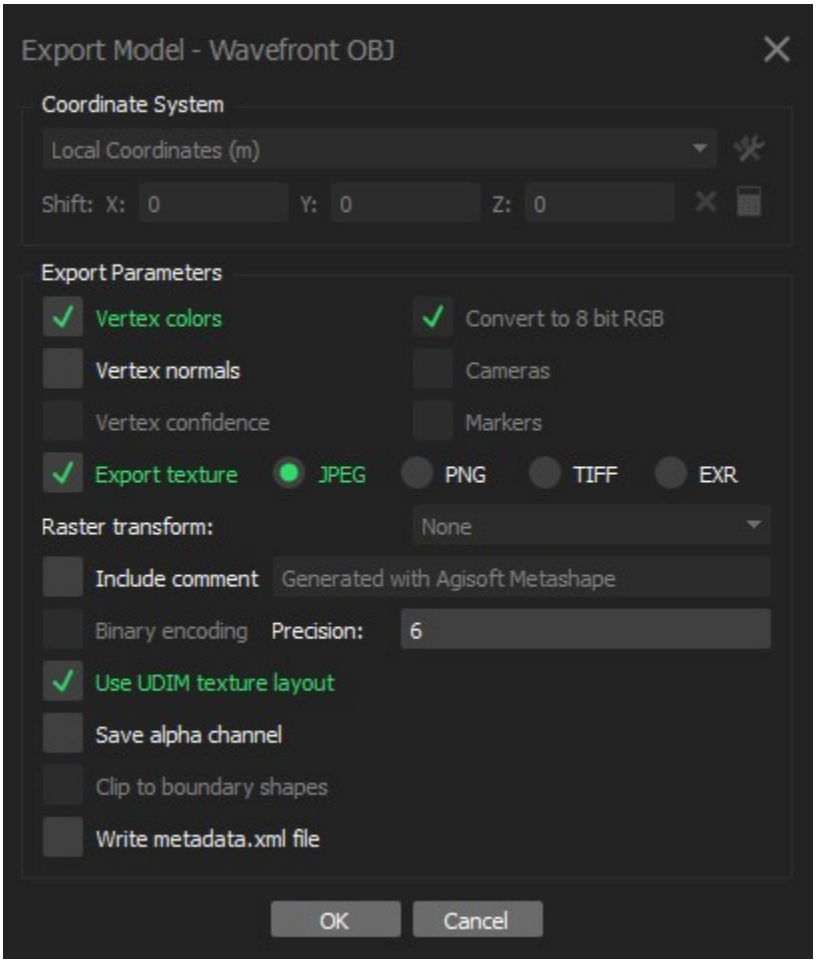
And after that you can start adjusting the texture.


File Edit Render Window Help
Back to Previous
Scene
ViewLayer
Texture Paint
View
Ring_PLN_Model_...
Clone
Radius 50 px
Strength 1.000
Brush
Texture
Texture Mask
Stroke
Falloff
Cursor
User Perspective
(1) Ring_PLN_Model_4
Inspection
Camera ... 2.7 m
Image A... 0.00
Orientation
Border
Cage
Image
IMG_2... 3
Bind History
Import Image
Bind Image
Calibration
Unit Millimet...
Pixels
Lens 26.7368 m
Sensor Fit Horizontal
Sensor 36.000000
Clip Start 0.1 m
Clip End 1000 m
Principal X 0.068691
Y 0.050155
Skew 0.00
Aspect ... 1.000000
Lens Distortion
Distorti... Brown-Co...
K1 0.112807
K2 -0.281098
K3 0.255662
K4 0.000000
P1 0.000000
F:\datasets\...\images\IMG_20210607_181845.jpg
9248 x 6944, Single Image, Straight, sRGB
Loaded into memory: No
IMG_20210607_181845 - object
Camera.037 - camera
IMG_20210607_181845.jpg - image
Item
Tool
View
Camera Painter
Draw
Rotate View
Texture Paint

Agisoft Metashape

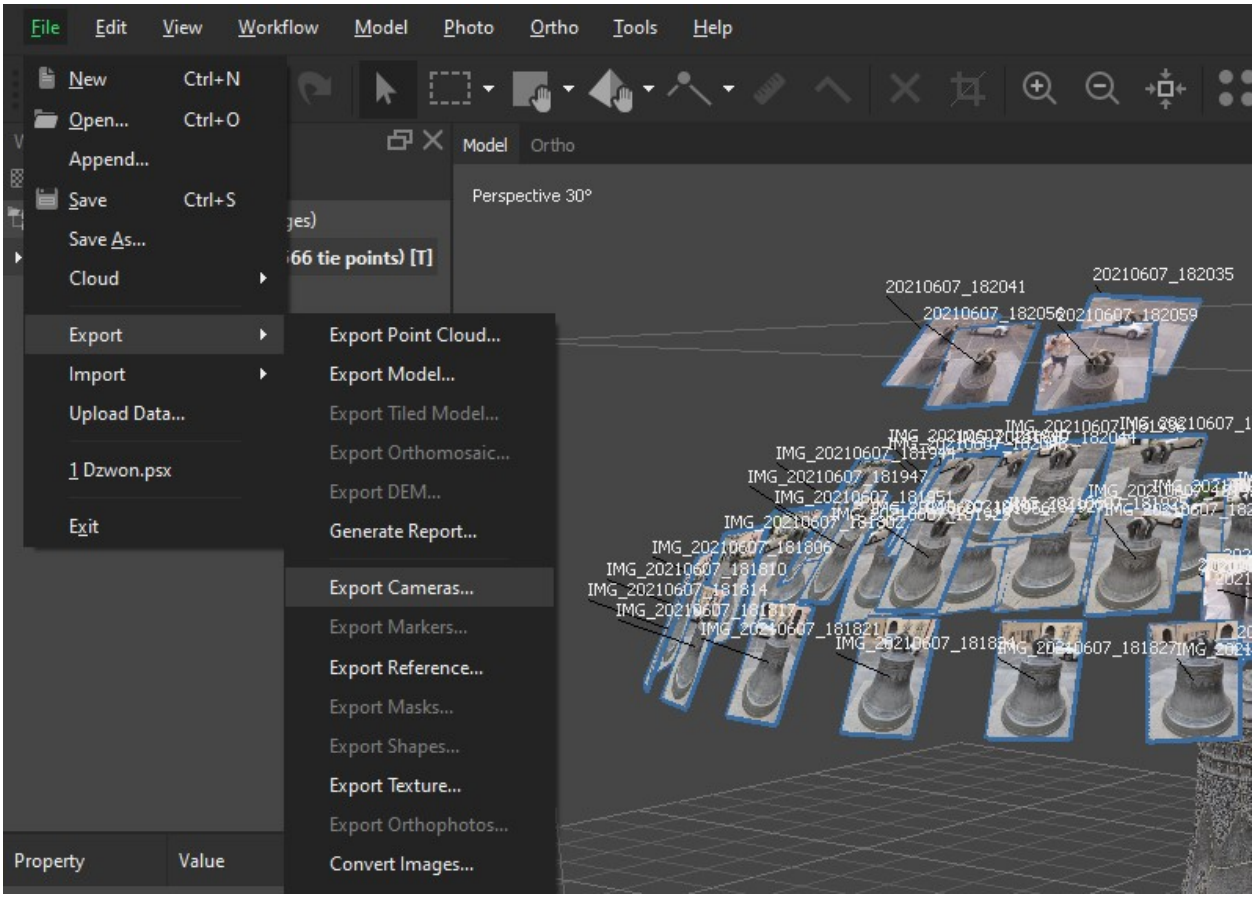
- **Export Object Wire**

Export the reconstructed scene. It is recommended to use the Wavefront (.obj) format for this, as Blender 3.2+ will import this file type the fastest and it can store all the necessary mesh data and not contain extraneous data such as camera position information that needs to be imported separately.



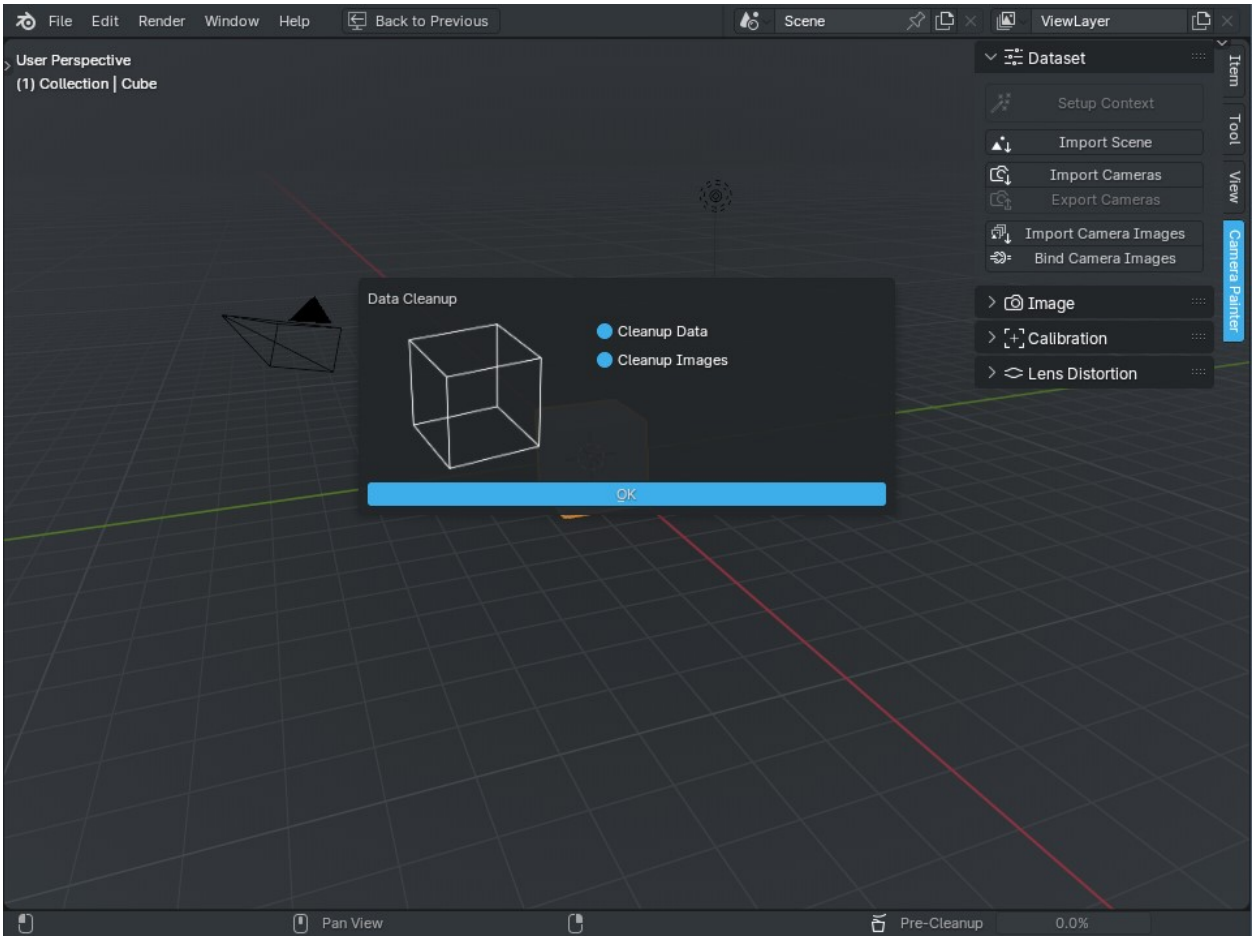
- **Export Camera Data**

Next, we export the camera data, for this we will use the Agisoft XML format.

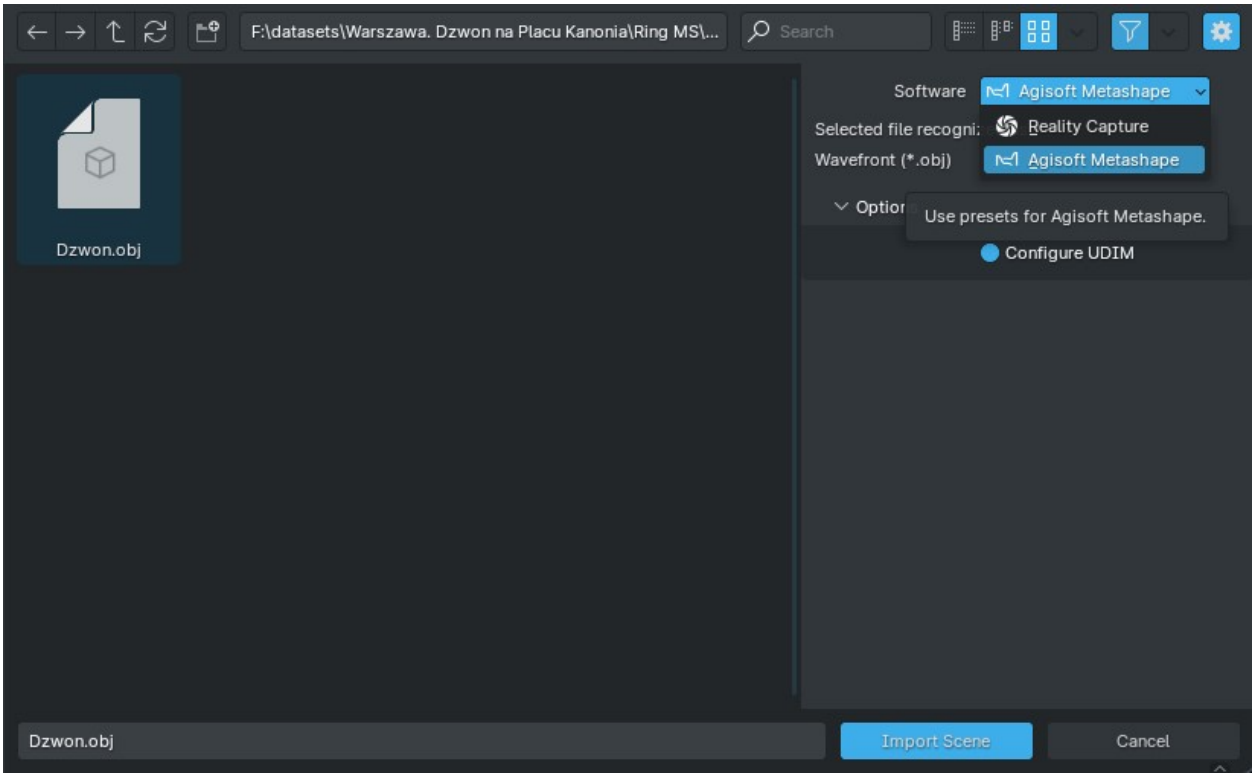


- **Setup Context**

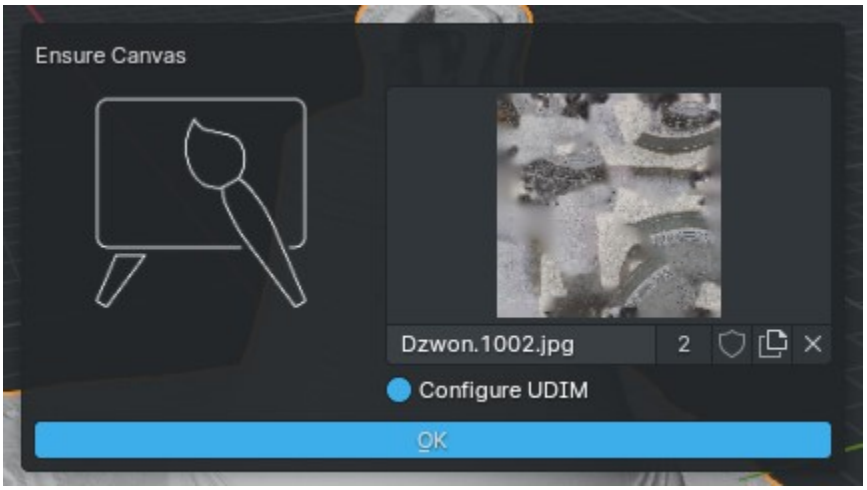
Next, you need to import this data. Start Blender and run *setup context*. There is nothing in the standard scene that will help us in our work, so this data can be deleted. This should also be done if, for example, you have finished adjusting one scene and want to move on to the next dataset. Therefore, the first question will be whether to clear the existing data:



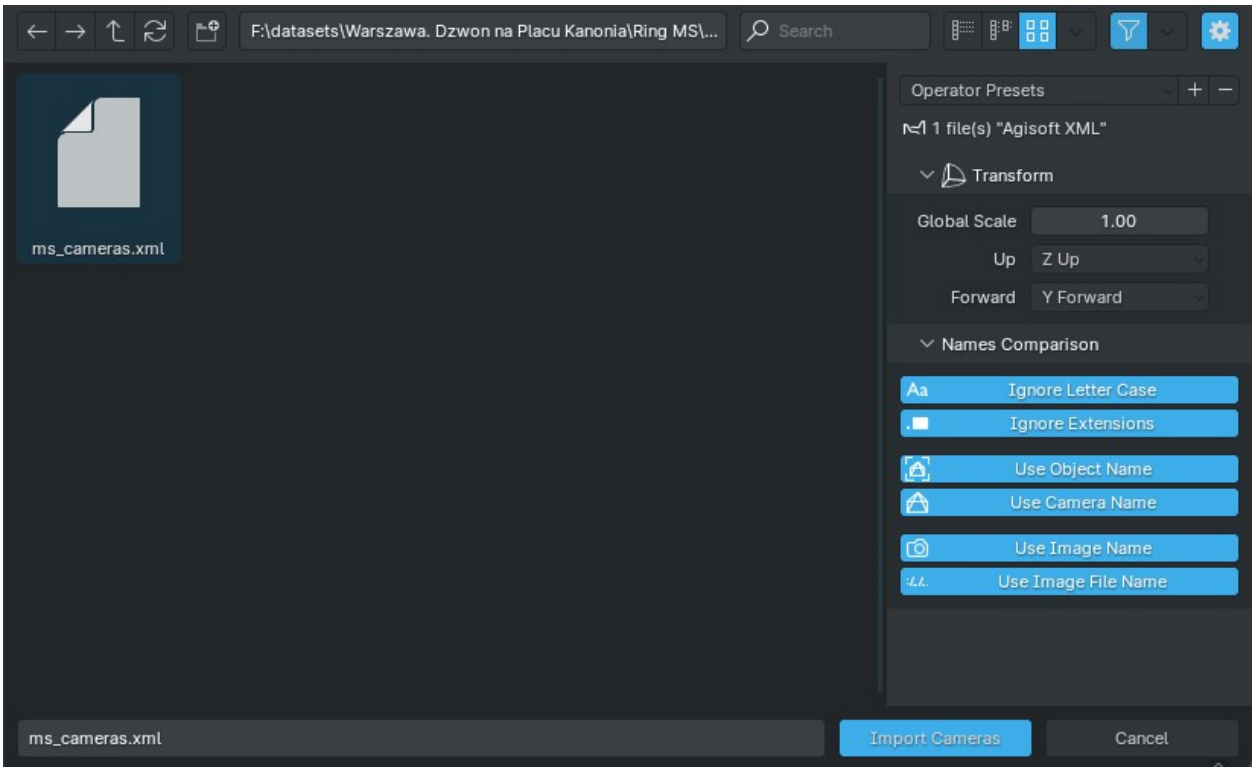
Select the previously exported scene file and the presets for Agisoft Metashape.



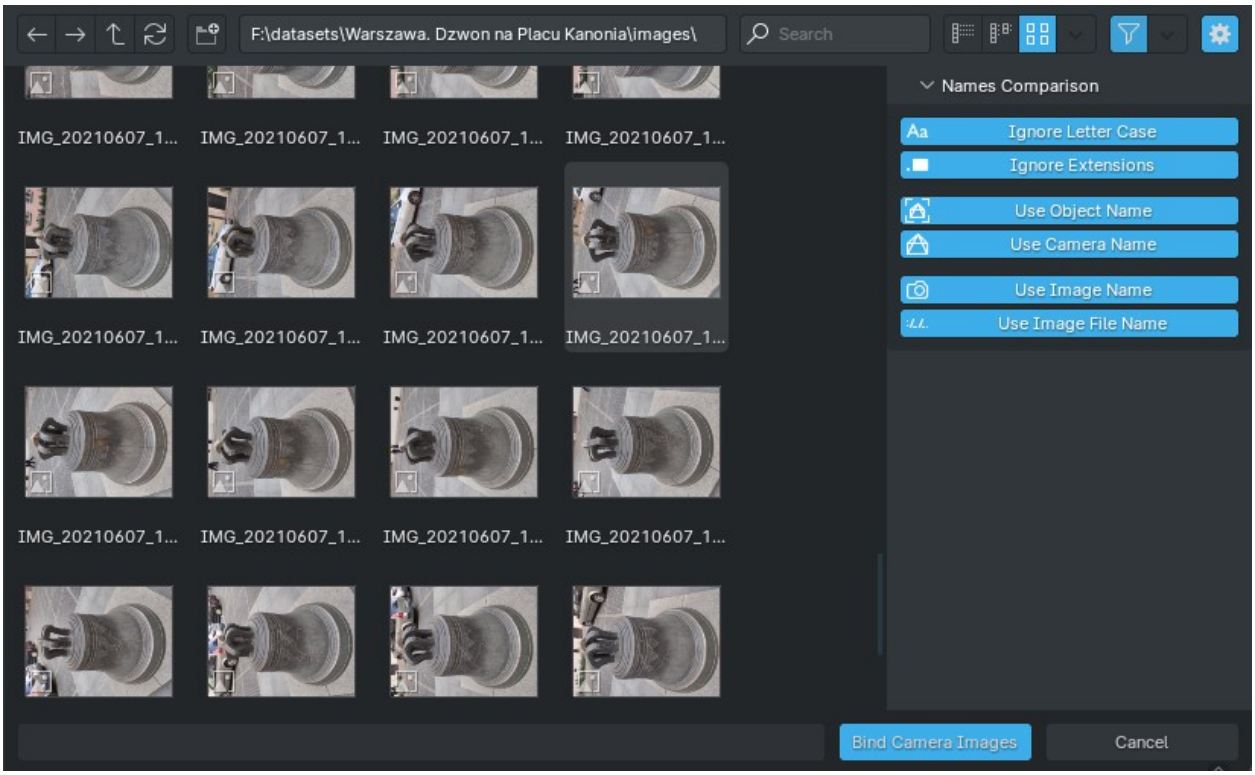
Next, it is necessary to choose the texture that we will adjust.



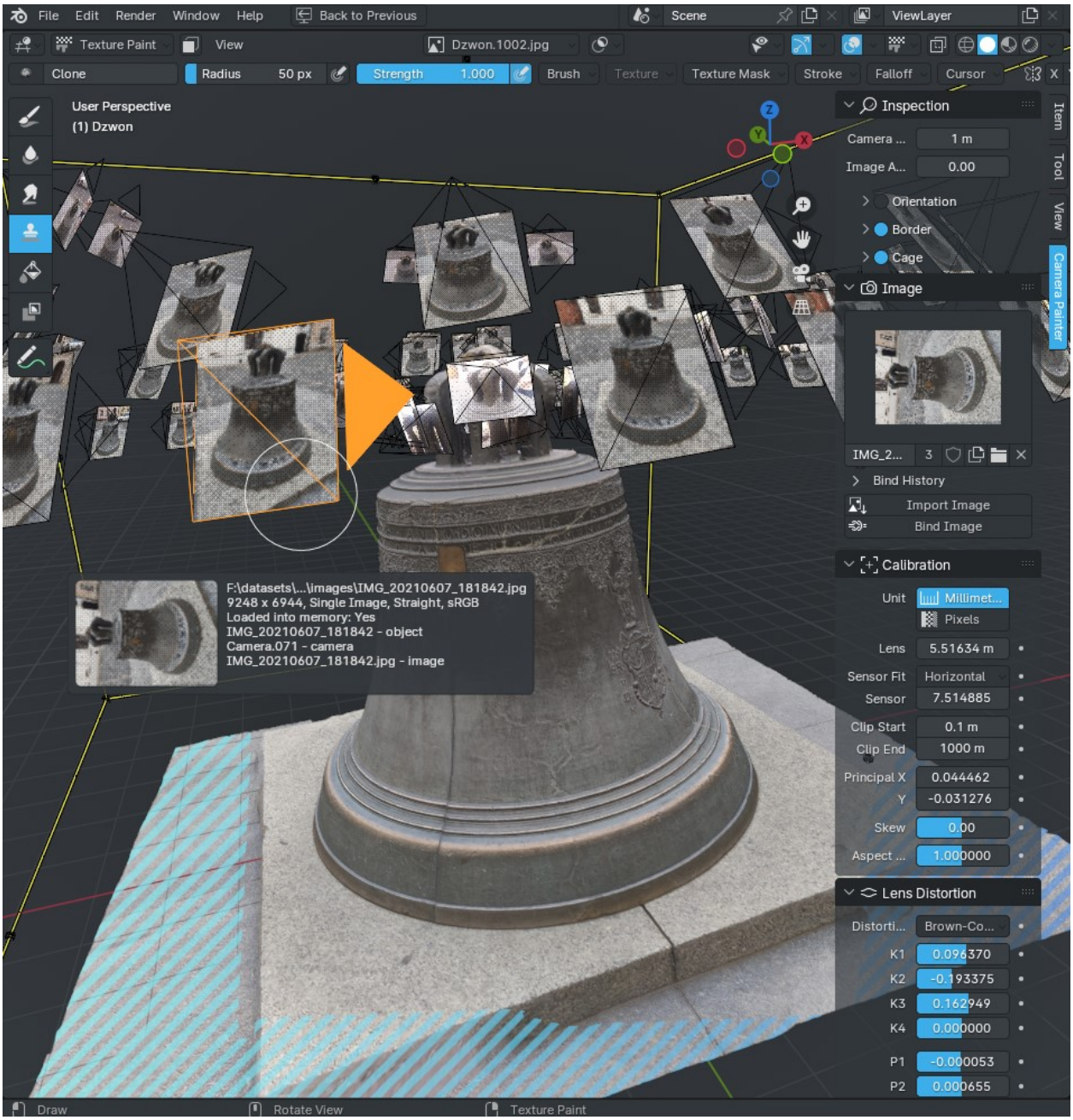
Then import camera data from an XML file.



Finally, you need to bind the image to the cameras. To do this, select the directory in which the images are located.



This completes the context setting, you can adjust the texture.



1.5 Answers

Here you can find answers to questions previously asked by users. We respect the privacy of the author and the source of the question is not indicated, but the questions themselves are verbatim, with minor editing (for example, the greeting was removed).

If you yourself have questions, you can ask them through the tracker on GitHub (this is the most official way) or through social networks.

In the github description you say "you can adjust the texture and camera data", what do you mean by that? I'm

interested in what this addon is, but it's hard to piece the meaning together from the descriptions

It allows you to paint over the texture of the reconstructed object with a clone brush from the original photo.

Does it support multicamera rigs with images of different w x h sizes ?

Yes, it supports. Here, of course, it should be noted that the sensor of imported cameras matters, and this will affect whether the distortion of the lens will be correctly calculated, because Blender strictly separates the height and width of the sensor, but the import operators automatically set its value. But yes, working with different image resolutions is one of the main purposes of the add-on. For example, if a camera with the largest resolution is installed vertically in the facial rig and is aimed directly at the face, it makes sense to add details from it

Can you input intrinsics via file?

Yes, I can and you probably can too I've also written some documentation for the add-on, so I think that might help.

But in general yes, of course you can import/export internal/external parameters of cameras - this significantly reduces the size of the data needed to work with the scan texture.

1.6 Data Cleanup

Setup context stage, during which you can clear existing scene data.

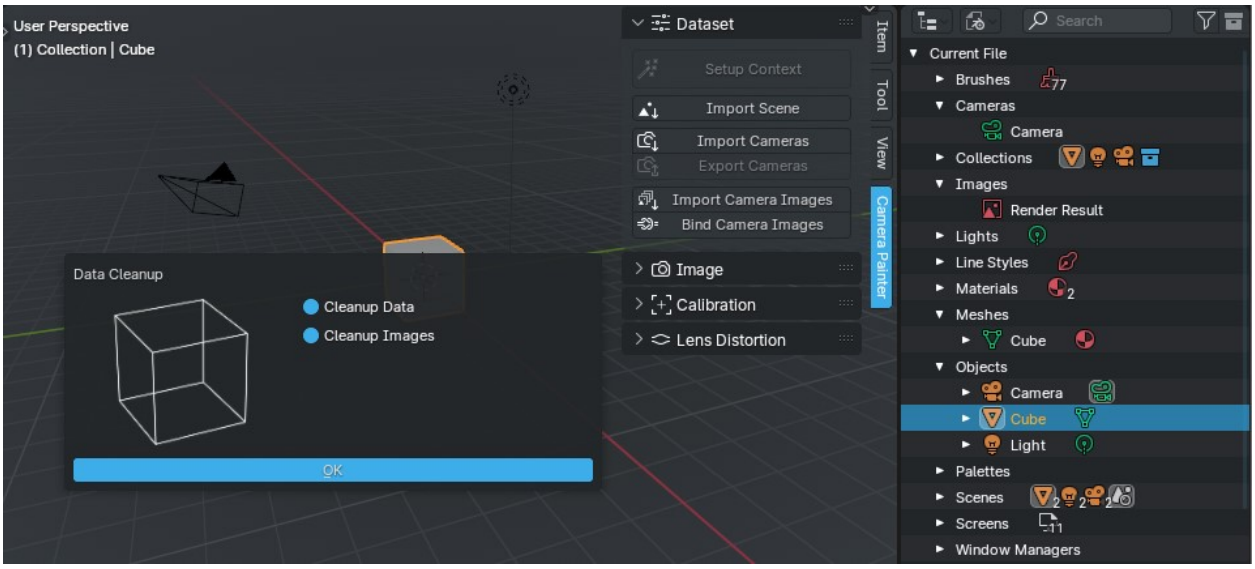
If the file contains:

- Cube
- Material
- Camera
- Lamp

that is, it is a standard scene, then it will be asked to *clean up this data*.

If there is only one "Render Result" image in the file, it will also be prompted to *cleanup images*.

If you want to leave everything as it is, you need to turn off all execution options and simply agree to execution - for example, if you need to complete the settings that were interrupted at one of the following stages.



1.6.1 Cleanup Data

Delete object, mesh, material, texture, light and camera data-blocks

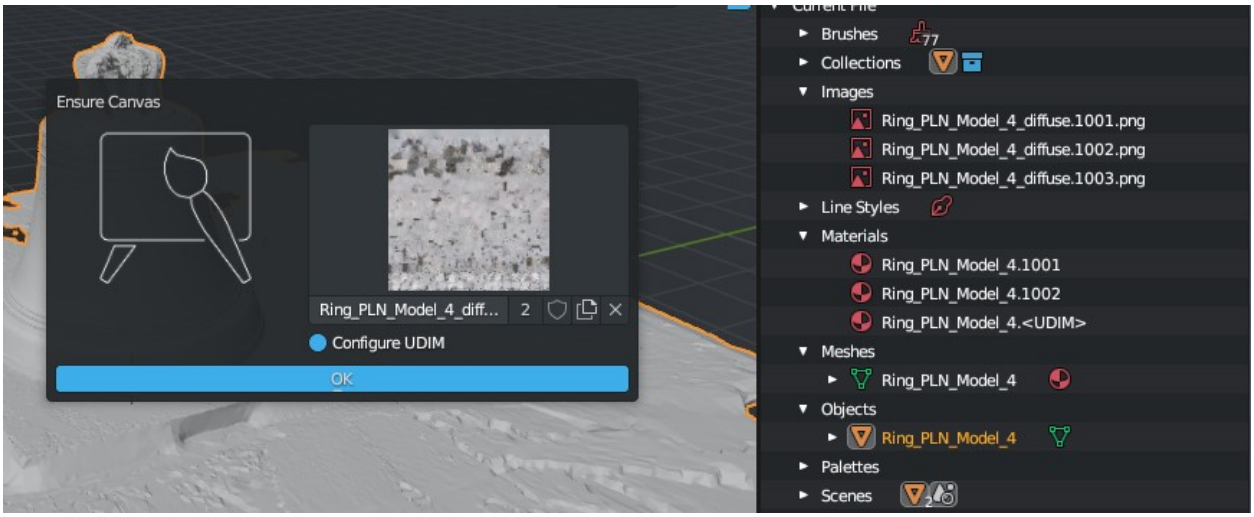
1.6.2 Cleanup Images

Delete all image data-blocks

1.7 Ensure Canvas

Stage of *context setup* during which you need to choose a canvas to paint on.

If you skip it and at the there is no canvas, then the addon work will continue but in a limited mode - obviously there will be nothing to paint on, but you can still work with import-export. If it is necessary to draw on an empty image and it is not previously in the file, the *creation operator* is provided. To quickly select an image, you can use the *quick canvas selection* operator.



1.8 Quick Select Canvas

Available during the *canvas selection stage*.

Often the name of the exported image is similar to the name of the exported object or simply after importing into Blender it is specified as a texture in one of the materials of the imported object.

Therefore, a search is first performed among the materials of the imported object - if the material contains an image texture node, this image will be proposed. Perhaps this is a fairly simple way of searching, but it is the fastest, so it is a priority.

Most often, the first method allows you to find a suitable image, if not - a search for an image with a similar name will be carried out among all those that are currently open. Their number can be quite large, so this method is relatively slow. Here, the name of the object will be used for the search and the possible UDIM pattern of the image will be taken into account.

In any case, the operator is designed to speed up trivial tasks, and you can always choose the image manually. For example, there is an option when almost the desired image was offered, you can confirm a quick selection and then scroll a little through the list of images to the desired one.

1.9 Create New Canvas

Available during the *canvas selection setup stage*. This is a simple operator to create a new canvas with a size of 2048x2048 pixels. It can be used if there is no exported texture.

To be fair, this operator allows to bypass certain limitations of the standard image creation operator, and its use is not mission-critical, but it can still be used in certain non-standard workflows.

1.10 Import Scene

Stage *setting up the context* as well as an independent operator. In fact, it's just a wrapper for standard import operators with presets.

If the presets are not suitable for any reason, you can always import the scene manually and skip this stage when setting up the context - just do not select any file.

So, in order to use this operator, you need to select a file to import and select the default settings for *third-party* files. This setting is saved as addon preference so you can do it once.

Next, the file type will be automatically determined using its extension. Files supported by the current version of the addon:

- `Wavefront (.obj)`, `Collada (.dae)` - used built-in Blender importers.
- `Autodesk (*.fbx)` - the standard FBX file import addon will be used, so it should be enabled.

If these requirements are not met, the files will be hidden in the file manager and refused import. However, this is more information for users who use their own Blender builds.

1.11 Import Cameras

The *setup context stage* and an independent operator for importing extrinsic and *intrinsic* camera parameters.

For import, you need to select one or more files - their type will be determined automatically. Unlike *scene importer*, the file type will be determined not only by the extension - as soon as the files are selected, they will be partially read and checked. The check will be based on the key characteristics of various file types of third-party software - it can be the first line with a file description or simply the number of elements in a line (for character-separated formats). For `XML` files, certain keys will be found in the tree and so on.

Files supported by the current version of the addon:

- **Reality Capture**
  - Metadata (XMP)
  - Internal/External camera parameters
  - Comma-separated Name, X, Y, Z
  - Comma-separated Name, X, Y, Z, Heading, Pitch, Roll
  - Comma-separated Name, X, Y, Z, Omega, Phi, Kappa
- **Agisoft Metashape**
  - Agisoft XML
  - Omega Phi Kappa (.txt)

1.11.1 Forward

Axis forward

X Forward
Use the global X axis as the forward direction

Y Forward
Use the global Y axis as the forward direction

Z Forward
Use the global Z axis as the forward direction

Negative X Forward
Use the negative global X axis as the forward direction

Negative Y Forward
Use the negative global Y axis as the forward direction

Negative Z Forward
Use the negative global Z axis as the forward direction

1.11.2 Up

Axis up

X Up
Use the global X axis as the up direction

Y Up
Use the global Y axis as the up direction

Z Up
Use the global Z axis as the up direction

Negative X Up
Use the negative global X axis as the up direction

Negative Y Up
Use the negative global Y axis as the up direction

Negative Z Up
Use the negative global Z axis as the up direction

1.11.3 Comparison Options

Options for comparing names. At least one of the options must be selected for each compared type (object, image, etc.)

Ignore Letter Case
Ignore character register for matching

Ignore Extensions
Use name only, no file extension when searching

Use Object Name
Use camera object name for comparison

Use Camera Name
Use camera data name for comparison

Use Image Name
Use image data-block name for comparison

Use Image File Name
Use image file name for comparison

1.12 Bind Camera Images

Associates images to cameras using names. Can be used both for all cameras present in the scene, and for the active camera (for the texture drawing mode, this is the scene's active camera, for object mode - the selected camera or the scene camera is also active if an object of a different type is selected).

The search for images first of all always takes place among already open files, then, if no matches are found, among files in the selected directory. For this, the operator in the user interface has two launch modes. Import mode should be used if the required images are not yet open - in this case, it will be possible to choose a directory for searching for files, and if matches are found - the necessary images will be opened and linked to the cameras. Binding mode there is no fundamental difference with the import mode, but there is no directory selection here - if the necessary files have not yet been opened then the directory from which images or camera data were last imported will be used. This mode is useful if, for example, other images were manually selected and need to be linked again by name.

1.12.1 Comparison Options

Options for comparing names. At least one of the options must be selected for each compared type (object, image, etc.)

Ignore Letter Case
Ignore character register for matching

Ignore Extensions
Use name only, no file extension when searching

Use Object Name
Use camera object name for comparison

Use Camera Name
Use camera data name for comparison

Use Image Name
Use image data-block name for comparison

Use Image File Name
Use image file name for comparison

1.13 Ensure Tool Settings

The last stage of setting up the work context (*Setup Context*).

First, the availability of the drawing object will be checked. The active object must be a mesh with at least one polygon and an active UV map. If such an object is not available, the first visible scene object that matches these parameters will be selected. At the end of this check, the following options are possible:

- Object found - execution will continue.
- The object is found, but it has no polygons - a message about this will be reported and execution will be interrupted.

- The object is found, it has at least one polygon, but there is no active UV map - execution will continue, but in a limited mode - such a message will be reported. Obviously, editing the canvas in this case will not work, but this mode of operation can be useful if, for example, you only need to change the parameters of certain cameras and export camera data back to third-party software.

If the execution continues, then the tool and the scene will be set up. This section can be used to understand how you can set up a work context manually:

- The work mode will be changed to texture paint.
- The Clone tool will be selected.
- Texture drawing mode will be changed to "Single Image".
- The use of the clone layer will be enabled.

At the end of the execution, the operator will change the lighting type in all viewers of the current program window to flat - this way of displaying the object is best suited for editing textures.

Next, the main workflow of the add-on will be launched - it is important that there is at least one camera in the scene, even if it is inactive. During work, the main operator will independently ensure that an active camera is set - it is from it that the projection will take place.

1.14 Setup Context

Sequential setting of the context for work - step by step the scene and necessary data will be imported and the parameters of the tools will be configured.

Execution is divided into stages, each of which is a separate call of automation operators. Each of the execution stages can be skipped if it is not needed in the current situation and go to the next one, or cancel its execution and thus complete the context setting at the current stage.

The sequence of execution stages is as follows:

1. *Data Cleanup*
2. *Import Scene*
3. *Ensure Canvas*
    - *Quick Select Canvas*
    - *Create New Canvas*
4. *Import Cameras*
5. *Bind Camera Images*
6. *Ensure Tool Settings*

1.15 Export Cameras

Exports camera data to files of third-party software.

The addon stores camera data with double precision, according to the IEEE-754 standard, so files imported from third-party software can be exported back without loss of precision.

For export, you need to choose the path and name of the file, and for some formats (for example, Reality Capture Metadata (XMP), which writes each camera in a separate file) - a directory. It is also necessary to choose the file format and the name source.

Files supported by the current version of the addon:

- **Reality Capture**
  - Metadata (XMP)
  - Internal/External camera parameters
  - Comma-separated Name, X, Y, Z
  - Comma-separated Name, X, Y, Z, Heading, Pitch, Roll
  - Comma-separated Name, X, Y, Z, Omega, Phi, Kappa
- **Agisoft Metashape**
  - Omega Phi Kappa (.txt)

Regarding file format selection, you can simply select an existing file to overwrite and the export format will be selected automatically, according to the existing file format. It should be noted here that if an existing file is selected, then to export to another file format, it is necessary to remove the selection from the file and clear the file name field, only then it will be possible to choose another format.

As for choosing the source of the camera names in the exported file, it depends on the dataset of the current scene. Most of the time, the standard settings are enough (use the name of the camera object), but in certain non-standard situations, you need to specify it yourself.

1.15.1 Number of Cameras

Write number of cameras into a file for Reality Capture CSV-like file formats

1.15.2 Calibration Groups

Select to export the information on the created calibration groups for Reality Capture (XMP)

1.15.3 Include Editor Options

Export editor states, e.g. enabled/disabled flags for texturing, meshing, and similar for Reality Capture (XMP)

1.15.4 Export Mode

Depending on how much you trust your registration, you can select the following options or you can also choose not to export camera poses for Reality Capture (XMP)

Do Not Export
: Do not export camera poses

Draft
: This will treat poses as an imperfect draft to be optimized in the future. The draft mode functions also as a flight log

Exact
: If you trust the alignment absolutely. By choosing this option, you are saying to the application that poses are precise, but the global position, orientation, and scale is not known

Locked
: This is the same as the exact option with the difference that the camera position and calibration will not be changed, when locked

1.15.5 Calibration Group

By defining a group for Reality Capture (XMP) we state that all images in this group have the same calibration properties, e.g. the same focal length, the same principal point. Use "-1" if you do not want to group the parameters

1.15.6 Distortion Group

By defining a group for Reality Capture (XMP) we state that all images in this group have the same lens properties, e.g. the same lens distortion coefficients. Use "-1" if you do not want to group the parameters

1.15.7 In Texturing

Whether to use an image to create an object texture for Reality Capture (XMP)

1.15.8 In Meshing

Whether to use an image to create the object mesh data for Reality Capture (XMP)

1.15.9 Forward

Axis forward

X Forward
Use the global X axis as the forward direction

Y Forward
Use the global Y axis as the forward direction

Z Forward
Use the global Z axis as the forward direction

Negative X Forward
Use the negative global X axis as the forward direction

Negative Y Forward
Use the negative global Y axis as the forward direction

Negative Z Forward
Use the negative global Z axis as the forward direction

1.15.10 Up

Axis up

X Up
Use the global X axis as the up direction

Y Up
Use the global Y axis as the up direction

Z Up
Use the global Z axis as the up direction

Negative X Up
Use the negative global X axis as the up direction

Negative Y Up
Use the negative global Y axis as the up direction

Negative Z Up
Use the negative global Z axis as the up direction

1.16 Remove Image from Bind History

Removes an image from the *bind history*. You can delete any image, even the current image, because the history of related images does not affect it.

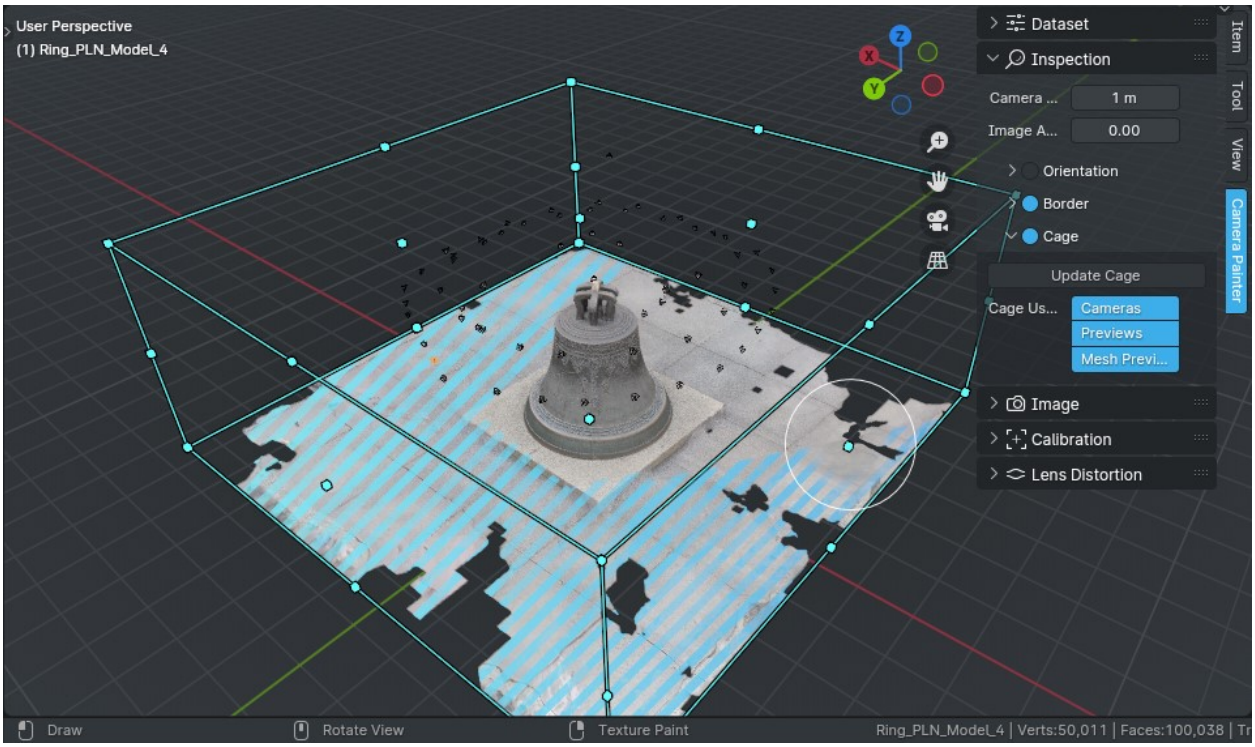
1.17 Show Preferences

Quick access to addon user preferences.

Obviously, the operator itself does not make much sense, since the custom settings are not difficult to reach in the usual way, but during operation it may be necessary to disable or enable some options, for example *Use Mesh Preview* or *Use Previews* will continue to work in this mode. It is also worth noting that in the user settings there is a link to the folder with the addon logs, which can tell a lot about a possible issue, so it will also help to get the current log faster and fix the situation.

1.18 Update Cage

Set the region to work to the current scene bounds. The boundaries of the object and the position of all scene cameras are taken into account. It would be set automatically after *setting context*, but if it has been changed, you can re-calculate the region in this way.



1.19 Preferences

1.19.1 Use Mesh Preview

Use mesh preview for brush and inspection. This option can be useful to significantly reduce the use of video memory, for example, if the purpose of using the addon is only to import and export cameras. The current implementation of Blender forces the display to save the object's data and re-render it, just to show the brush preview. Of course, this is convenient, but it significantly increases the load on the system. Therefore, if it is not critical for you to see a preview of the brush you are drawing with, you can disable the preview

1.19.2 Use Previews

Use image previews. Whether to generate and display image previews in the viewport. This does not significantly increase the amount of video memory, but it can cause small friezes in the viewport. The generation system works according to the principle of displaying the necessary previews. That is, all camera frames in the observer's field of view will be added to the render queue. Of course, the faster you read data from the disk, the faster it will be generated. The camera on which the cursor is hovered on is always added to render queue first

1.19.3 Dithering Strength

8x8 Bayer dithering strength. The resolution of the previews is rather small - 128x128 pixels. Of course, this add-on performs smoothing of the generated previews, but for a clearer understanding of what is shown, you can use dithering and mix it with the original preview

1.19.4 Smooth Previews

Use fast approximated anti-aliasing for preview images. Blender generates previews as Nearest, so readability is lost. This option does not add more data from the original image but performs smoothing and thus makes it more readable

1.19.5 Cameras Transparency

Transparency of cameras in the viewport.

1.19.6 Tooltip Preview Size

The size of the preview image in the tooltip. Does not affect the quality of the displayed preview, only its size

Normal
: Standard size, 128x128 pixels

Large
: Stretched to double size

1.19.7 Tooltip Position

The location of the pop-up tooltip with information about the camera data on which the cursor is hovered

Static
Appears near the brush and remains in the place where it appeared

Floating
Appears and follows the brush

Fixed
Appears only at the bottom of the screen, in the middle

1.19.8 Software

Preset for software for which the operation will be performed

Reality Capture
Use presets for Reality Capture

Agisoft Metashape
Use presets for Agisoft Metashape

1.19.9 Select Framebuffer Scale

Select framebuffer scale percentage, available in "Developer Extras" section. This exists mostly for optimization purposes, since the smaller size of framebuffer significantly reduces the use of both memory and CPU. The point is that the data from framebuffer will be read not only in one particular place, but also to determine which cameras in the field of view of the observer and require rendering of preview. Changing this option requires restarting the active main operator

1.19.10 Log Level

The level of the log that will be output to the console. For log to file, this level value will not change

Debug
Debug messages (low priority)

Info
Informational messages

Warning
Warning messages (medium priority)

Error
Error messages (high priority)

Critical
Critical error messages

1.20 Window Manager

1.20.1 Configure UDIM

The add-on uses some standard methods for importing, but in certain situations this creates incorrect data for the add-on to work with. If this option is activated, after importing the object, the materials for working with UDIM textures will be set, and after selecting the canvas, the image settings will be checked and adjusted

1.21 Scene

1.21.1 Unit

Display units of camera parameters

Millimeters
: Camera parameters in millimeters

Pixels
: Camera parameters relative to image sizes in pixels

1.21.2 Highlight Cameras Orientation

Use a different color for cameras whose images have landscape (including square) and portrait orientation

1.21.3 Cameras Landscape Orientation Color

The color of the display of cameras in which the images have a landscape (including square) orientation

1.21.4 Cameras Portrait Orientation Color

The color of the display of cameras in which the images have a portrait orientation

1.21.5 Highlight Projected Border

Whether to highlight the borders of the frame of the projected image on the object

1.21.6 Highlight Projected Border First Color

The first color that will be used to highlight the parts of the object that are outside of camera frustum

1.21.7 Highlight Projected Border Second Color

The second color that will be used to highlight the parts of the object that are outside of camera frustum

1.21.8 Highlight Projected Border Type

The fill type of the highlighted area

Fill Color
Single color outline

Checker
Checker pattern outline

Lines
Lines pattern outline

1.21.9 Highlight Projected Border Facing

Which side of the polygons that are outside the frame of the projected image to display. an be useful for a scene where there is a closed space, for example a room or a street. At least one of the options should be selected

Front
Show the front of polygons

Back
Show the back of polygons

1.21.10 Viewport Cameras Size

The size of the cameras in the viewport

1.21.11 Current Image Viewport Transparency

Transparency of the image when viewed from the camera

1.21.12 Cage

Allows you to distinguish a certain region for work. Useful for large scenes with lots of cameras

1.21.13 Cage Usage

In which way the region for work affects the scene display

Cameras
Hides the cameras

Previews
Stops rendering of previews outside the region

Mesh Preview
Hides the projection of the object outside the region

1.22 Camera

1.22.1 Image

Camera binded image. It will be used as a paint brush. Also, its size affects the distortion parameters in the pixel coordinate system (in which way - described in more detail for the corresponding parameters), and therefore - also the possibility of exporting to certain file formats

1.22.2 Millimeters Focal Length

Focal length of the camera in millimeters. The value is stored with double precision

1.22.3 Pixels Focal Length

The focal length of the camera in pixels. Calculated by the formula:

$Pixels\ Focal\ Lens = F * P / S$

where F is the focal length in millimeters, P is the larger side of the image, S is the size of the camera sensor in millimeters

1.22.4 Sensor

The size of the camera sensor in millimeters, taking into account the larger side of the attached image. This means that if the sensor fit option is set to horizontal - the width of the sensor will be used for calculations, for vertical fit type - height accordingly. These values can be set manually. But for when working with photogrammetric scenes, it is necessary to use the automatic type of sensor adjustment. In this case, if the image has a landscape orientation (or the sides are equal), the width of the sensor will be used, and in the case of a portrait - the height of the sensor. As you can see, there are opportunities for using non-standard workflows, but the main mode of operation is the automatic sensor type. Also note the camera-bound image, this is important for importing and exporting to some file formats. The value is stored with double precision

1.22.5 Millimeters Skew

The horizontal skew of the image in millimeters relative to its center, excluding principal deviation. The value is stored with double precision

1.22.6 Pixels Skew

Horizontal skew of the image in pixels relative to its center without taking into account deviation. Calculated by the formula:

$Pixels\ Skew = S * P$

where S is the skew value in millimeters, P is the size of the larger side of the image

1.22.7 Aspect Ratio

Image aspect ratio correction factor. It means the ratio of the height of the image to its width - that is, values greater than 1.0 will stretch it vertically, smaller values will compress it. The value is stored with double precision

1.22.8 Millimeters Principal X

The deviation from the center of the image in millimeters along the X axis. Calculated by the formula:

*Millimeters Principal X = Px * S*

where `Px' is the deviation factor, `S' is the size of the camera sensor

1.22.9 Millimeters Principal Y

The deviation from the center of the image in millimeters along the Y axis. Calculated by the formula:

*Millimeters Principal Y = Py * S*

where `Py' is the deviation factor, `S' is the size of the camera sensor

1.22.10 Pixels Principal

The deviation from the center of the image in pixels. Calculated by the formula:

*Pixels Principal = Pxy * L + (Sxy / 2)*

where *Pxy* is the deviation factor, *L* is the larger side of the image, *Sxy* is the image size

1.22.11 Distortion Model

Mathematical model of lens distortion. Note that distortion coefficients may be inconsistent between different distortion models

None
: No distortion, only correction

Division
: The division model with two radial distortion coefficients. This is the simplest mathematical model of linear division of image coordinates, which can be used if the distortion is small, for example, for scenes where the shooting took place close to the object

Polynomial
: A polynomial model with four radial and two tangential distortion coefficients. It uses polynomial functions instead of simple linear division, so it is a more accurate and flexible model that can be used for complex scenes with significant lens distortions

Brown-Conrady
: Brown-Conrady polynomial model with four with four radial and two tangential distortion coefficients. It is the most accurate and flexible model that can be used for complex scenes with significant lens distortions. In general, it is used when a simple linear division model is no longer sufficient

1.22.12 K1

Represents linear radial distortion. It corrects or introduces distortion that increases linearly with radial distance. The value is stored with double precision

1.22.13 K2

Represents cubic radial distortion. It corrects or introduces distortion that increases with the cube of the radial distance. The value is stored with double precision

1.22.14 K3

Represents quintic (fifth-order) radial distortion. It corrects or introduces distortion that increases with the fifth power of the radial distance. The value is stored with double precision

1.22.15 K4

Represents seventh-order radial distortion. It corrects or introduces distortion that increases with the seventh power of the radial distance. The value is stored with double precision

1.22.16 P1

Represents linear tangential distortion. Corrects or introduces distortion that increases linearly with the distance from the image center. The value is stored with double precision

1.22.17 P2

Represents quadratic tangential distortion. Corrects or introduces distortion that increases with the square of the distance from the image center. The value is stored with double precision

1.22.18 Bind History

Images that were previously binded to this camera. The option is used if, for example, it is necessary to draw some area from another image, and then return to the previous one. Of course, this is not a standard workflow, but it is used sometimes

Image

The image that was previously attached to this camera