Kintsugi 3D Builder

User documentation – version 1.4.0

Acknowledgements

Kintsugi 3D was developed with support from a grant from the National Endowment for the Humanities (NEH PR-290101-23).

Additional student research support came from a Minority Student Research grant from the National Science Foundation through the Wisconsin Louis Stokes Alliance for Minority Participation (WiscAMP) program.

Thanks to Carla Schroer and Mark Mudge from Cultural Heritage Imaging, Charles Walbridge, Lou Brown, Pierre Ware, and Dan Dennehy from the Minneapolis Institute of Art (Mia), Darcy Hannen (UW-Stout MFA candidate), Seth Berrier (UW-Stout faculty), Michael Ludwig, Sam Steinkamp, and Gary Meyer (University of Minnesota) for their contributions to this project.

UW-Stout undergraduate students who have contributed to this project include: Mey Amor, Ian Anderson, Tyler Betanski, Kyle Boatwright, Jacob Buelow, Atlas Collins, Zoe Cuthrell, Luke Denney, Elliot Duffy, Augusto Freitas, Chloe Ferguson, Tyler Garcia, Kyle Boatwright, Melissa Kosharek, Josh Lyu, Victor Mondragon, Isabel Smith, Blane Suess, Isaac Tesch, Alexis Williams, and Nathaniel Willius.

Alex Kautz contributed to this project as part of an REU at the University of Minnesota.

The Kintsugi 3D logo, color scheme and font selections were designed by Isabel Smith.

The “Kintsugi” name was originally suggested by Luke Denney.

Contents

Overview

Developed by a team at the University of Wisconsin – Stout with support from the Minneapolis Institute of Art (Mia) and Cultural Heritage Imaging (CHI) under a grant from the National Endowment for the Humanities (NEH), Kintsugi 3D[1] [2] is a novel software platform for synthesizing empirically-based roughness, specularity, normal, and diffuse textures from an image set containing photographs of an object captured with flash-on-camera illumination. What sets Kintsugi 3D apart from other workflows that produce specularity maps is that all the textures produced are empirically-based: they are derived directly from photographic data using classical optimization methods, and the reconstruction error from this optimization process can be recorded as metadata for the object as documentation of the fidelity of the digitized form.

Kintsugi 3D is an evolution of its predecessor, IBRelight[3]: a tool developed as part of Michael Tetzlaff's (Kintsugi 3D project lead) doctoral thesis that originally had a use case of being a tool for cultural heritage professionals to generate images and videos from photogrammetric models by reprojecting the original photographs onto the 3D model. What was unique about IBRelight was its use of a flash-on-camera photography technique that made it possible to change the lighting in software based on which flash images were selected for blending. However, IBRelight had substantial hardware requirements which were a barrier preventing it from being used for general-purpose dissemination of digitized heritage objects. By implementing a texture processing technique originally described by Nam et al.[4] and refined by Tetzlaff,[5] IBRelight has evolved into Kintsugi 3D Builder, with a user experience redesigned to specifically target the application of building textures and materials for use in lightweight 3D viewer applications.

The Kintsugi 3D platform features its own Viewer application[6] for public access to finished digitizations in the highest possible quality, using a custom shader designed specifically for materials derived from photographs. The goal of this viewer is to support the rest of the Kintsugi 3D platform with a lightweight app for public access to this robust reproduction quality, while striving for feature parity with comparable viewers such as Sketchfab or Smithsonian Voyager.

---

[1] https://github.com/michaelt919/Kintsugi3DBuilder

[2] Brown, L.; Walbridge, C.; Tetzlaff, M. "Kintsugi 3D: An Empirically-Based Photogrammetry Production Pipeline." IS&T Archiving Conference. 2024, 76-80.

[3] Tetzlaff, M.; Meyer, G.; Kautz, A. "IBRelight: An Image-Based 3D Renderer for Cultural Heritage." *IS&T Archiving Conference.* 2018, 93-98.

[4] Nam, G., Lee, J. H., Gutierrez, D., and Kim, M. H. "Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography." *ACM Transatctions on Graphics,* vol. 37, no. 6. 2018, pp. 267:1-267:12. https://dl.acm.org/doi/abs/10.1145/3272127.3275017

[5] Tetzlaff, M. "High-Fidelity Specular SVBRDF Acquisition from Flash Photographs." *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 30, no. 4. 2024, pp. 1885-1896. https://ieeexplore.ieee.org/abstract/document/10012127

[6] https://github.com/UWStout/Kintsugi3DViewer

Kintsugi 3D also supports exporting in standard texture formats to support existing efforts using established viewers like Sketchfab or Voyager. The simplicity and open access of the Kintsugi 3D platform makes this available even to institutions without the infrastructure or support to otherwise develop such hands-on physical or digital interactive experiences.

Kintsugi 3D still relies on Agisoft Metashape or Reality Capture for camera alignment and 3D reconstruction; it merely replaces the final stage of texture generation. As such, it is an extension, not a replacement, for established photogrammetry solutions. However, Kintsugi 3D does change the photogrammetry pipeline in certain significant ways. Professional photographers at many institutions currently capture image sets that utilize white backgrounds with uniform lighting on the object in each image set. While this makes it possible to easily mask images from the contrast between the background and object, it also makes achieving the necessary uniform illumination for accurate textures more challenging, and empirically deriving specularity from such images is not possible. There is also a risk of color issues in the textures due to bounce lighting or interreflections that, among other things, dull out colors, reducing the texture fidelity.

In contrast, Kintsugi 3D, like its predecessor IBRelight, uses a photographic technique that leverages a flash mounted on the camera. There are two primary modes of capture: against a black background in a studio environment, or in-gallery. These two options reduce unintentional bounce light and offer accessibility and flexibility in terms of how the photos are taken, while providing essential reflectivity data for Kintsugi 3D to reconstruct specular maps, which most other photogrammetry workflows cannot replicate.

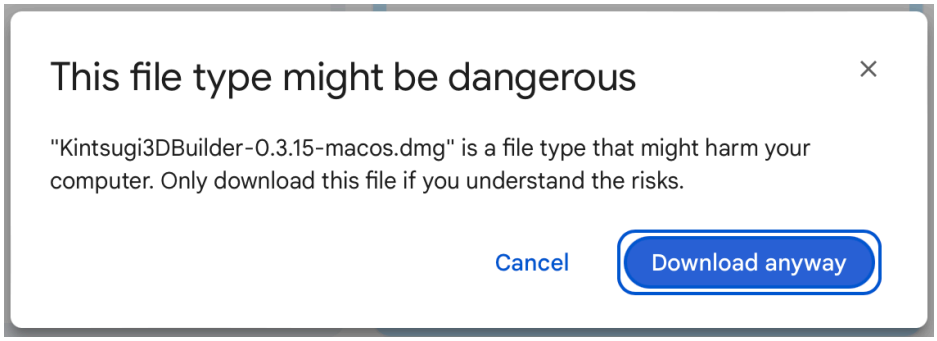
Platforms

Kintsugi 3D Builder is a downloadable standalone application that is available as an executable with installers for Windows 10+ and MacOS.

The software is available as open source code, and can be built for other platforms by interested users.

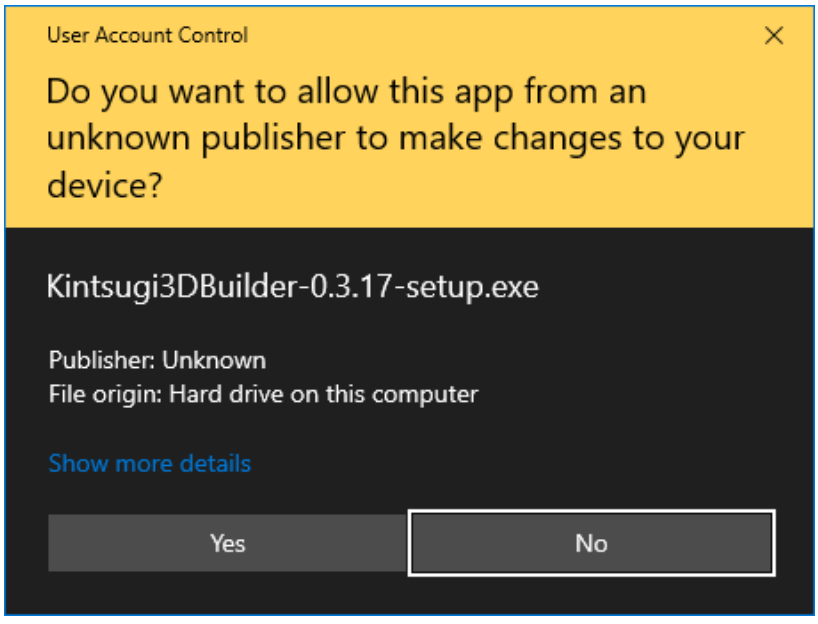
Installation

Kintsugi 3D is an open-source software project that is currently in “beta.” Because of this, many browsers and operating systems flag it with a false positive as a potential virus. You may see a window like the following when downloading (exact details will vary by operating system and web browser):

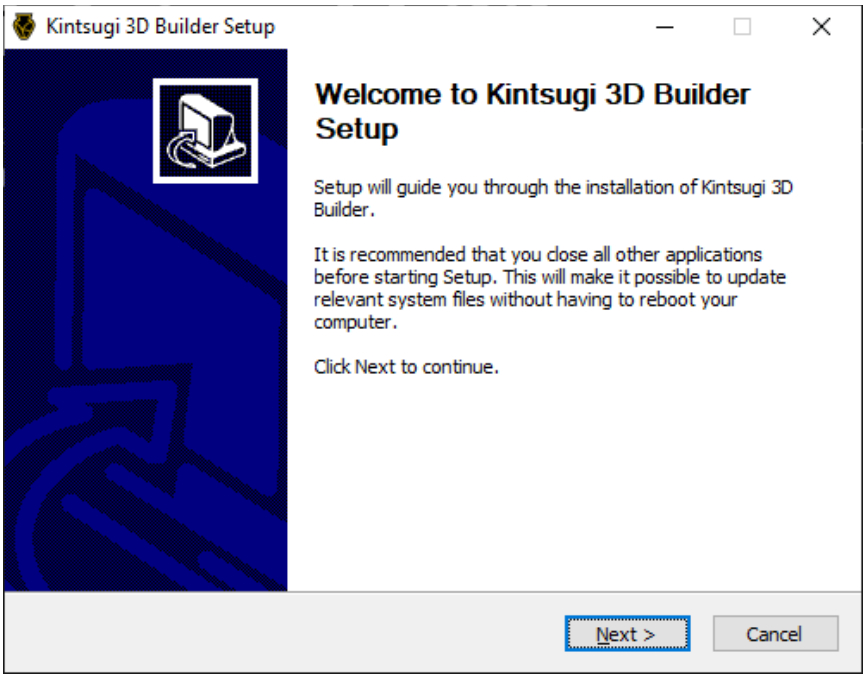


Windows

On Windows, after downloading, double-click the downloaded setup file to run the installer. You will see a prompt like the following:

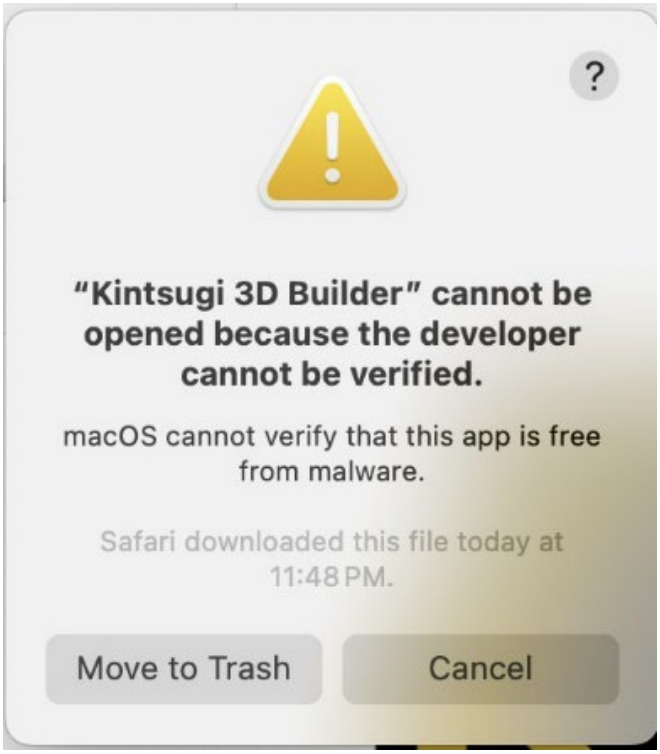


After clicking “Yes”, the setup wizard will run. Click through to install Kintsugi 3D.

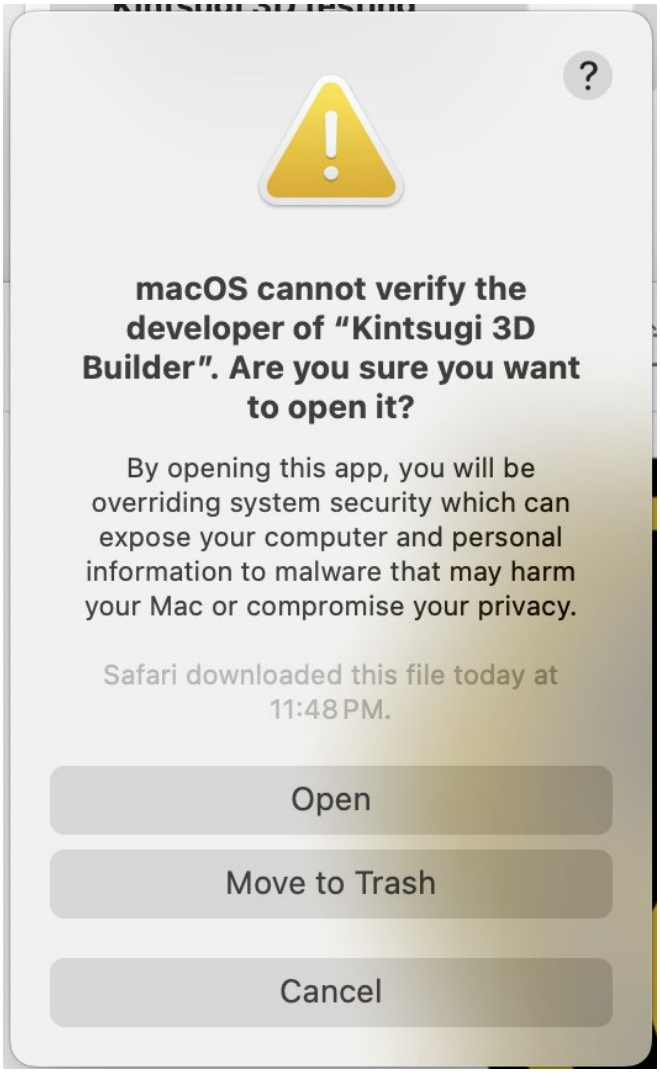


Mac OS

On Mac OS, the downloaded DMG file will contain packaged apps for Kintsugi 3D Builder and Kintsugi 3D Viewer, which can be copied into your Applications folder using “drag and drop.” Because the software is currently unsigned, for the time being on MacOS, you will need to bypass Gatekeeper in order to run the apps. If you simply try to run the downloaded apps by “double-clicking”, you will see a message like the following:



To fix this issue, find Kintsugi 3D Builder in Applications folder, right click and choose “Open.” The following dialog should appear:



Choose “Open” to bypass Gatekeeper and launch Kintsugi 3D Builder.

Photography guidelines

Kintsugi 3D works on 3D models that were captured with a specific lighting approach (described below) and that were processed using Agisoft Metashape or Reality Capture.

Photogrammetry software requirements

Either Reality Capture or Agisoft Metashape 2.x are fully supported. Agisoft Metashape 1.x projects may work but are not officially supported and may stop working with later versions of Kintsugi 3D.

Basic photogrammetric capture requirements

The primary requirement for Kintsugi 3D is the illumination of the subject. Other aspects of acquiring images for use in photogrammetry remain unchanged.

Basic photography requirements

Kintsugi 3D expects that the object is photographed under a very specific kind of illumination - an on-axis / camera-mounted light source. A common example of this is the built-in flash that is available on many digital SLR cameras. It is best for this light source be as small as possible so that it is very close to the underlying assumption, built into Kintsugi 3D, that the light source is infinitely small. It is also important that the light provided by the flash is significantly brighter than any other ambient illumination in the photography environment – ideally, there would be basically no ambient light (a pitch-dark room when the flash is off). Good photographs for Kintsugi 3D Builder will exhibit strong highlights, and will probably look underexposed compared to what you're used to for photogrammetry.

FAQ

- **Won't Metashape / Reality Capture have trouble aligning the images if they are dominated by a flash highlight?** In theory the use of flash images could reduce the quality of the camera alignment and mesh reconstruction. With large datasets (200-500 images) and careful adherence to other best practices, we have found that Metashape still produces reasonable results. However, see "other considerations for photogrammetry" below for suggestions on how traditional photography can complement the camera-mounted flash to improve confidence in the reconstruction.
- **Will the highlights be baked into the texture map?**
  Kintsugi 3D Builder is intended to replace the texture map produced by Metashape **/** Reality Capture, and fits a specular model that accounts for highlights. As such, if all goes well, the highlights in the photographs should not appear in Kintsugi's diffuse / albedo texture map, but will determine the shape of the fitted specular material.
- **The images look too dark. Is this a problem?**
  With Kintsugi 3D Builder, it is important to expose images to minimize clipping of the specular highlight (see advanced guidelines below) in order to capture the full dynamic range of the highlight. However, this requires images to look "underexposed" as typically understood by experienced photographers. This may raise several concerns:

- **Will Metashape / Reality Capture have trouble aligning cameras and reconstructing the model from underexposed images?**
  In practice, this seems to generally not be an issue for many datasets. However, if the image exposure does turn out to be problematic, a conceivable workflow would be to tonemap the images at two different levels and use the brighter version for 3D reconstruction and the darker one for Kintsugi 3D Builder.
- **Is it a good idea to use ambient light in addition to flash to brighten up the darker parts of the photographs?**
  Kintsugi 3D Builder assumes that there is no ambient light in the scene. Because of this, if the ambient light level is high enough to significantly affect the appearance of the object, the fitted specular model will be inaccurate. A better alternative is to use two sets of images – one with flash on in otherwise pitch darkness, and the other with no flash but with ambient light on – and use them all together in Metashape or Reality Capture to build the model. The non-flash images can then be disabled prior to loading the project into Kintsugi 3D Builder.
- **Will the textures produced be darker if the photographs are underexposed?**
  With the use of an X-Rite Color Checker chart, Kintsugi 3D Builder can calibrate the textures to contain true albedo – representing the percentage of incident light reflected. As such, even if the photographs are underexposed, provided that the Color Checker was photographed under the same conditions, Kintsugi 3D will account for this in the math when building the textures and the result should not be any darker.
- **Will the textures be grainy / noisy if compensating for underexposed photographs?**
  In the extreme, this is possible. However, because Kintsugi 3D blends / averages many photographs to build the textures, the "exposure" of the textures as far as signal-to-noise ratio is concerned scales with the ***sum*** of the exposures over ***all*** the images that are visible to each pixel. As such, you actually have more exposure than you think you do looking at an individual image.

- **How shiny is too shiny?**
  Kintsugi 3D Builder is designed to reconstruct specular materials ranging from very rough objects with just the slightest hint of gloss to nearly mirror-like metal surfaces. However, photogrammetric reconstruction in Metashape or Reality Capture may break down as surfaces become mirror like and it starts to pick up features in the reflections. The same is true for extreme translucency / transparency. That said, with careful adherence to photogrammetry best practices and some of the "other photogrammetry considerations" listed below, we have seen Metashape successfully reconstruct bronze objects with strong specular highlights as well as jade artifacts exhibiting significant translucency (subsurface scattering). Determining the precise limits of this and finding solutions to extend the boundaries of what can be reconstructed is an area of open research.

Advanced guidelines for Kintsugi 3D

It is best to not use automatic camera settings such as aperture, shutter speed, and ISO, when taking photographs for photogrammetry. Avoiding automatic camera settings (including flash brightness in addition to the settings listed above) is also important when using Kintsugi 3D. These settings all affect the effective exposure of the photographs and, when set to "automatic mode," the exposure may change between photographs. Kintsugi 3D makes an implicit assumption that all images have the same exposure, and if this is not the case, there may be unexpected changes in brightness in the rendering produced by Kintsugi 3D.

Ideally, the flash would be a "point light" that only has "natural," inverse-square falloff as the subject gets further away from the camera. In practice, most flashes will have a limited cone of illumination, so the goal is to keep the subject within that cone and try to use a flash that provides lighting that is as uniform as possible across the cone until it starts to fall off near the edge of the cone.

The depth of field should be set to be as large as is required by the shape of the imaging subject, or, if this is not optically practical, as large as possible: to increase the depth of field, the aperture can be decreased so long as is does not cause diffraction blurriness, and the camera can also be moved away from the subject so long as the subject is still a reasonable size in the field of view. Sharp images also make it easier for the photogrammetry software to process the images, and also improve the quality of the textures produced by Kintsugi 3D.

It is also important that the light provided by the flash is significantly brighter than any other ambient illumination in the photography environment. If this is not the case, ambient light will introduce inaccuracies when the object is relit. Here is one way to check if the ambient light level is a problem:

- First, several test images should be taken with the flash on to set the proper exposure level.
- Then, the flash should be turned off, and another picture should be taken at the same exposure levels as in the first step.
- If the photograph taken with the flash off is essentially black with the object not visible at all, that means that the flash is bright enough.
- If the object is still visible, then the amount of ambient light either needs to be reduced, or the flash needs to be brighter.
- If possible, it is useful to use a black background and place the object on a black surface so that there is minimal ambient light resulting from scattering of the light of the flash itself.
- It is also important to make sure that there are not interreflections from the flash off of mirror-like surfaces in the room that may result in additional, unwanted illumination.

It is important to understand when using on-axis flash photography for Kintsugi 3D that a desirable exposure level is often different than it would be for traditional photography. In other circumstances, a photographer would properly expose the non-specular parts of an object, with some "clipping" of the specular highlights to white being expected. The goal, when taking photographs for Kintsugi 3D, is to accurately measure the specular highlights, which means that

any clipping of the specular highlights must be minimal. This will mean that the photographs taken may appear "underexposed" to the eye of the experienced photographer, but this underexposure is necessary for Kintsugi 3D to capture the required specular information. On the other hand, the underexposed parts of the images will usually correspond to diffuse reflectance that can still be determined with an acceptable amount of precision by combining all of the available photographs.

All of these recommendations require a somewhat subjective evaluation of the brightness and quality of an image. Because this may be difficult to discern on the small screen on many digital SLR cameras, it may be best to look at some preliminary photographs on a higher quality computer monitor and make sure the focus and exposure are set correctly, before taking a full set of photographs.

Color calibration charts, which are useful for color balancing in general-purpose photography, are even more valuable when taking photographs for Kintsugi 3D. Kintsugi 3D is able to use a photograph taken with the MacBeth ColorChecker chart (sold by Calibrite)[7] directly in front of the object to obtain the exposure and tonemapping information that is necessary to convert the pixel values in the photographs into absolute reflectance measurements.

Kintsugi 3D currently assumes that in this photograph, the distance from the camera to the calibration chart is approximately the same as the distance from the camera to the closest point on the object for some calibrated camera in the photogrammetry project. To satisfy these assumptions, the calibration chart should be placed either right in front of the object (as close as possible without touching the object) or at least in a location that is the same distance from the camera as the front of the object. If the photograph containing the calibration chart is not usable for photogrammetry (for instance, if the chart occludes part of the object), a similar photograph should be taken without the chart but from the same camera position and orientation, without changing the flash power, exposure settings, or the distance from the camera to the object, and that photograph should be included and calibrated as part of the photogrammetry project.

The image on the right shows an example of what a good color calibration reference photograph for Kintsugi 3D looks like.

It is also best to shoot images originally in RAW format rather than a tonemapped format such as JPEG, PNG, or TIFF. One benefit of doing this is that it makes it possible to adjust the exposure of the images after the photography session. It also allows for more control when

[7] https://calibrite.com/us/product/colorchecker-passport-photo-2/

tonemapping the images to a format like JPEG, PNG, or TIFF, resulting in a more accurate interpretation of reflectance, even when using a Color Checker chart.

Other considerations for photogrammetry

In addition to being suitable for eventual use in Kintsugi 3D, it is important that the photographs taken are also usable by photogrammetry software in order to reconstruct a 3D geometric model of the object. Unfortunately, many of the constraints just described - the exposure and lighting restrictions in particular - may in fact make it more difficult to reconstruct an object using photogrammetry software like Agisoft Metashape or Reality Capture. However, there are a few ways around this issue.

One simple practice is to use different tonemapping settings for Metashape / Reality Capture and Kintsugi 3D. That is, if the images have all been stored in a RAW format, they can be tonemapped at a higher exposure for 3D reconstruction in photogrammetry software and again at a lower exposure for use in Kintsugi 3D.

It may be helpful to also take a supplementary set of photographs using the best practices for traditional photogrammetry – in other words, bright ambient illumination, a white background and resting surface (if possible), and a higher exposure setting. This set of photographs supplements the on-axis flash photographs for the purpose of photogrammetry, but should not be used for rendering in Kintsugi 3D, and they do not replace the flash photographs. Metashape is known to be able to handle both kinds of photographs simultaneously, and the processing it performs actually benefits from the presence of the additional photographs that include ambient illumination.

One should also ensure that the object has a clearly visible silhouette and a sufficient number of distinct features that are not ambiguous due to symmetry. If symmetry or homogeneity (the lack of recognizable features) are an issue, it may be useful to place another object or even a printed image next to or underneath the object to provide features that the photogrammetry software can use to align the images.

Finally, if the object is being moved in each image with a fixed background, for example, when using a turntable to rotate the object, it is useful to take a picture of the background so that it can be used to generate a “mask” that will prevent the photogrammetry software from matching features in the background rather than features on the object. Or, if using Metashape, employ the “exclude stationary tie-points” alignment feature.

Quick start

There are two ways to import a project into Kintsugi 3D Builder: by importing a Metashape project (.psx) or using loose files. Reality Capture datasets may only be imported as loose files.

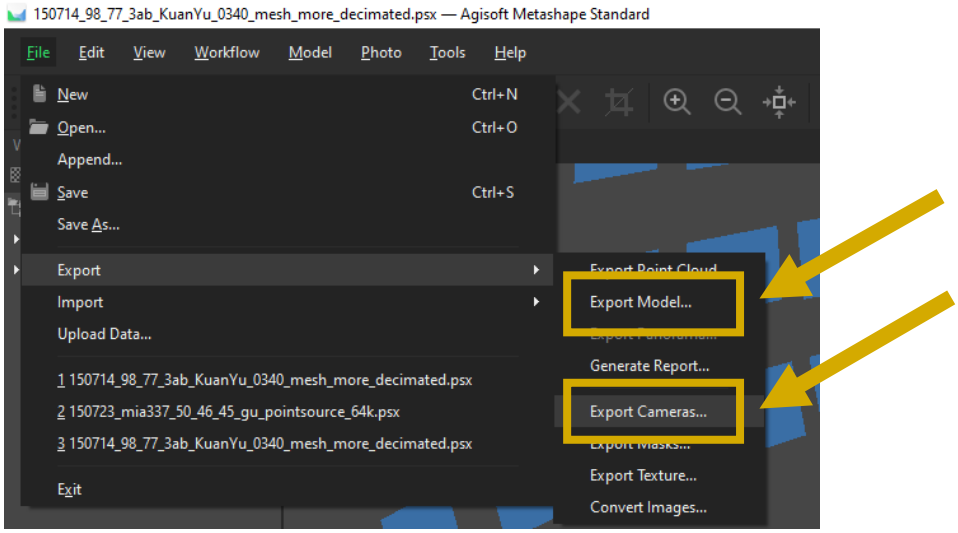
1. Import Metashape project

To prepare a Metashape project to be imported into Kintsugi 3D Builder, make sure the following requirements and guidelines are met, then skip ahead to the section titled "Import into Kintsugi 3D Builder."

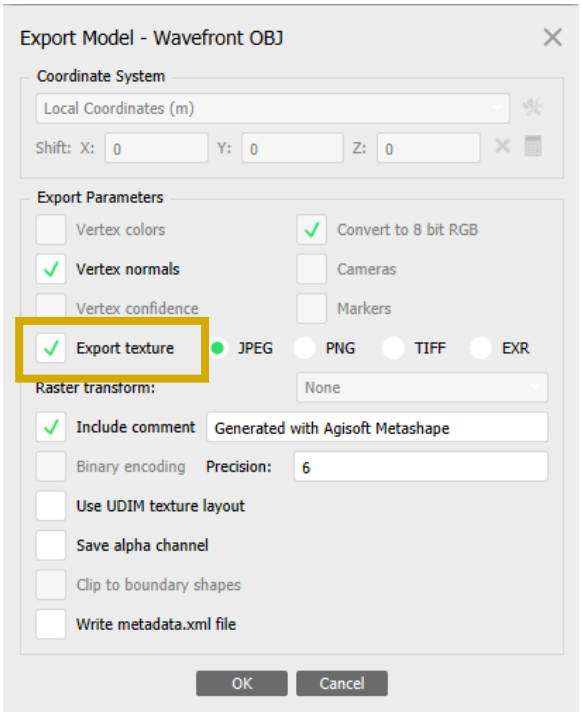
- Make sure you have finished all the processing steps, so that you have a complete model of the object's geometry.
- Make sure that a texture has been generated after performing any smoothing or decimation. Note that the texture itself will not be used by Kintsugi 3D Builder for the most part, but the process of generating a texture creates texture coordinates that are required for processing textures in Kintsugi 3D Builder.
- If you have multiple chunks, you will only be able to export and use the cameras from one of them, so make sure that you have all the photos you need in the same chunk. One option is to combine or copy the cameras into one chunk prior to importing.
- If you used a model transformation, make sure that you reset it using: Model > Transform Object > Reset Transform. Although Kintsugi 3D Builder may be able to interpret model transformations correctly in some circumstances, model transformations are known to cause problems, so we recommend this step for now.
- Disable any cameras corresponding to non-flash photographs, as well as any other photographs that you don't want to be used for image-based rendering for any reason.
- If you used higher exposure versions of the images for Metashape processing, now is the time to replace them with lower exposure versions that will be used by Kintsugi 3D Builder. It may be simplest to just create a copy of the directory containing your images and Metashape project and then write over the old high-exposure images with the low-exposure versions.

2. Export individual files from Agisoft Metashape

Sometimes the automated import from a Metashape project may not be successful, or more control of the files being imported is required. You can export the data required by Kintsugi 3D Builder by following the steps that are described in this section. Kintsugi 3D requires two files to be exported from Metashape: the 3D model ("Export > Export Model…") and the camera calibration data ("Export > Export Cameras…"):



For the 3D model, it is important that the “Export texture” checkbox is enabled; otherwise, the model will not be usable in Kintsugi 3D Builder. The Wavefront OBJ and PLY formats are currently supported for import.

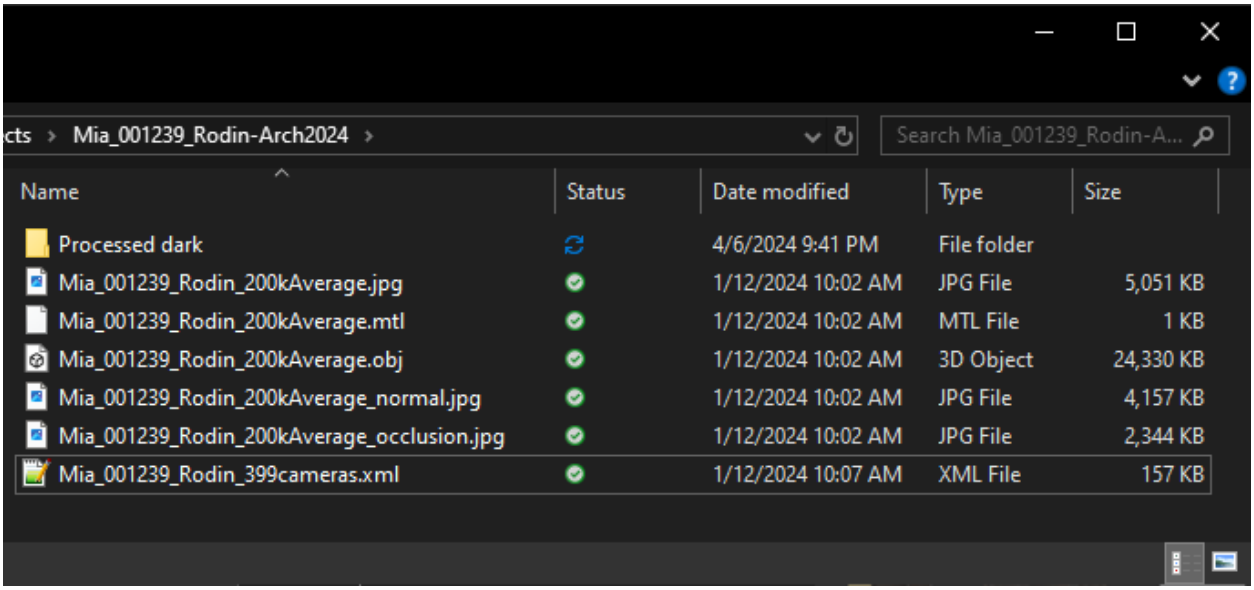


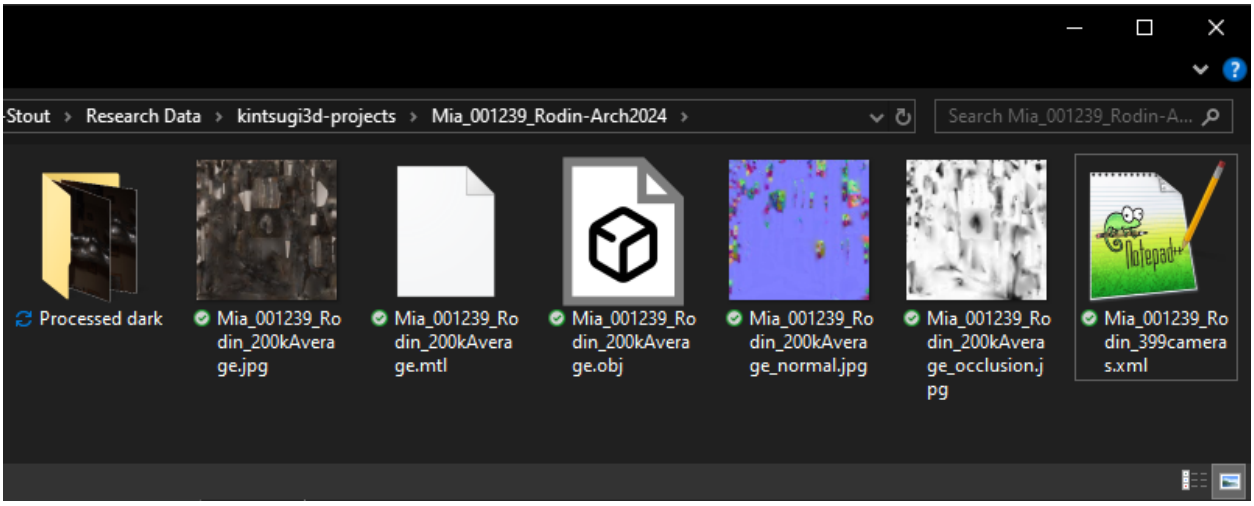
If you have multiple quality levels for your model (i.e. different poly counts), it is important to export the version that you want to ultimately distribute with the textures from Kintsugi 3D Builder. In general, textures are not transferrable between different meshes of the same object.

You may optionally want to generate ambient occlusion (AO) maps in Metashape. Kintsugi 3D Builder does not support generating AO maps, but can pack an AO map imported from Metashape into an “ORM” (occlusion / roughness / metallicity) texture when processing other

textures, and reference that texture in an exported a glTF model. If you generate an AO map in Metashape, make sure you do it after any decimation, remeshing, and/or UV editing.

After exporting the cameras and the 3D model, your data folder should look like the following, containing the cameras XML file, the 3D model (mesh), and the original source photographs:





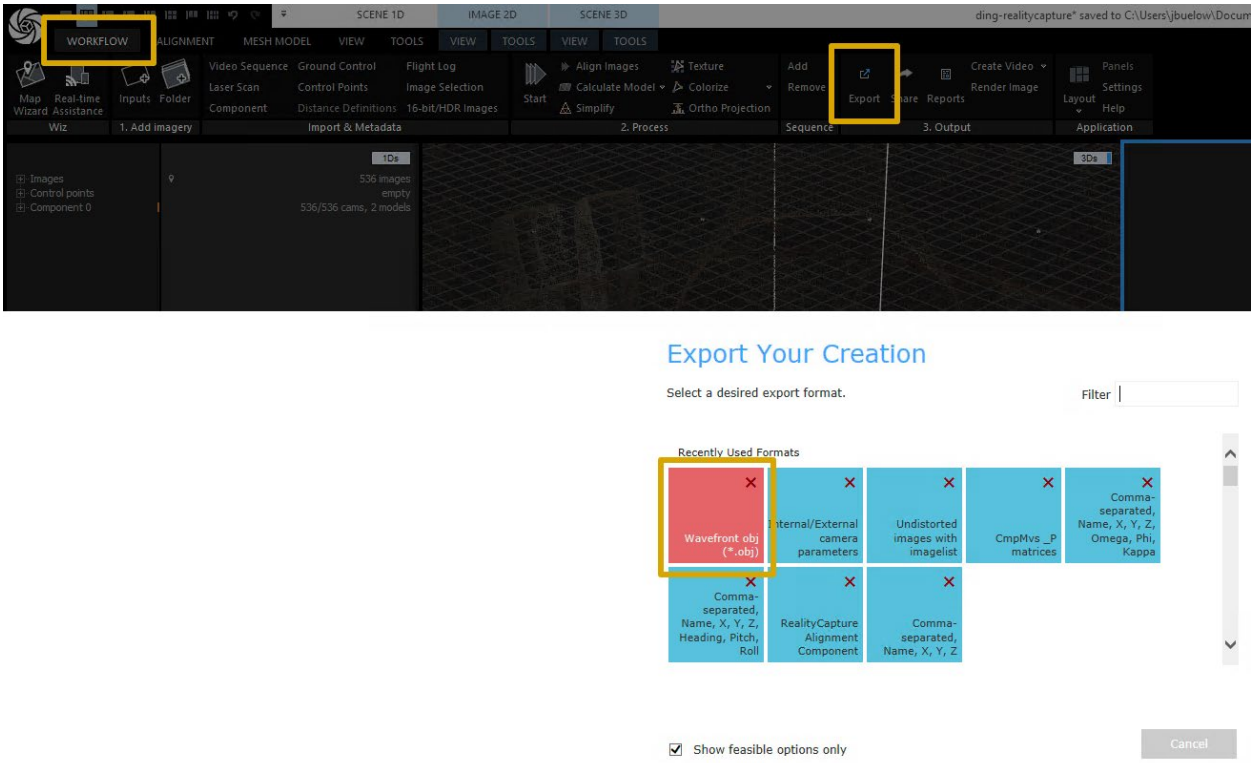
When using Wavefront OBJ, you will also find a material (.mtl) file. You will also see the color texture (not used by Kintsugi 3D but required for Metashape to export UVs) along with a normal map and occlusion maps which may optionally be imported into Kintsugi 3D Builder. If provided, Kintsugi 3D Builder will pack the occlusion map into an “ORM” map after processing textures, and the normal map may be used as a starting point for normal map refinement when processing textures.

3. Export individual files from Reality Capture

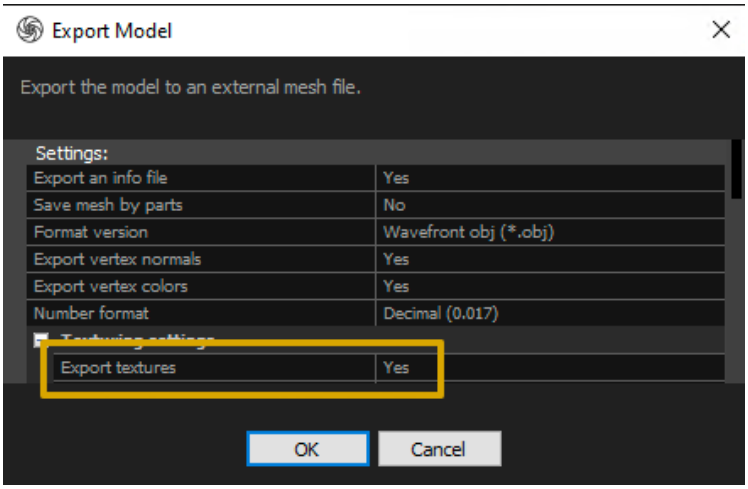
If you have a model in Reality Capture that has a computed and textured mesh, you can export the data necessary to process the model with Kintsugi 3D Builder by following these steps.

Kintsugi 3D Builder requires two files to be exported from Reality Capture: a 3D model in obj format, and the camera calibration parameters.

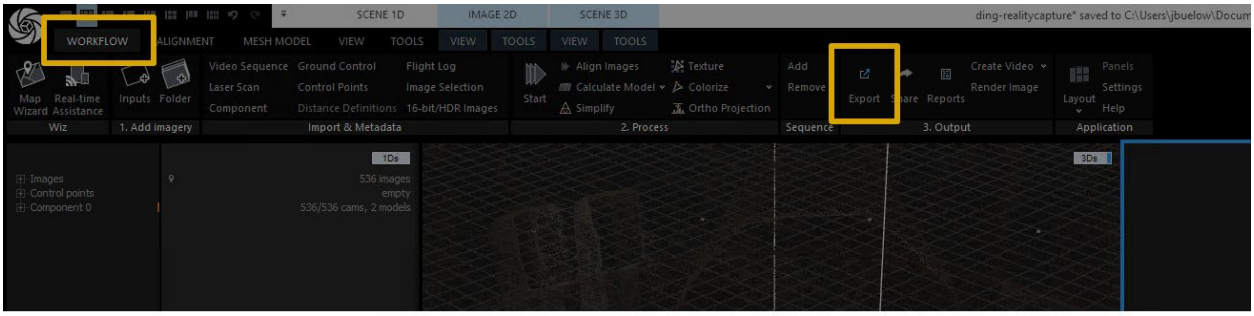
Export the 3D model from the workflow tab (Workflow > Output > Export), then select “Wavefront obj (*.obj)” from the list of formats.

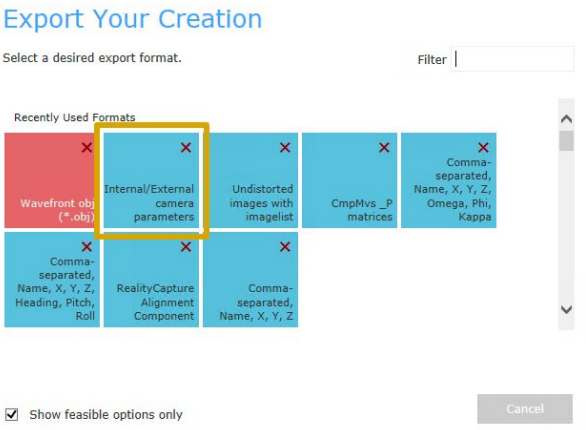


Select a location and filename for the obj file, then once prompted for export settings, ensure Export textures is set to “Yes”.

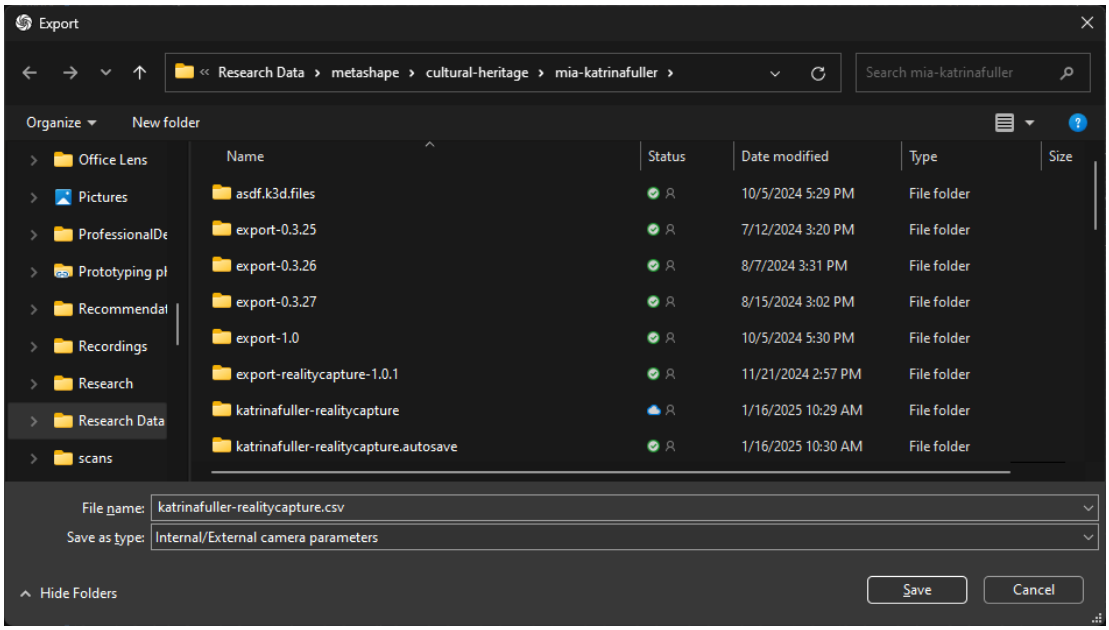


Export the camera calibration parameters from the Workflow tab (Workflow > Output > Export), then select “Internal/External camera parameters” from the list of formats.

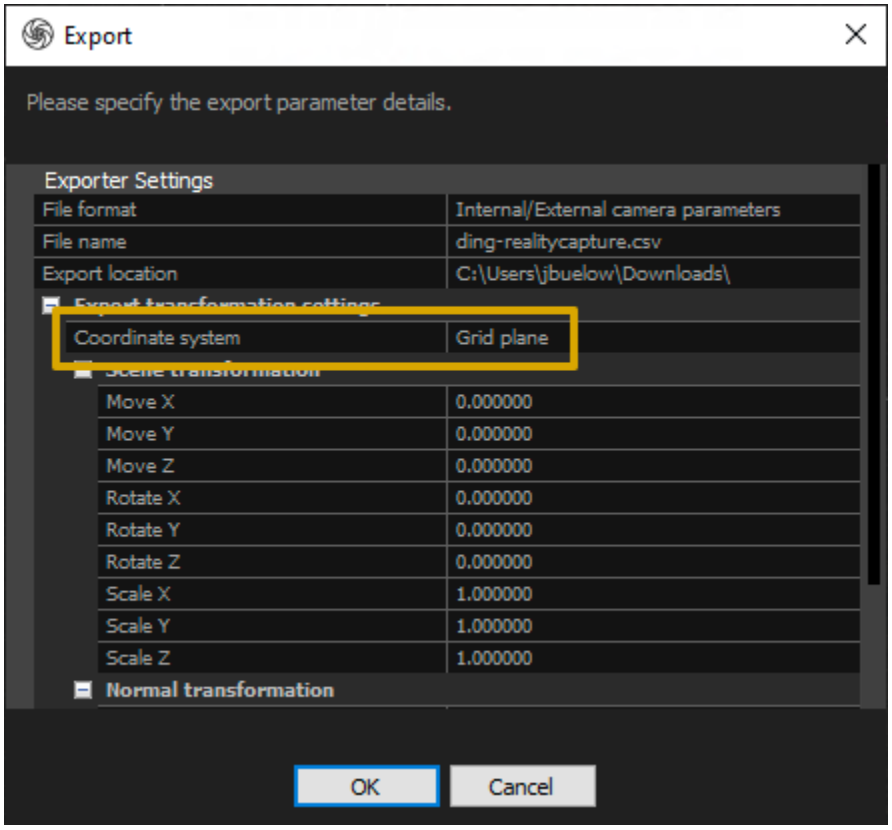




Select a location and filename for the CSV file, leaving the “Save as type” set to “Internal/External camera parameters”:

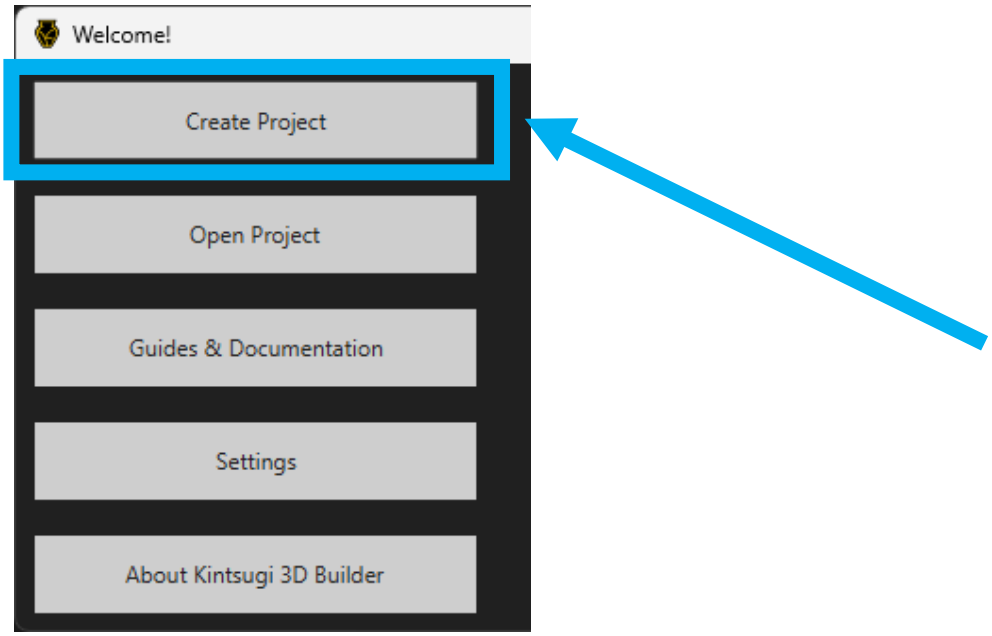


Once prompted for export settings, ensure Coordinate system is set to “Grid plane”.

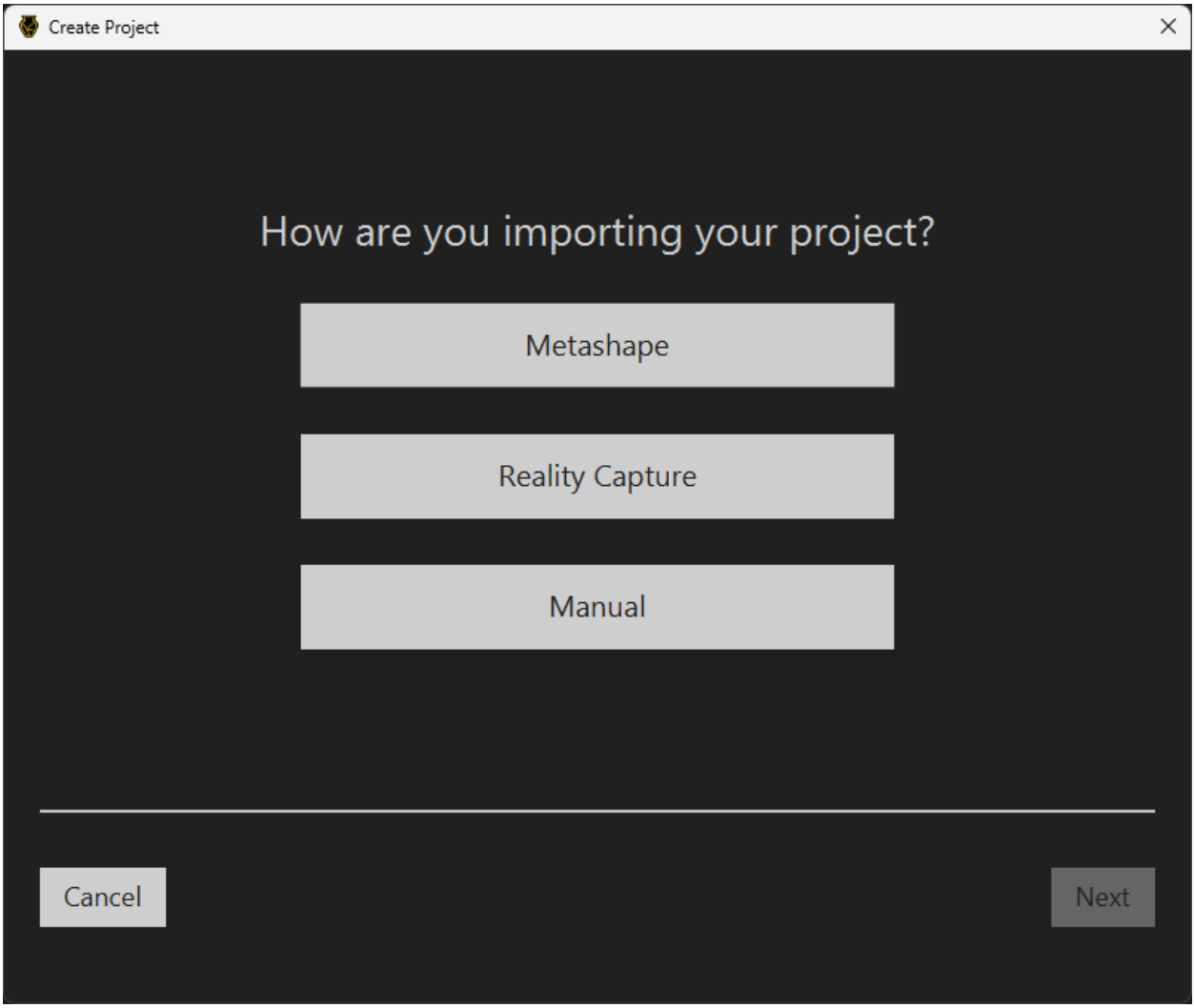


Import into Kintsugi 3D Builder

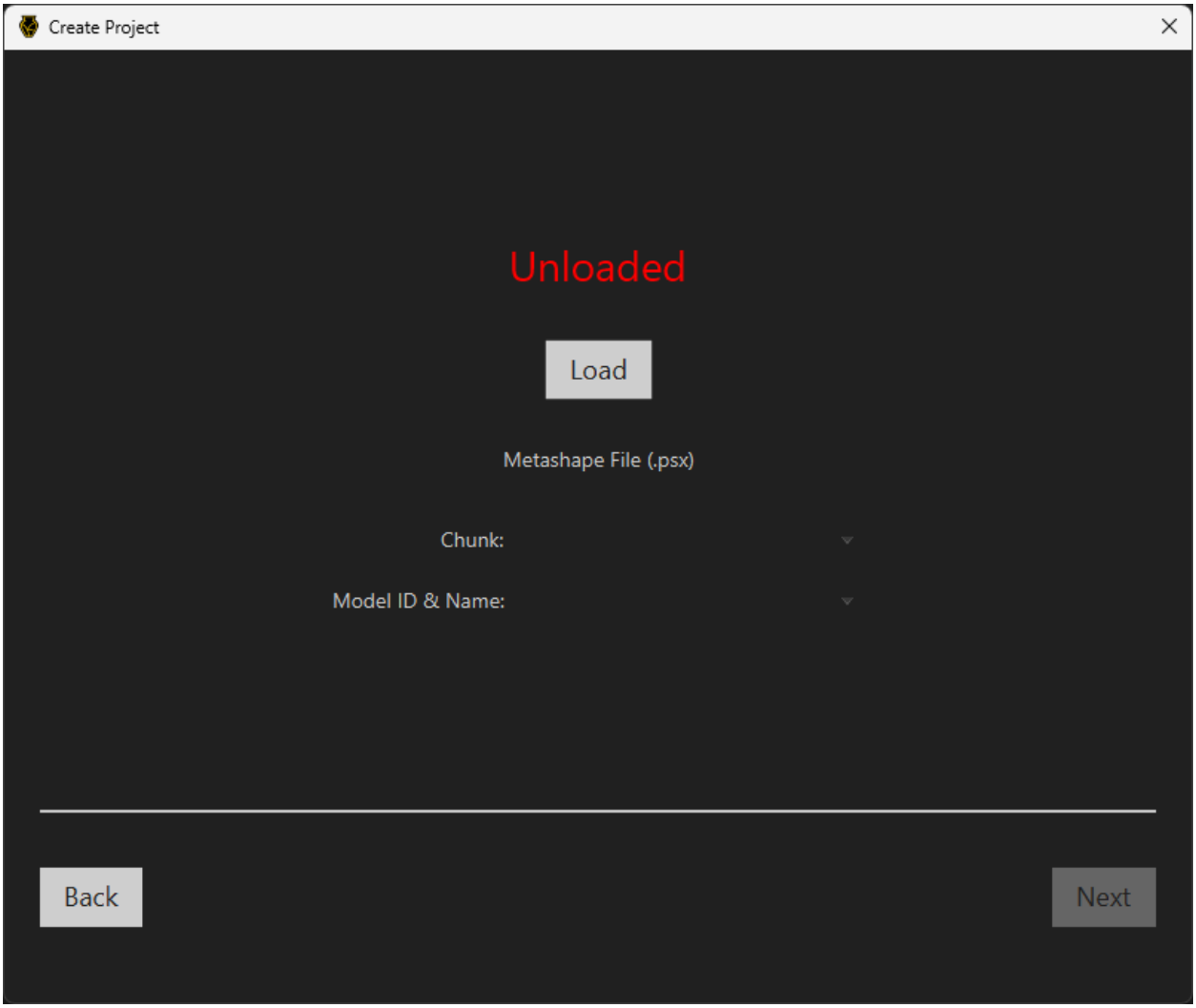
To import photogrammetry data into Kintsugi 3D Builder, choose “Create Project” from the Welcome Window or “File” menu.



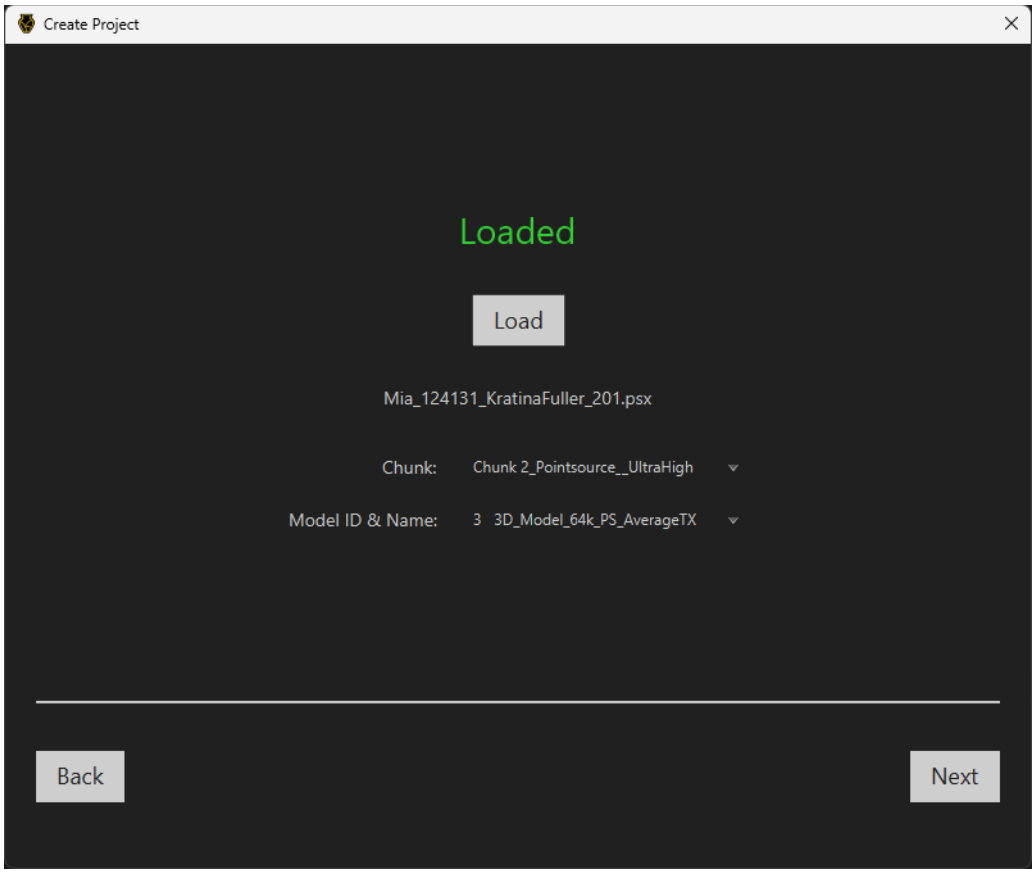
You will be prompted with three import options: Metashape, Loose Files, or Reality Capture. Note that you can navigate forwards and backwards through pages in the Create experience using the A and D keys. Use A and D to move backwards and forwards one page, respectively.



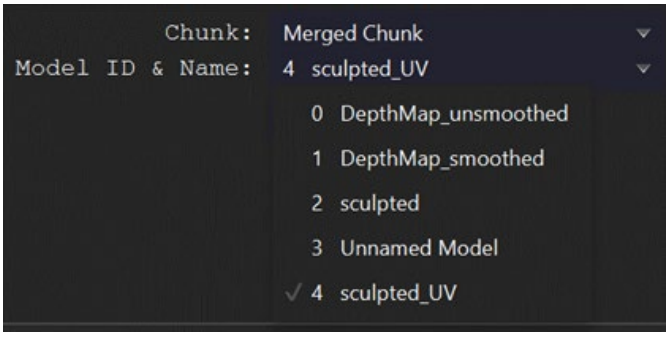
Metashape Import



You will be prompted to import an Agisoft Metashape file (.psx). Upon uploading a .psx file, all dropdowns will be populated to reflect your data.

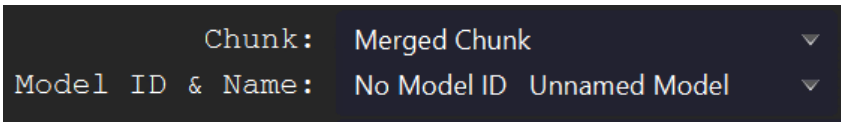


All chunks and models will retain the names they were given in Metashape. You can select your desired chunk from the dropdown. Within a chunk, you can select a desired model.

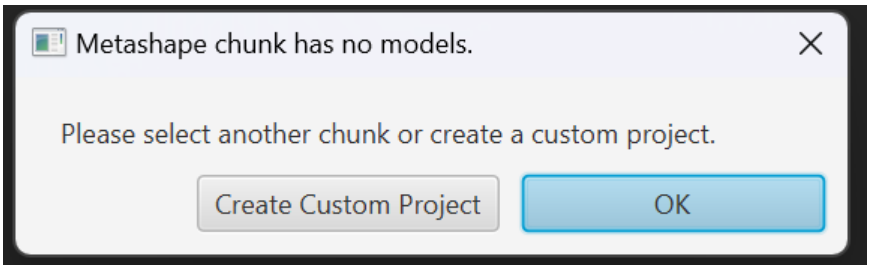


By default, the chunk labeled as “active” in Metashape is selected by the interface. Similarly, the model within that chunk marked as “default” will be selected by default.

If a model is missing an ID or a name, the interface will indicate this.



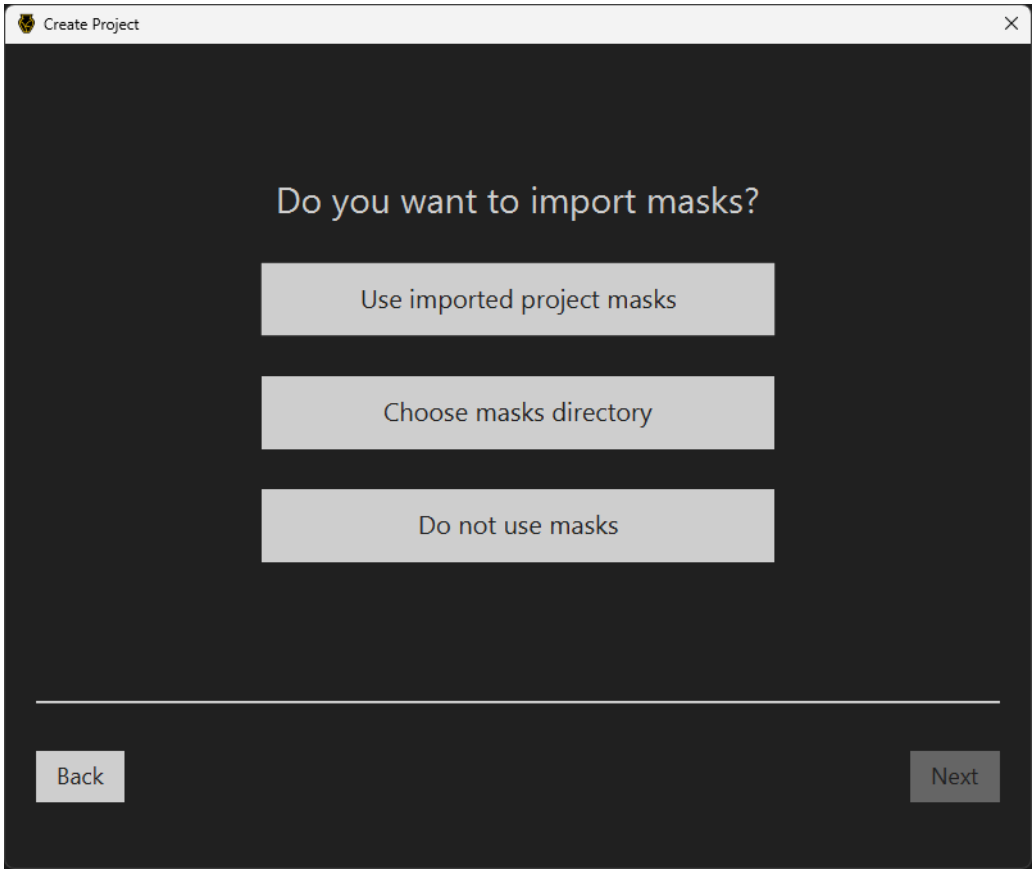
Upon importing your Metashape file or selecting a chunk, you may get a popup indicating that your chunk has no models.



You cannot create a project using this chunk's data. Hit "OK" to dismiss the popup.

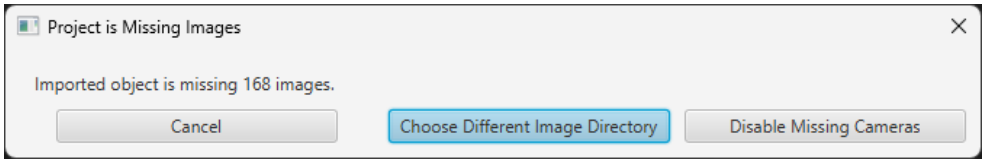
Alternatively, hit "Create Custom Project" to open the custom loader. Using the custom loader, you can import individual camera XML, mesh, and image files.

Click "Next" to proceed; you will be prompted to select a source for mask images or to not use masks.



Masks can be either imported from a Metashape project, or from a directory. If using the directory option, mask images should have the same name as the original photographs, or should have the same name with the suffix "_mask" appended. The file extension for the masks does not need to match the file extension for the source photographs. As an example, IMG_1234_mask.png" would be automatically associated with "IMG_1234.JPEG."

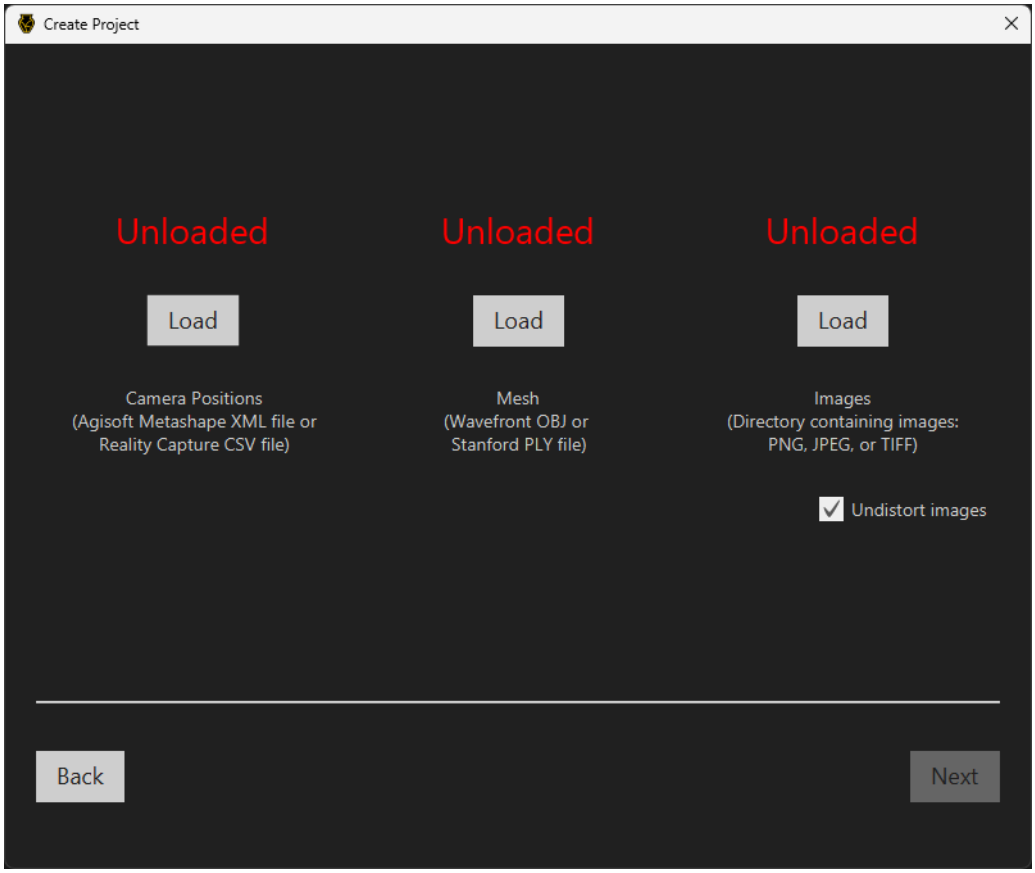
Click "Next" to advance. Kintsugi 3D Builder will at this time attempt to find all of the images referenced by your photogrammetry project. If any are missing, you will see a dialog like the following:



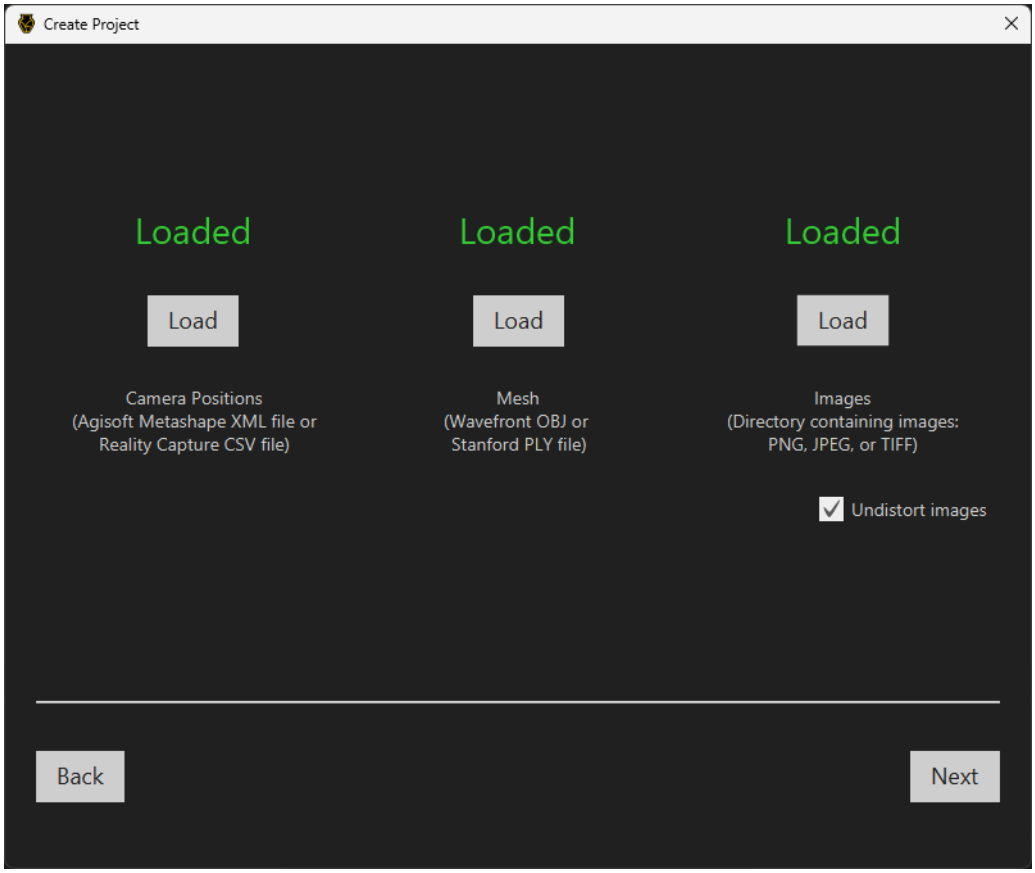
If the photographs were moved to another directory, you can select "Choose Different Image Directory" to correct the issue.

Loose Files / Reality Capture

If you opt to export your camera calibration and 3D model from Metashape and import them as loose files into Kintsugi 3D Builder, you will be prompted to browse for the camera calibration data, 3D model, and image directory. You will have the same experience if importing data exported from Reality Capture.



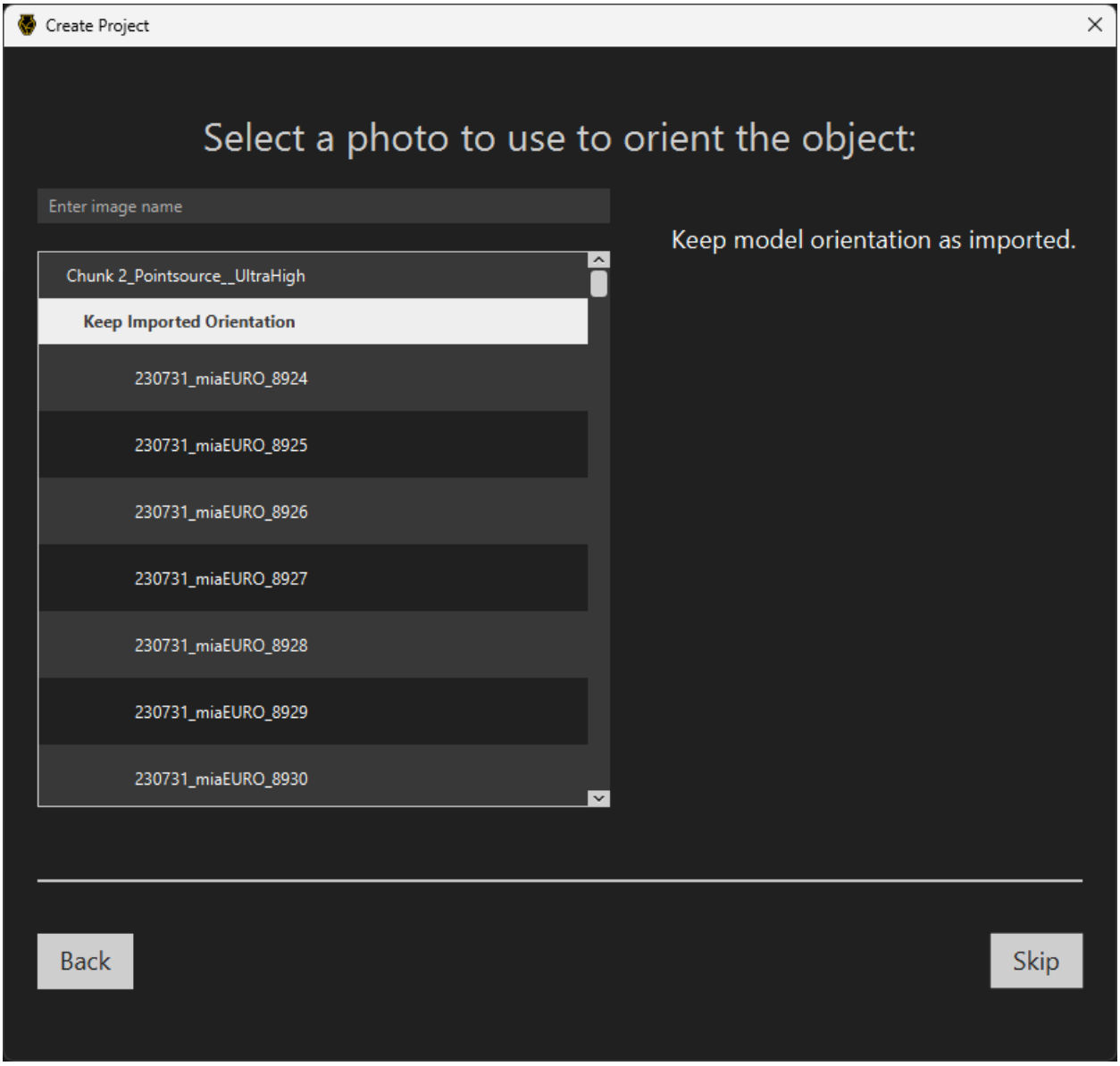
The interface will indicate when each file has been successfully located:



You can also choose to disable Kintsugi 3D Builder's built-in image undistortion by unchecking the checkbox labelled "Undistort images." This is an advanced feature that can be used to fall back on the undistortion implementation of the photogrammetry software in case there are distortion issues observed after importing into Kintsugi 3D Builder.

Model orientation

After selecting the files, you will be prompted to select a view for determining model orientation:



By default the option to “Keep Imported Orientation” is selected, which will use the original orientation of the 3D model as imported. Alternatively, you can select a view that will be used to determine the model’s orientation so that the default orientation matches the orientation in that photograph. If the photograph is 90 or 180 degrees off from the desired orientation (i.e. if the photograph was taken in Portrait orientation), you can use the buttons on the right side of the window to fix the orientation.

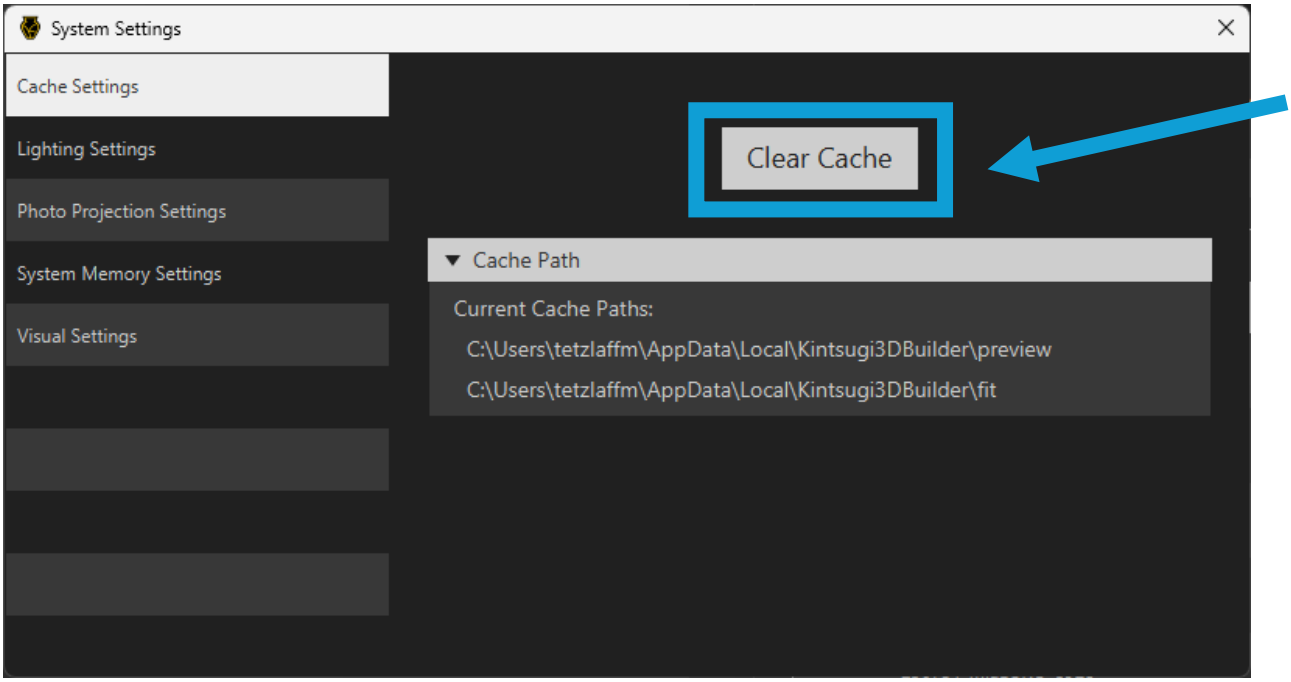
Hit Confirm to finish creating your project. You will be asked to provide a filename and location for your Kintsugi 3D project. Kintsugi 3D projects use the .k3d file extension.

Warning: Creating a new project when another project is already open will close the old project without saving. Make sure you always save your work before creating a new project.

Preview-resolution undistorted images

It is normal for it to take several minutes to import the source photographs for a Kintsugi 3D project. During this time, Kintsugi 3D Builder is reading, "undistorting," and downscaling each image to support previewing the projection of the photos onto the 3D model. Kintsugi 3D Builder continues to reference the original, full-resolution images and will use those for the specular fit process.

The "preview" images are stored in hidden folders under the user home directory: within "AppData/Local/Kintsugi3DBuilder" on Windows or "Library/Application Support/Kintsugi3DBuilder" on Mac OS, and are cached to make opening the project quicker in the future. To free hard drive space, the cache can be cleared manually if desired from the settings menu:



The cache directories can also be accessed by clicking on the paths listed in the settings menu.

File menu

Create New project

CTRL + N or CMD + N

This allows you to go through the process of importing photogrammetry data to set up a new project, as described in the previous section.

Open Project / Save / Save As

Open Project: CTRL + O or CMD + O

Save: CTRL + S or CMD + S

Save As: F12

Kintsugi 3D Builder allows you to save your work as a project that you can come back to later. There are two ways of saving in Kintsugi 3D Builder. The first option is to save a full project as a .k3d file, which will save project settings and the processed textures in addition to the camera information for the photographs and the file paths to the triangle mesh and the original images. This is the normal way to save a project in Kintsugi 3D Builder. The second option is to save only a .vset file, which will only save the camera information for the photos and the file paths – no project settings or texture processing results will be saved if you save as a .vset file. This is intended for use in situations when you want to distribute your data but don't need to provide a full project file. Note, however, that when you save as a .k3d file, Kintsugi 3D Builder still will save a .vset file in the same directory to store the camera information for the photos and file paths.

Once you have saved a project, you can open it again later. Warning: Opening a project when another project is already open will close the old project without saving. Make sure you always save your work before opening another project.

Close Project

CTRL + W or CMD + W

This will close the current project and release the resources being used on the graphics card. Don't forget to save your work before closing a project. Although this feature is provided for convenience, it is recommended that you completely exit Kintsugi 3D Builder periodically to reduce the likelihood of crashes.

Additional Export

This menu item has a submenu containing several useful options for exporting images from Kintsugi 3D Builder:

- **Generic**: This is a general-purpose tool that permits the use of custom shader programs. This is intended to be used by users with experience in computer graphics programming.
- **Single Screenshot**: This tool exports the current image in the 3D viewport at a custom resolution.

- **Resample**: This tool exports a sequence of images rendered from the perspective of the cameras in another view set file.
- **Orbit Animation**: This tool exports a sequence of images, beginning with the current image in the 3D viewport, and then orbiting the object in a manner that maintains the same inclination and camera distance while changing the azimuth at a constant rate. The number of frames and the resolution of the frames can be specified.

Export glTF

This allows you to export the 3D model with the processed textures for use in other programs. For convenience, it is included under both the File and Workflow menus. Details are described under the "Workflow" section of this document.

Recent Projects

Recently opened or saved projects will be shown here for convenient access.

Clear Recent Projects

Clear Missing Projects

Removes references to items not found in file explorer. Will not modify your file system.

Clear All Projects

Removes references to all recent projects. Will not modify your file system.

Settings

Manipulates settings for Kintsugi 3D Builder. Click here to navigate to Settings documentation

About Kintsugi 3D Builder

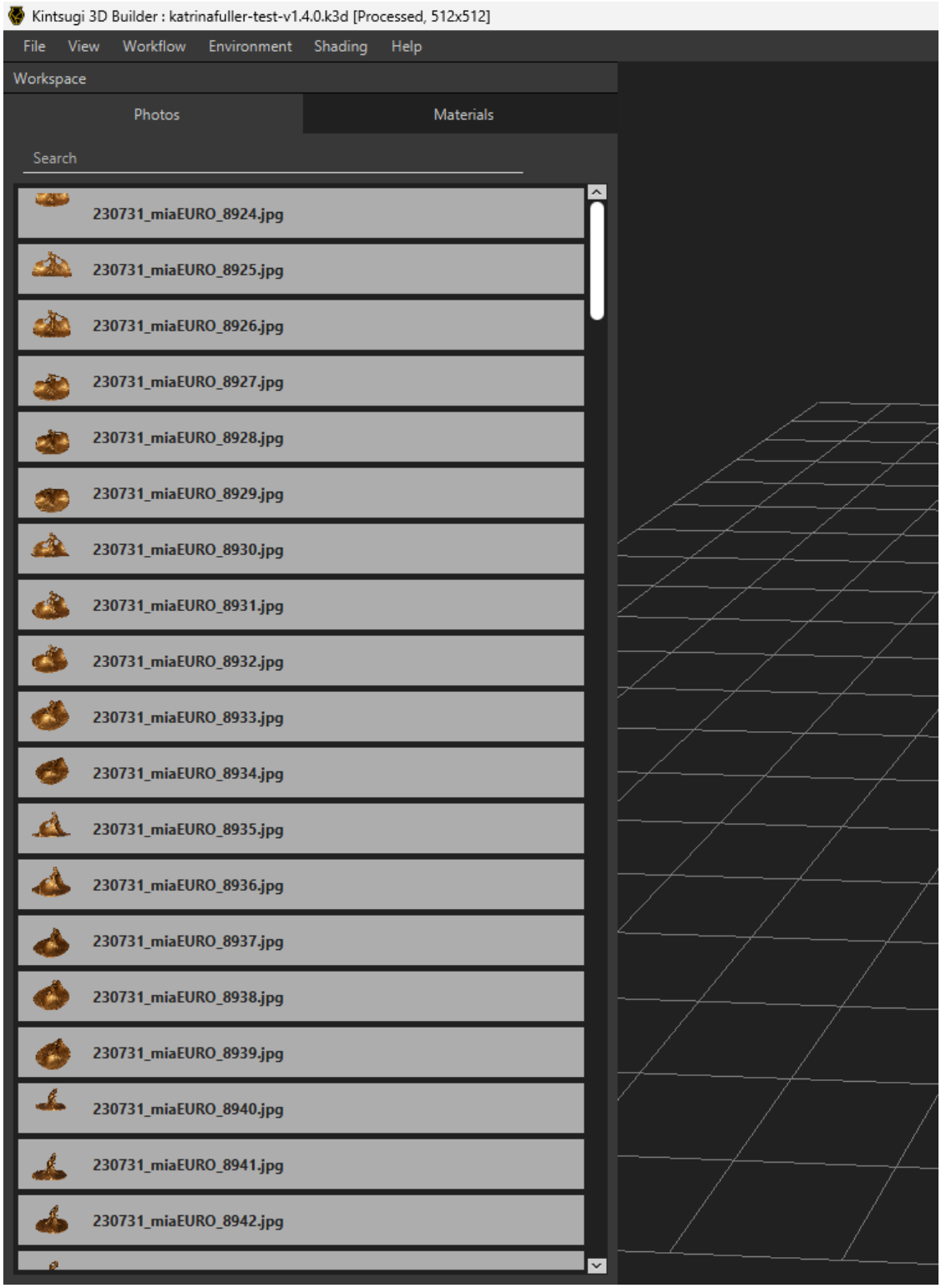
Opens a window with version and copyright information for Kintsugi 3D Builder.

Exit

This completely exits Kintsugi 3D Builder. Don't forget to save your work before exiting.

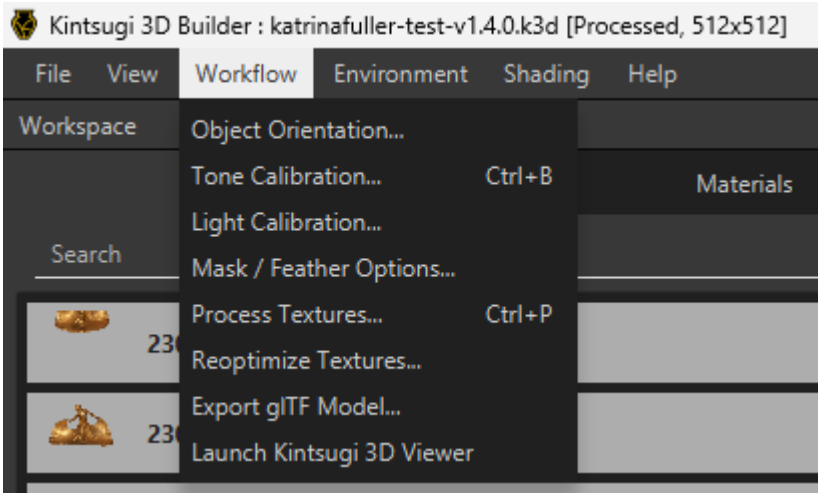
Workspace

The “Workspace” panel shows you all the elements of your Kintsugi 3D Builder project. Initially, this will show you a list of photographs referenced by the project:



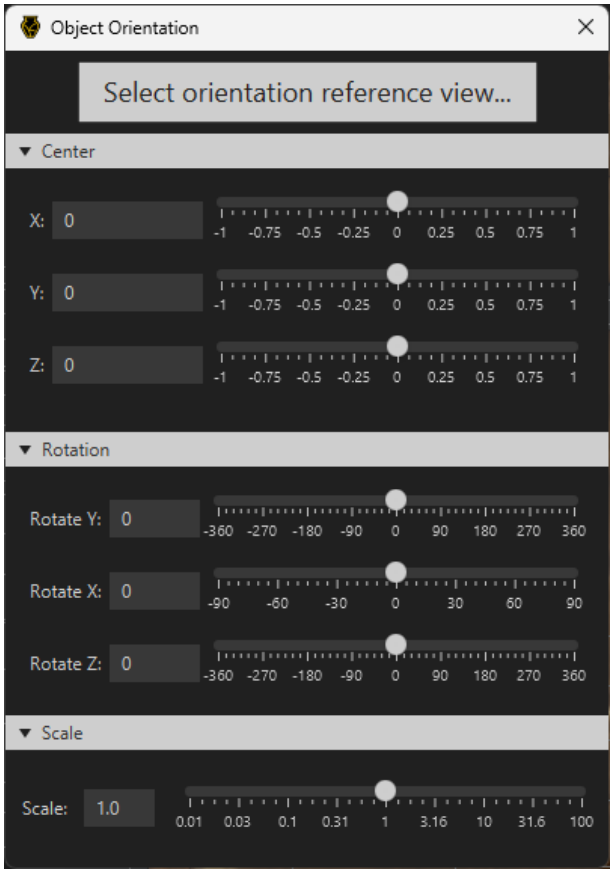
Workflow

The “Workflow” menu is the central place for performing the steps necessary to process textures and export a model with empirically based color and specularity.



Object Orientation

While Kintsugi will try to orient the object based on a reference photograph, the orientation, center, and scale of the 3D model can be adjusted manually.



These adjustments are applied on top of any orientation specified based on a reference view when the project was created. To change the image used for the original reference orientation, you can click the “Select orientation reference view” button at the top of the window.

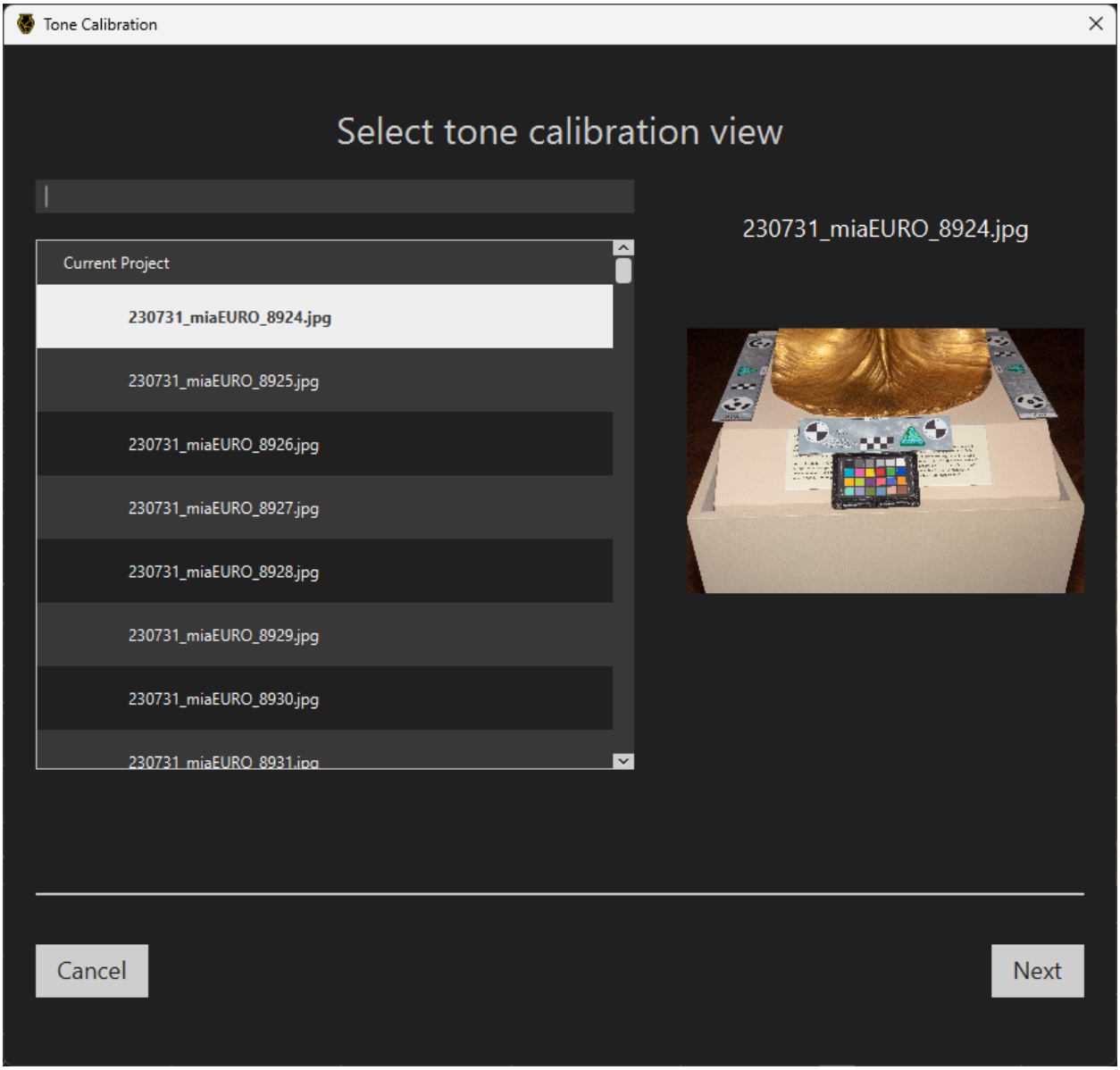
Tone Calibration

CTRL + B or CMD + B

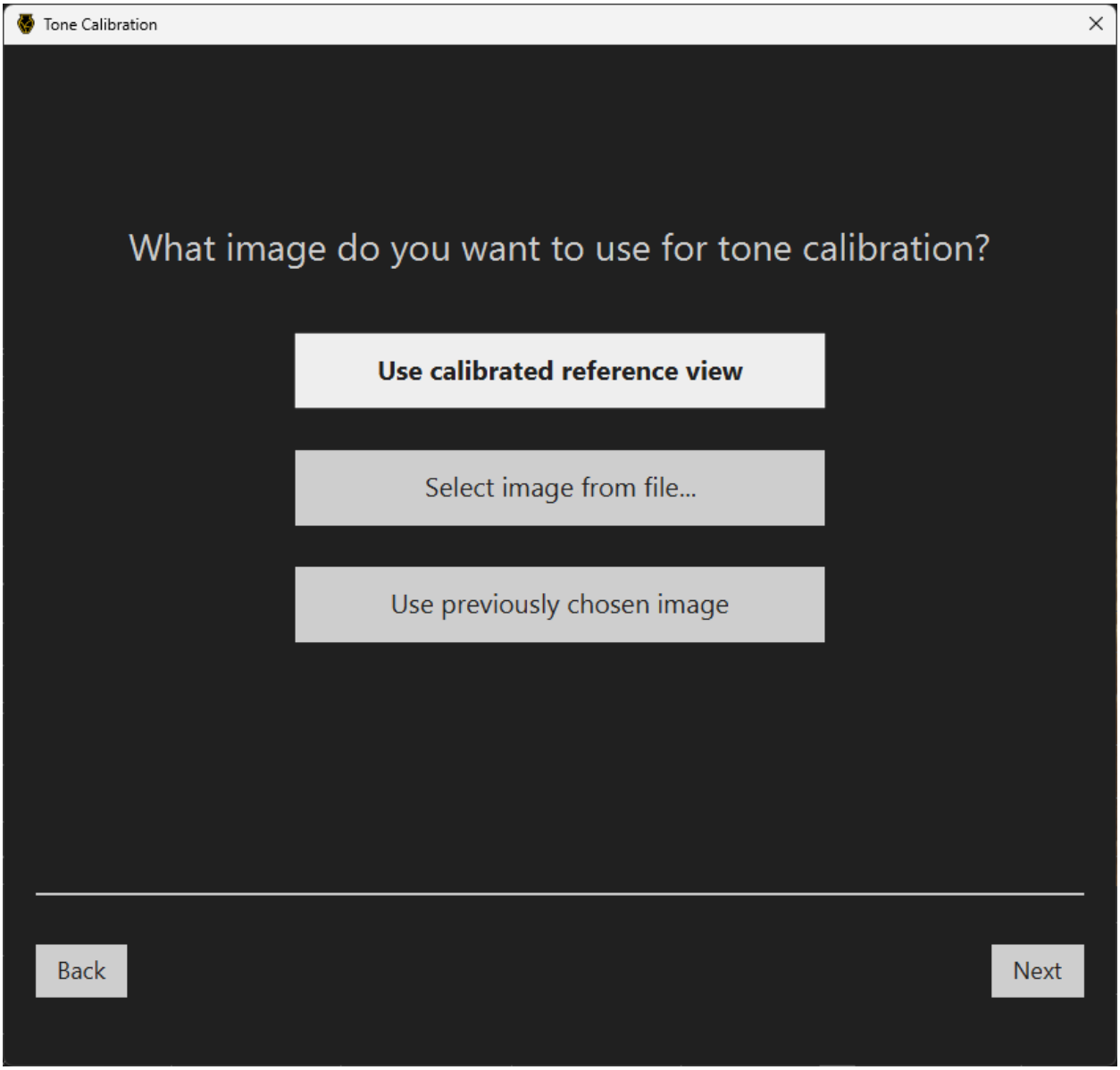
One fundamental advantage of an empirically-based workflow for photogrammetric material capture is the ability to leverage established tone and color management tools from 2D photography. The tone calibration task is used to ground the textures generated in absolute reflectance measurements, using a color calibration chart as reference. (Currently, the MacBeth ColorChecker, sold by Calibrite, is supported.[8]) This can somewhat “undo” tone mapping applied to the raw images under the assumption that all the flash photographs were taken under a light source with the same intensity and have had the same tone mapping applied. This is accomplished by leveraging the fact that each of the grayscale squares on the calibration chart has a known reflectance value. Only the grayscale squares on the chart are used to calibrate only the “value” component of color; it is assumed that color management for hue and saturation is primarily taking place in a tool external to Kintsugi 3D.

To perform the tone calibration task, first select a reference view from among the calibrated cameras imported from photogrammetry. This reference view does not need to contain the calibration chart, but should ideally be taken from the same vantage point or a similar vantage point to a photograph taken that does contain the calibration chart. This reference view is used to estimate the distance from the camera to account for distance-based falloff, if enabled. It is assumed that the distance from the light to the calibration chart is the photograph containing the calibration chart is approximately the same as the distance from the light to the closest point on the object in the reference view. To satisfy this assumption, the chart should be photographed directly in front of or next to the object, under the same flash lighting and with the camera in the same place as another image that was taken for image-based rendering, as described earlier under the photography guidelines.

[8] https://calibrite.com/us/product/colorchecker-passport-photo-2/

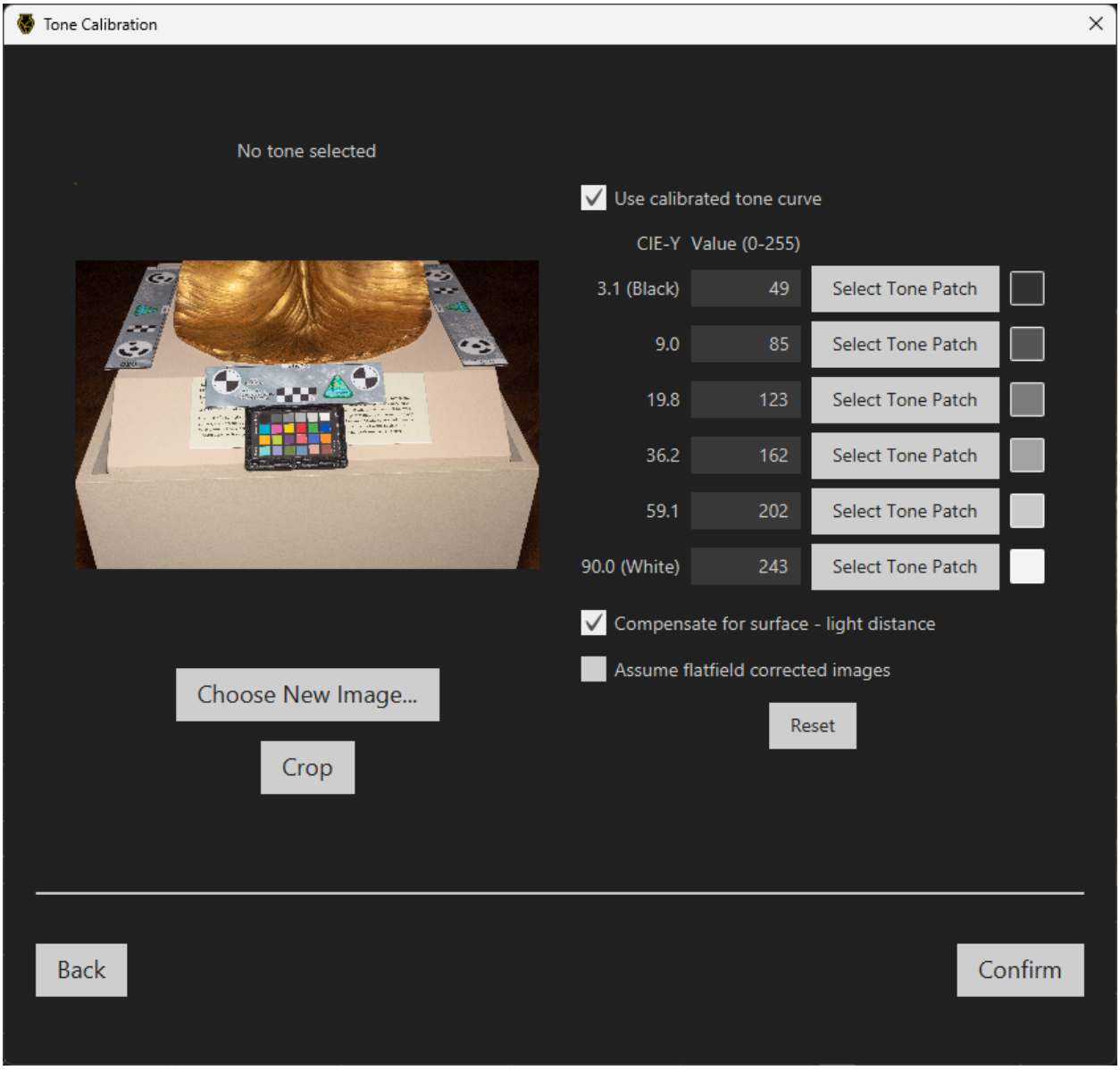


Click “Next” to confirm. You will be presented with options for selecting the photograph containing the calibration chart.

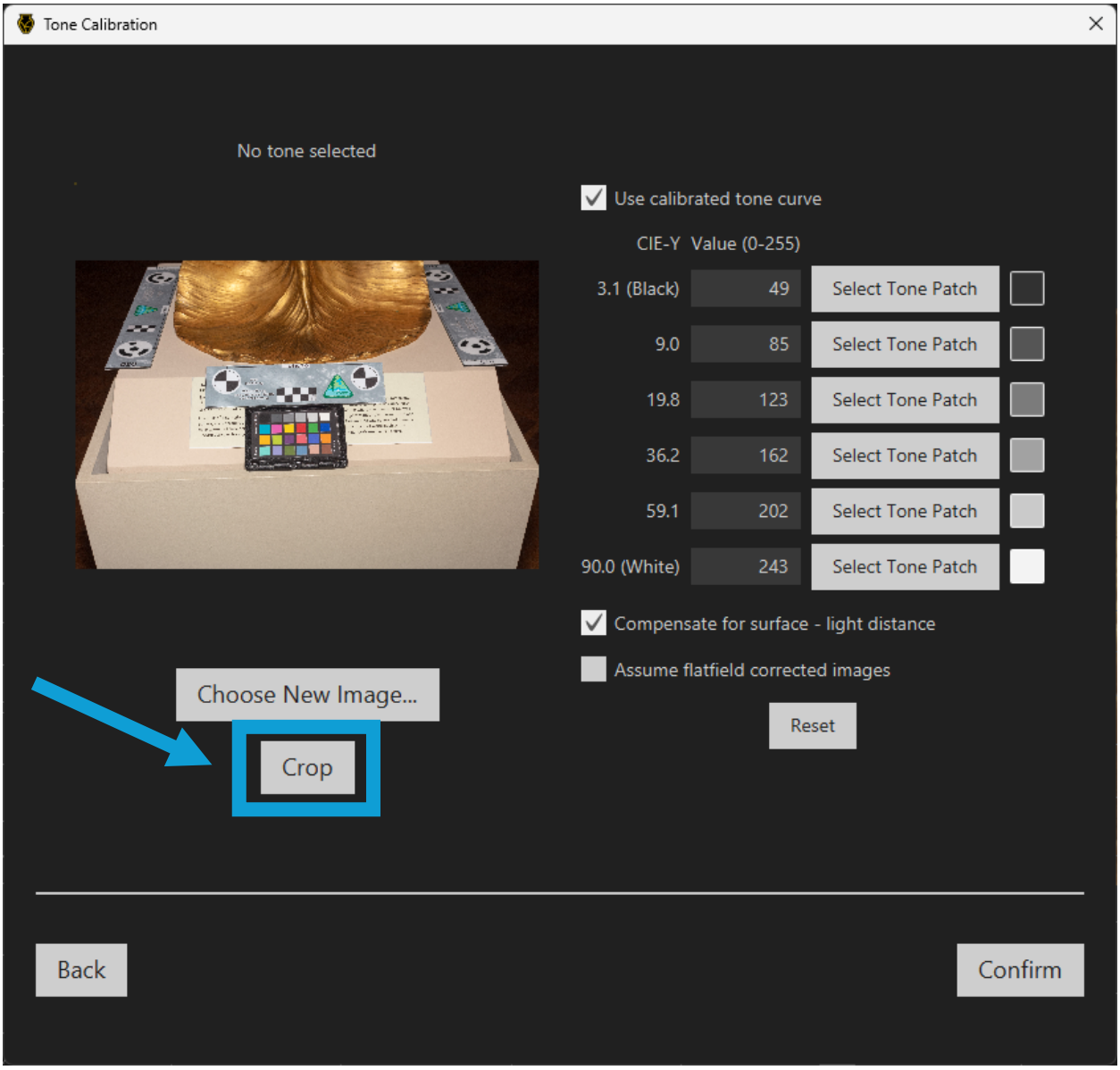


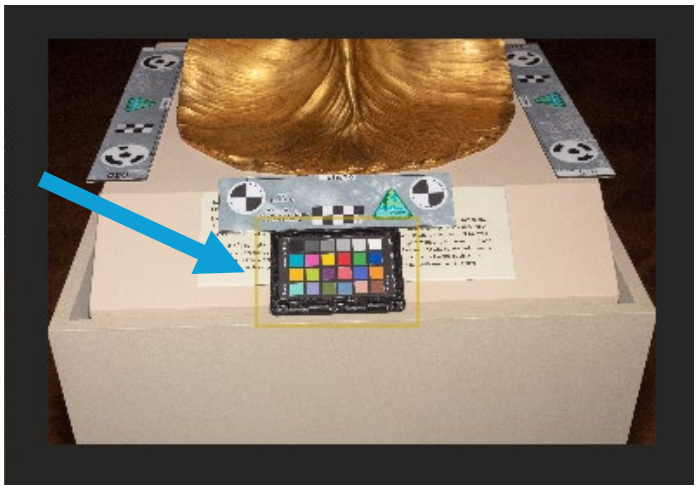
If the selected image contained the calibration chart, you can leave the default of “Use Primary View” selected and click “Next” to proceed to perform the calibration on that photograph. Otherwise, choose “Select Image…” to select the photograph of the calibration chart from your hard drive. (If you need to repeat this step for any reason, on subsequent iterations you will also have an option to “Use Previous Image” which will remember the photograph previously used for this task.)

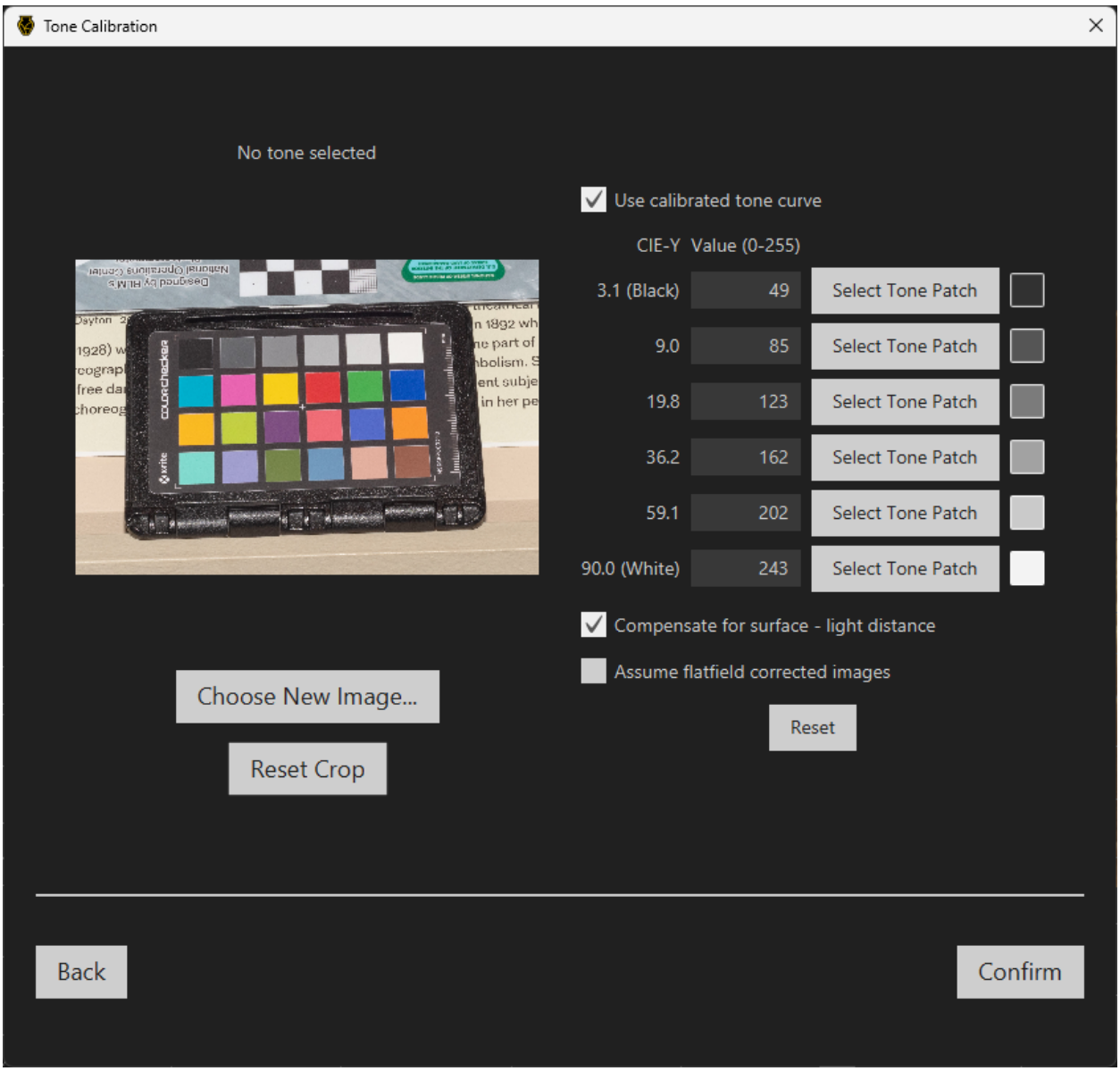
After specifying the image containing the calibration chart, click “Next” to advance to a screen that will allow you to perform the calibration itself:



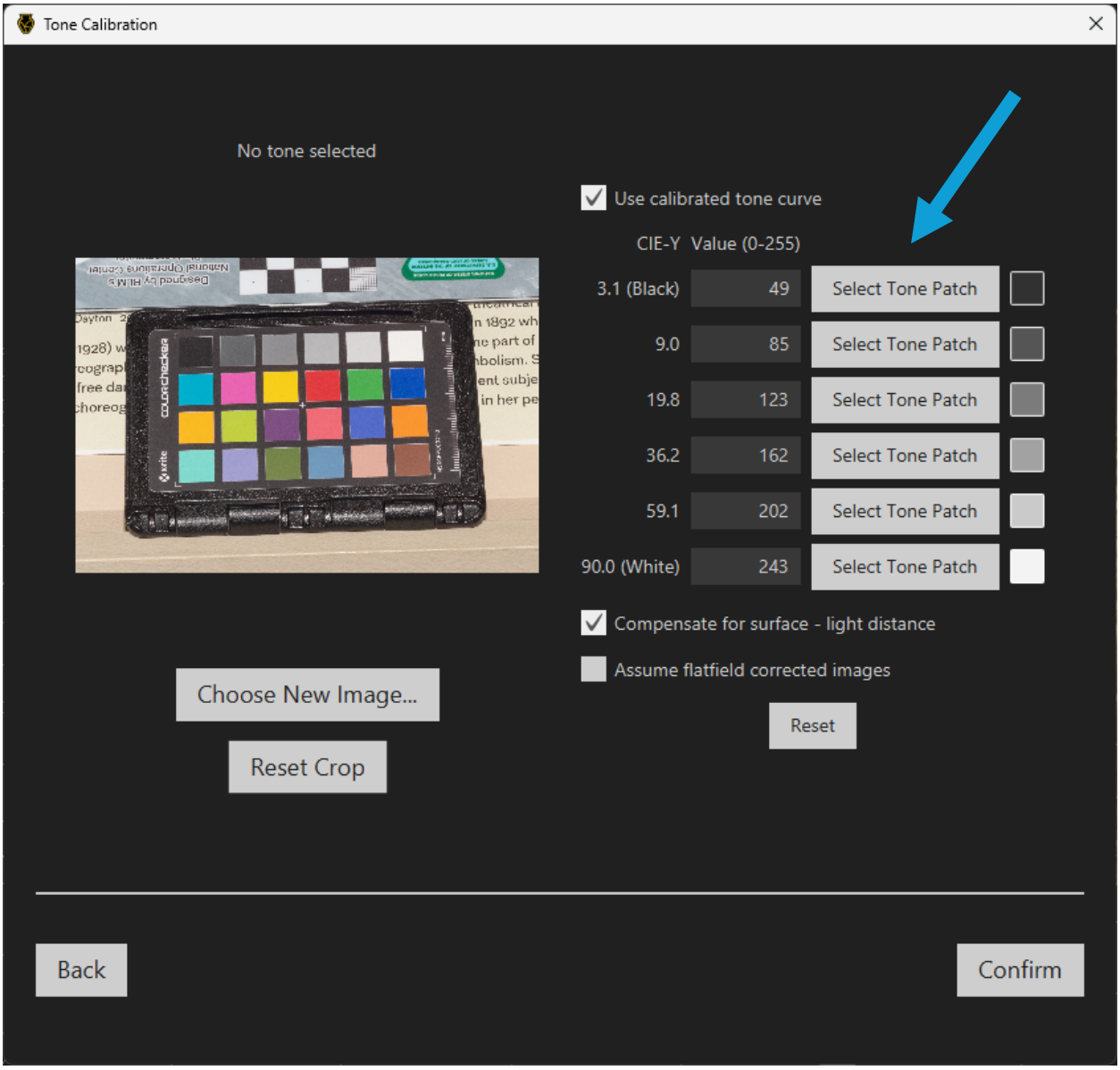
The photograph can be cropped to make patch selection easier:

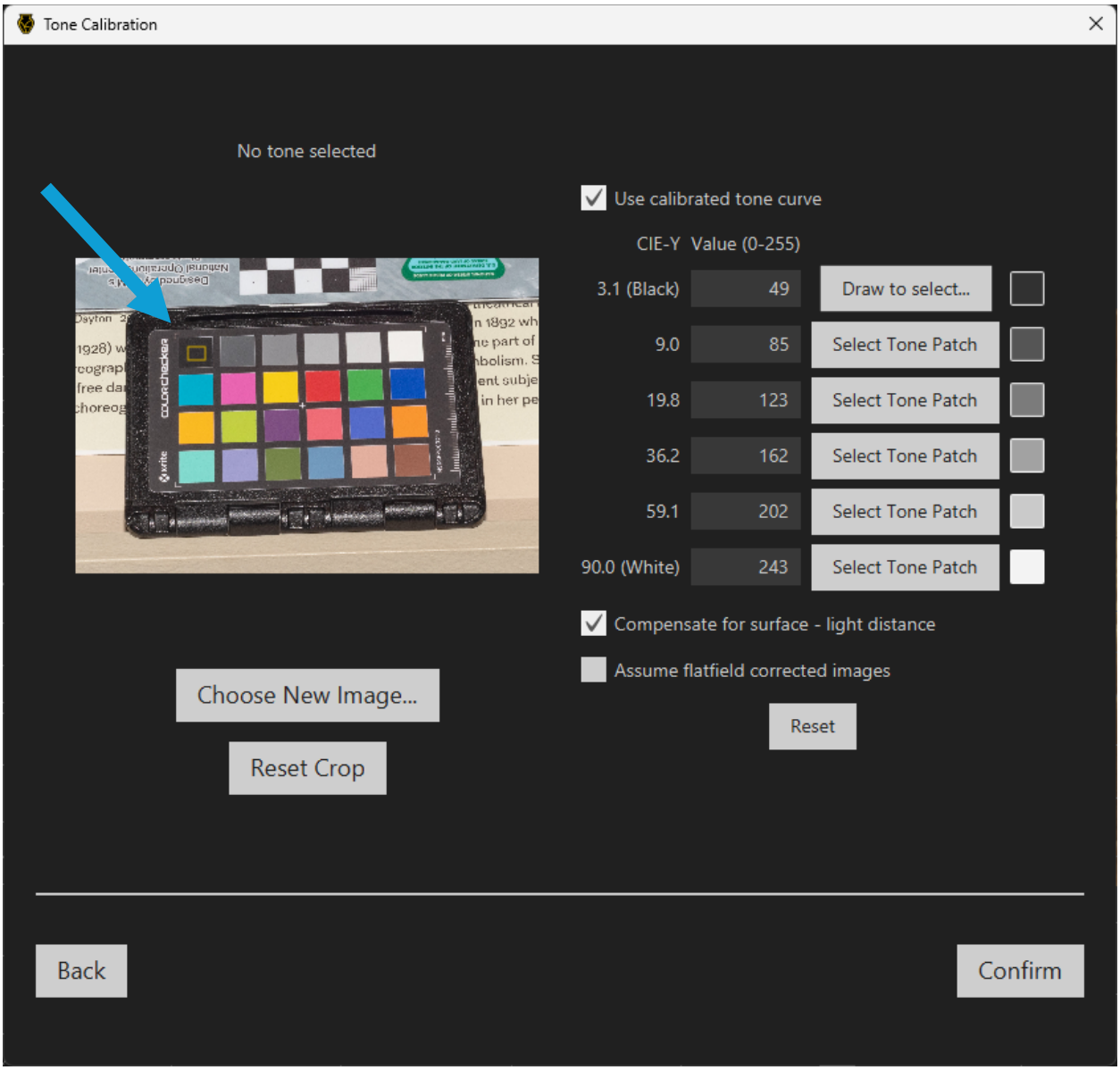




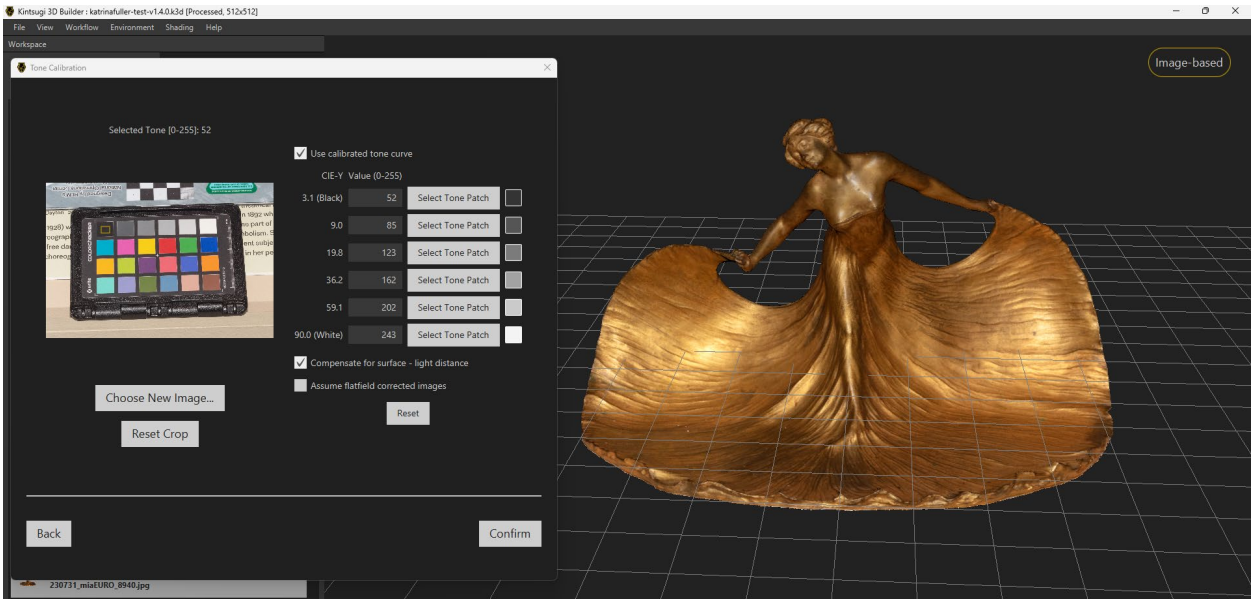

Tone Calibration
No tone selected
Use calibrated tone curve
CIE-Y Value (0-255)
3.1 (Black) 49 Select Tone Patch
9.0 85 Select Tone Patch
19.8 123 Select Tone Patch
36.2 162 Select Tone Patch
59.1 202 Select Tone Patch
90.0 (White) 243 Select Tone Patch
Compensate for surface - light distance
Assume flatfield corrected images
Reset
Choose New Image...
Reset Crop
Back
Confirm

Next, for each of the six grayscale squares, select a tone to calibrate on the right side of the interface using the corresponding button labelled “Select Tone Patch,” and draw a rectangle inside the corresponding patch in the photo to capture the reference value for that patch.





As the patches are selected, the model's appearance will be changed in real-time by applying the specified tone curve.



By associating the known reflectance of the squares with the pixel values observed, Kintsugi 3D Builder can then map the pixel values in any of the photographs to absolute reflectance values. The software is additionally able to automatically account for the falloff in intensity as the distance from the light source increases. If the task was performed successfully, the appearance should be reasonable, albeit possibly brighter or darker than before.

If the task was performed incorrectly, the appearance may appear strange; for instance excessively bright or dark and lacking the correct shading and color (this can happen, for example, if the tones are selected in the opposite order).

Two additional settings are available as part of the tone calibration process:

- “Compensate for surface – light distance” is enabled by default, and indicates that inverse-square falloff should be applied when determining the reflectance represented by a particular pixel in a photograph. If disabled, parts of the object that are further away from the light in more images may appear darker, and possibly other unusual artifacts like bias in normal maps may occur depending on where the photos where taken. The main reason to disable this would be if a dataset includes photographs taken with different exposure settings to explicitly compensate for different light distances.
- “Assume flatfield corrected images” is an option available when “compensate for surface – light distance” is enabled. If “assume flatfield corrected images” is enabled, the surface – light distance will be taken to mean the plane parallel to the camera passing through a point – resulting in shorter distances near the edge of the photo. This assumes that a process has been used prior to importing into Kintsugi 3D Builder to perform “flatfield correction” to account for the greater light distance at the edge of images.

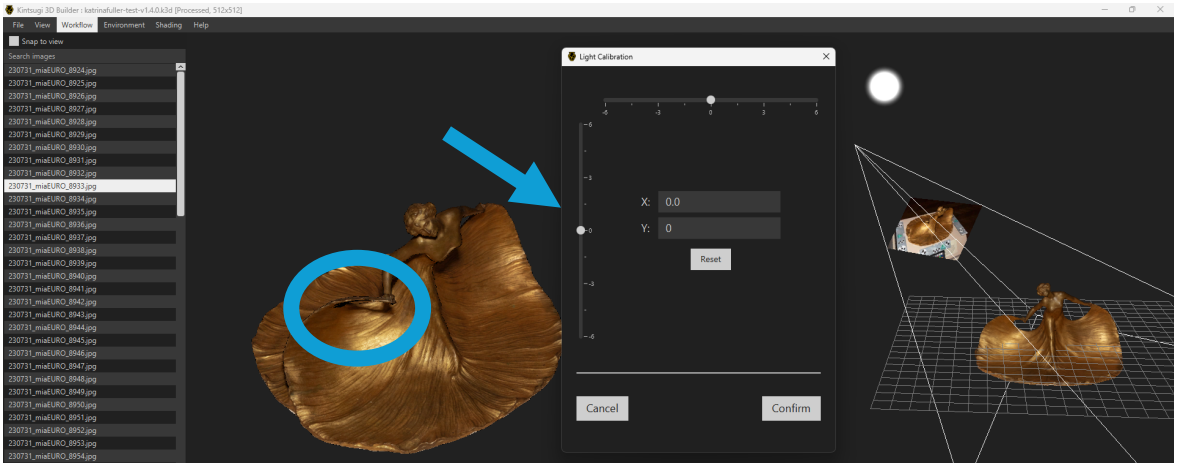
Light calibration task

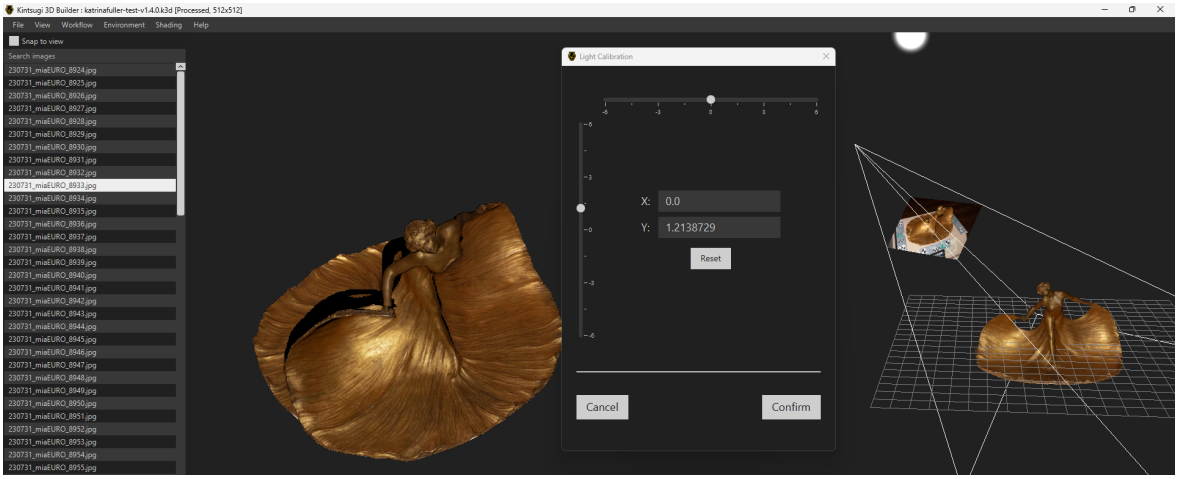
In practice, a light source can never be perfectly coincident with a camera. The light calibration task is used to calibrate the physical location of the light source relative to the camera lens axis. This is important for producing accurate normal maps.

In this task, you will see two views of the object. On the left, you will see a projection of a single photograph onto the object. You can change which photograph is used for this task using the list of photographs on the far left of the interface. It is important to choose an image that has clearly visible shadows. On the right, you will see a 3D visualization of the object, the camera, and the light source.

Your goal is to use the sliders to adjust the perspective of the single photo projection on the left until the object occludes all shadows that it had cast upon itself in the projected photograph, and the silhouette of the object aligns with the silhouette of the shadows. When this is the case, the virtual camera will be located at the same position as the light when the photograph was taken.

You may see some “blank” areas that appear when doing this, which were occluded from the camera in the original photograph; this is expected and is not an indication that a mistake was made in this step.



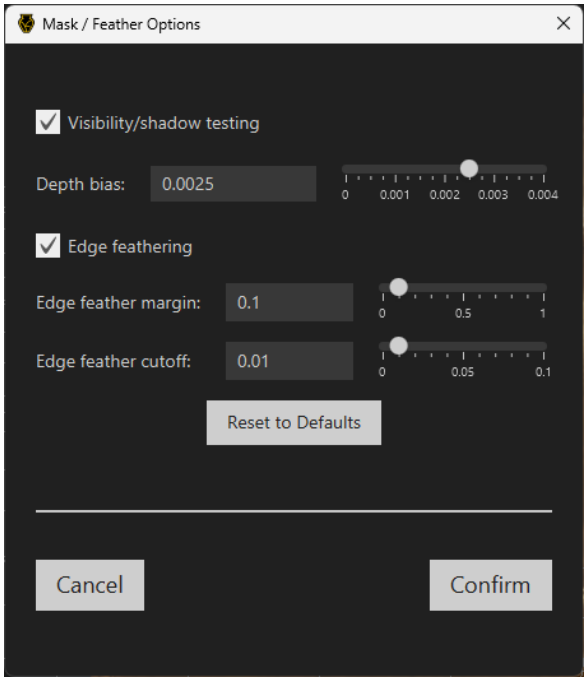


Mask / Feather Options

Kintsugi 3D Builder provides features to assist with two challenging edge cases with input data:

- Parts of the object may be hidden or shadowed by other geometric features on the same object.
- For large objects especially, parts of the object may be outside of the image frame, that is, a given photograph may clip off certain parts of the object at the edge of the photo.

The “mask / feather options” dialogue offers settings that attempt to address these issues.



- Visibility/shadow testing: When enabled, this option eliminates views bleeding through onto parts of the geometry that they could not see. “Preload Visibility and Shadow Testing” (under the global settings menu, section “visual settings”) is required for this to have an effect. The light calibration discussed in the previous section is used to determine the location of shadows.
    - Depth bias: Minimum value for detecting a difference between the depth in the depth image and the value in the current view. Make this small to help get less bleeding across visibility discontinuities. Increase it to eliminate more bleeding.
- Edge feathering: When enabled, this option decreases the contribution of pixels near a photograph’s edge to reduce seams at those edges.
    - Edge feather margin: The threshold (as a fraction of image size) at which the contribution of pixels starts to decreases.
    - Edge feather cutoff: The threshold (as a fraction of image size) at pixels no longer affect rendering or texture processing at all.

The effect of these settings will be seen immediately in the preview for “image-based” or “image-based with textures” shading modes, and will also affect the results of texture processing.

Process Textures

CTRL + P or CMD + P

Basic settings

“Process textures” is the step where the color, specular, and normal map textures are generated. You can choose a resolution from among several standard square, power-of-two options – higher resolutions will take more time to process (expected processing time scales with the total number of pixels in the texture).

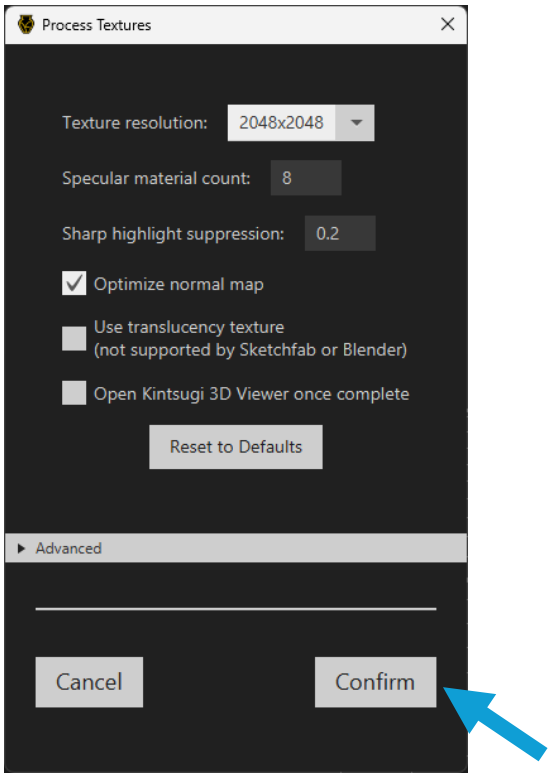
Kintsugi 3D Builder internally uses a “material basis” representation of specularity – meaning that a certain number of “basis materials” are chosen, and appearance at each pixel is a weighted average of those basis materials. Kintsugi 3D also optimizes a normal map to improve the accuracy of its expectations about where specular highlights will appear.

In addition to resolution, other options include:

- Specular material count – can increase or decrease depending on material complexity. Increasing material count will increase the time it takes to process.
    - Default value of 8 will work fine for most objects.
    - Can be lowered for homogenous materials (i.e. one solid color).
    - Can be increased for more complex materials (i.e. Cloisonné).
- Sharp highlight suppression prevents overfitting to specular outlier pixels. This enforces essentially a minimum roughness for specular highlights.
    - Default value of 0.2 will work fine for most objects.
    - Decrease this value to allow for sharper / shinier specularity.
    - Increase this value to further suppress specularity and any potential specular artifacts. At a value of 1.0, all specularity will be almost entirely suppressed.
- Optimize normal map: this can be disabled in cases where artifacts have been introduced by normal optimization.
- Use translucency texture – only enable if your object exhibits translucency (i.e. jade, wax, etc.). This introduces an additional “constant term” texture that approximates the appearance of translucency. Enabling translucency has minimal impact on processing time but may result in strange looking results for objects that don’t need it. The “pseudo-translucency” / “constant” texture is also not portable to most existing renderers and viewers, such as Blender or Sketchfab.

If Kintsugi 3D Viewer is installed alongside Kintsugi 3D Builder, the “Open Kintsugi 3D Viewer once complete” option can be used to automatically open the processed model in Kintsugi 3D Viewer after processing finishes.

Once the processing settings have been determined, click “Confirm” to run the texture optimization process.



Advanced settings

Other advanced settings are available for users who want to fine-tune the algorithm in more detail:

- Specular resolution: The number of sample points in each basis material function. Increasing this value will make the appearance of highlights more natural (although increasing beyond the default value of 90 will likely be imperceptible) and increase processing time. Decreasing this value will lower the quality of each basis material function but will reduce processing time.
- Specular complexity: This value must be a fraction between 0 and 1, defaulting to 1.0. Similar to specular resolution, decreasing this value will reduce processing time but potentially also lower quality. It is expected that specular complexity will impact quality less than specular resolution for the same reduction in processing time. At low values, this may have a similar effect to "sharp highlight suppression."
- Metallicity: This value must be a fraction between 0 and 1, defaulting to 0.0. Typically, a value of 0 or 1 exactly would be used, although intermediate values are allowed. This does not directly set the metallicity as in a PBR sense, but based on the same physical principles, a metallicity of 1.0 will cause diffuse color to be interpreted as metallic specular reflectance. Accordingly, when metallicity is set to 1.0, the (Lambertian) diffuse color texture will be completely black. Values in between 0.0 and 1.0 can also be specified to reduce but not eliminate the role of the Lambertian diffuse texture. The main use case for this would be to force an object's appearance to be represented as entirely non-Lambertian, metallic reflectance. Using this feature is expected (perhaps

counter-intuitively) to significantly reduce the quality of the “metallicity” material representation.

- Specular smoothness: This value must be a fraction between 0 and 1, defaulting to 1.0. This setting prevents discontinuities in the appearance of highlights for the basis materials. Setting to 0.0 disables this completely, while intermediate values reduce its impact without eliminating it altogether. The main use case for reducing this setting this would be if a surface actually has an unusual reflectance profile, which this feature would otherwise suppress. Such materials are extremely uncommon and not expected to occur in typical usage.
- Smith masking / shadowing: Enabled by default. This enables a physically accurate formula for specular light reflectance. If disabled, a simpler formula will be used for the “Cook-Torrance” mathematical model of light reflectance. Typically, this will have a minimal impact. The main use case for this would be when targeting a simpler viewer that uses a more basic lighting formula.
- Convergence tolerance: The threshold at which the algorithm terminates, defaulting to 0.00001. Typically, this quantity will be small enough that it will be displayed using scientific / engineering notation (i.e. 1.0E-5 representing 0.00001). If the root-mean-squared-error (RMSE) decreases by less than this threshold between subsequent iterations, the process ends. Setting this to a lower value will cause the algorithm to run longer for theoretically more accurate results, while setting it to a higher value will cause it to end sooner with possibly less accurate results.
- Minimum normal damping: This setting, defaulting to 1.0, affects how aggressively normals are adjusted. Setting this value lower could theoretically reduce processing time, but in practice is more likely to make the algorithm more unstable, increasing processing time and/or degrading results quality. Conversely, setting it higher will generally increase processing time but improve stability.
- Unsuccessful normal iterations allowed: Within the overall iterative process, the normal map optimization is also iterative. While some iterations are expected to fail to decrease error significantly (resulting in automatic adjustment to the damping parameter), if this happens multiple times in a row, that is typically an indication of convergence. This setting (defaulting to 8) determines the number of times that the normal map optimization can fail to reduce the RMSE by more than the convergence tolerance before the normal map is determined to have converged for the current overall iteration of the texture processing algorithm. Decreasing this setting will reduce processing time, but may also cause the process to end prematurely without significant optimization of the normal map. Increasing it may improve quality but will also increase processing time.
- Normal smooth iterations: This setting, when set higher than the default of zero, can enable an extra pass to smooth out the normal map by performing a simple average of adjacent pixels. This can be used as a quick fix if undesirable artifacts are appearing in the normal map due to issues with the source data; however, note that this process is itself not empirical. More iterations will make the normal map smoother (but also deviate more from the empirical basis), and increase processing time.

More information about the algorithm can be found in technical papers.[9] [10]

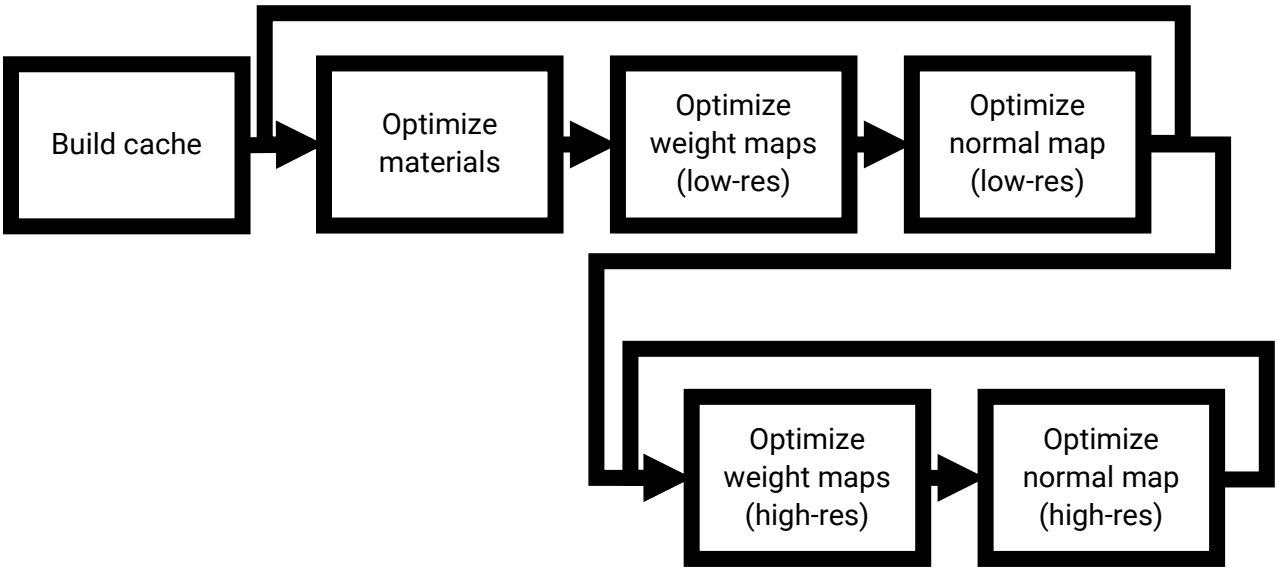
Processing algorithm

The process will take some time to complete – ranging from minutes to hours depending on the specified resolution and the hardware Kintsugi 3D Builder is running on. There are three stages:

- Build cache: This retrieves each of the original high resolution images and breaks them apart into texture-space "chunks" that can be loaded separately to reduce memory usage when performing high resolution processing. This step will be skipped if the cache has already been generated for a particular project with the same target texture resolution.
- A low-resolution fit to generate the basis materials.
- A high-resolution fit to generate the high-resolution textures.

When the process begins, the progress bars modal will open. This modal provides time estimates and other useful information. Click here for more information about the progress modal.

The full algorithm is illustrated in the following diagram:



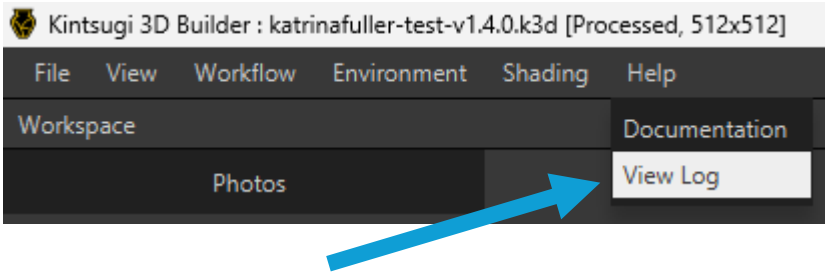
Using the log window

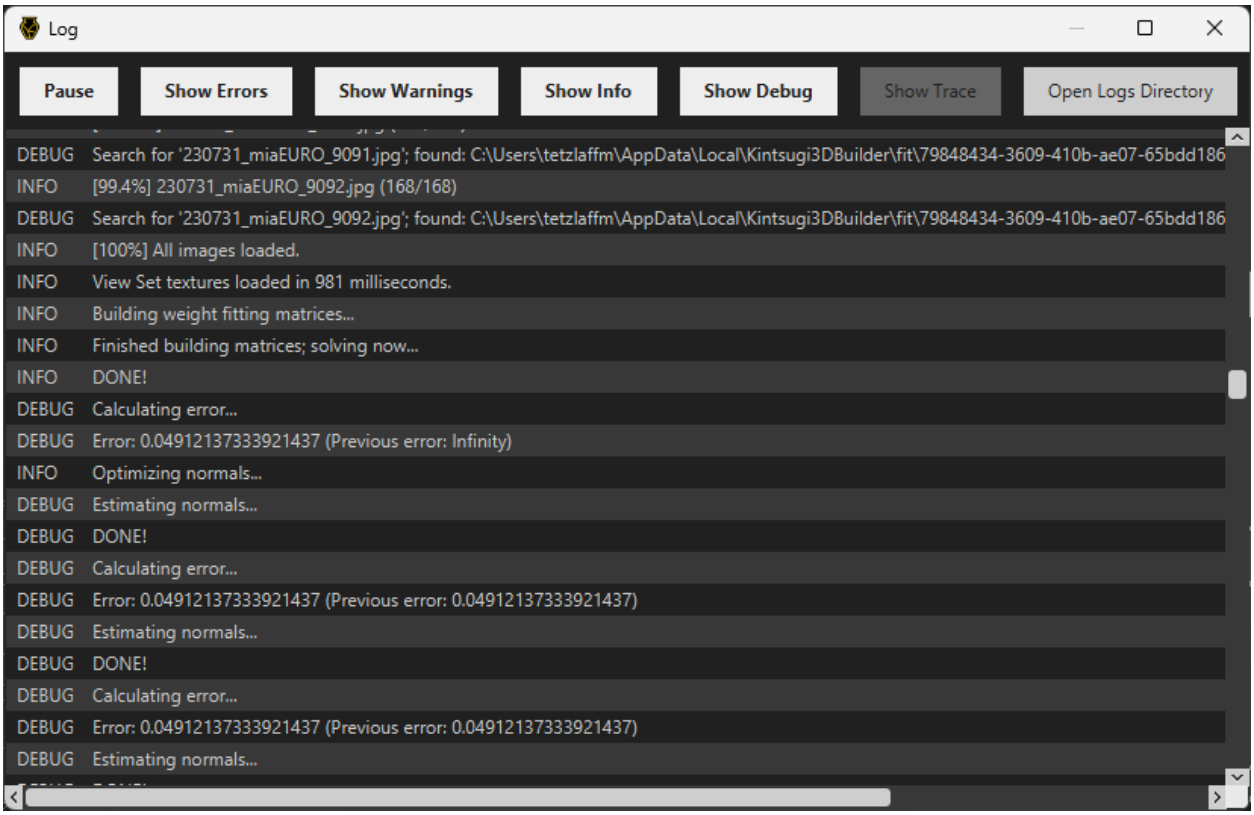
For more detailed information about the progress of the optimization process, a console log window is available under "Help":

---

[9] Tetzlaff, M. "High-Fidelity Specular SVBRDF Acquisition from Flash Photographs." *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 30, no. 4. 2024, pp. 1885-1896. https://ieeexplore.ieee.org/abstract/document/10012127

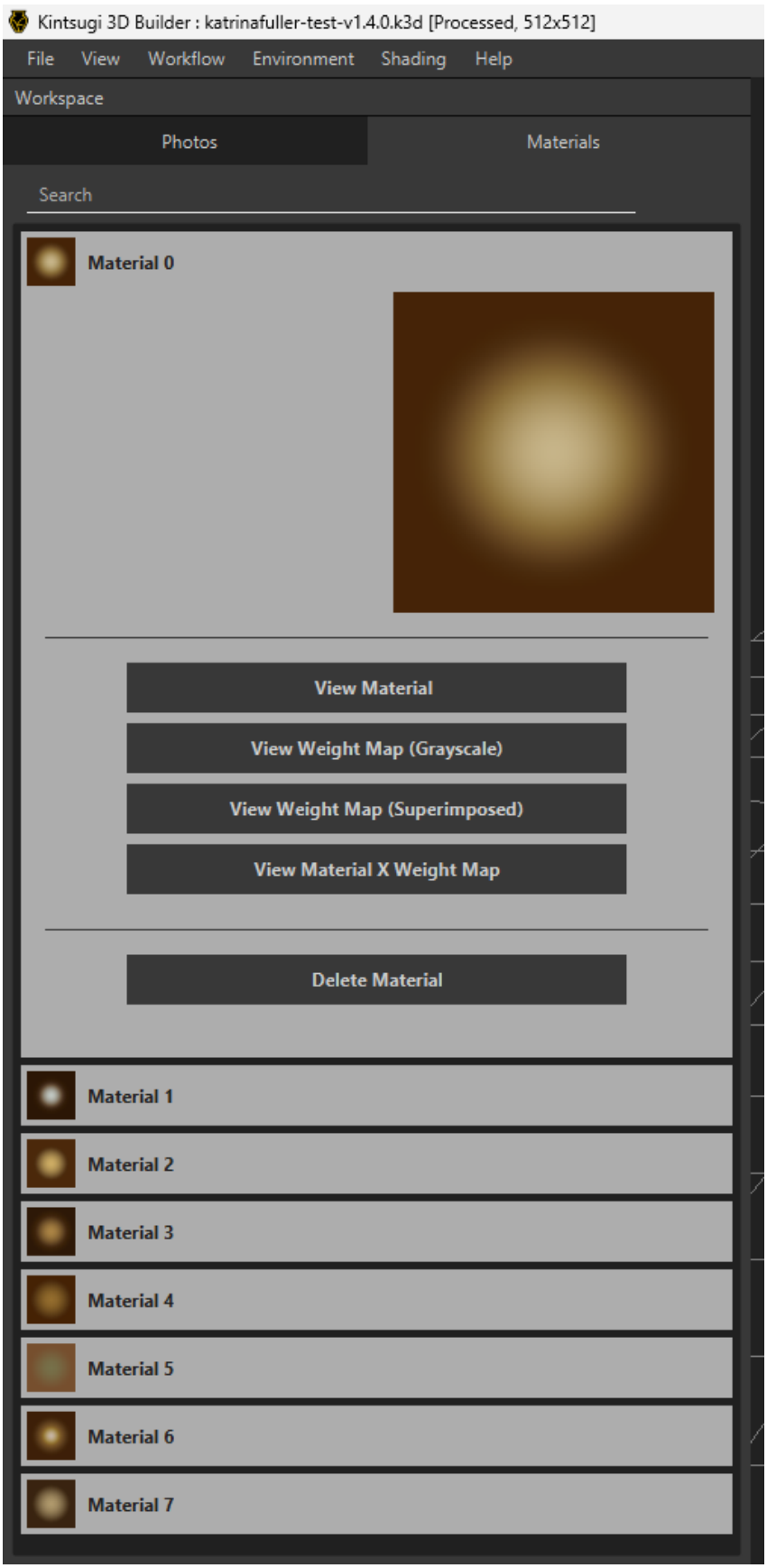
[10] Nam, G., Lee, J. H., Gutierrez, D., and Kim, M. H. "Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography." *ACM Transatctions on Graphics,* vol. 37, no. 6. 2018, pp. 267:1-267:12. https://dl.acm.org/doi/abs/10.1145/3272127.3275017





Basis material palette

After processing textures, two things will happen. First, the active shader will automatically change to “Material (basis)” to show the results of texture processing – for more details, refer to the section on shading modes. Second, the “Materials” tab in the Workspace panel will show a list of the basis materials that were generated. This is sometimes referred to as the “material palette.”

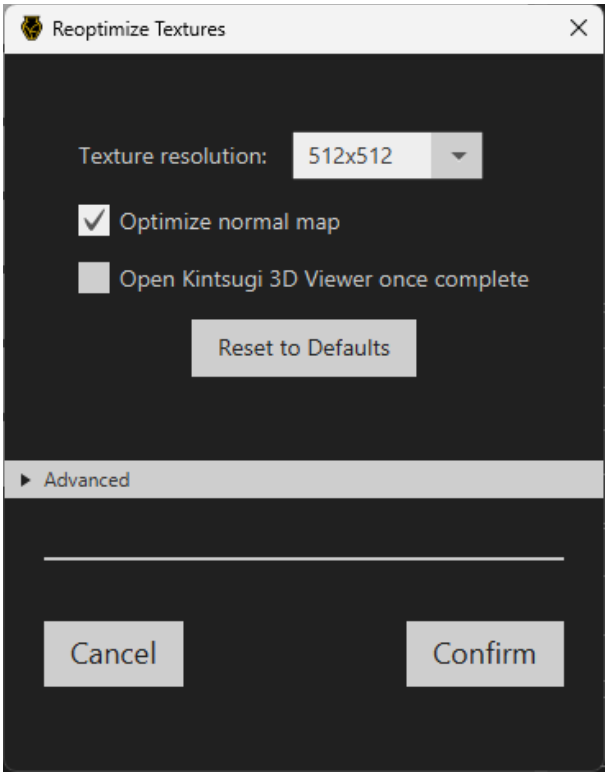

Kintsugi 3D Builder : katrinafuller-test-v1.4.0.k3d [Processed, 512x512]
File
View
Workflow
Environment
Shading
Help
Workspace
Photos
Materials
Search
Material 0
View Material
View Weight Map (Grayscale)
View Weight Map (Superimposed)
View Material X Weight Map
Delete Material
Material 1
Material 2
Material 3
Material 4
Material 5
Material 6
Material 7

For each basis material, a card can be expanded which contains shortcuts to the four main visualization shaders for that material (see more about these shaders in a later section on shading modes). In addition, each basis material can be deleted to remove its impact on the object's appearance. **Once deleted, a basis material can no longer be recovered.**

Typically, just deleting a basis material will leave gaps in the object's appearance. If the goal is for other basis materials to fill in for a basis material that was deleted, the "reoptimize textures" task will need to be run.

Reoptimize Textures

The reoptimize textures task allows you to run just the last stage of the process textures method again (optimize high-res weight maps and normal map). This can be useful for refreshing these textures, for instance, after removing a material from the material palette. The basis materials will not be regenerated by this step, which saves time and preserves any manual modifications to the basis material palette.

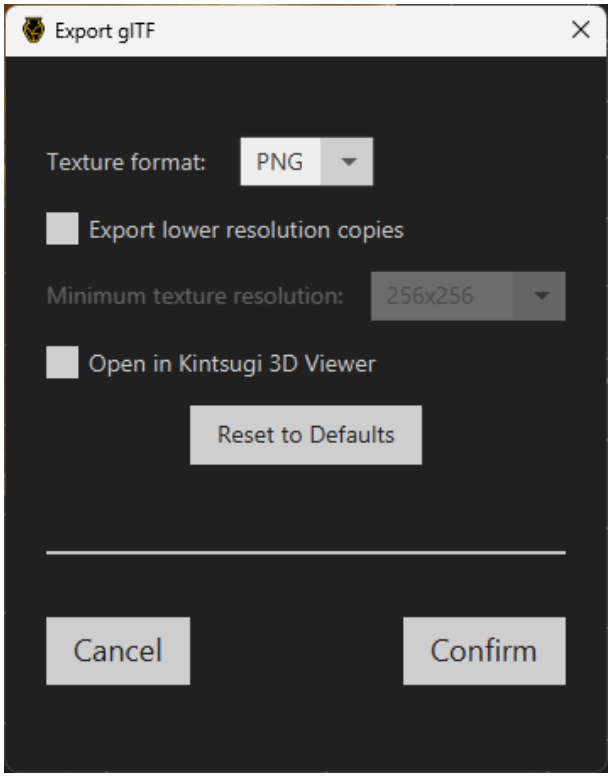


The options available for this step are a subset of the options available when processing textures originally. Notably, the texture resolution may be changed (although this may require a time-consuming step to regenerate the texture-space cache) and normal map optimization can be toggled.

Export glTF

The “export glTF” task allows you to save a copy the model with processed textures for use in external programs. Kintsugi 3D extends the standard glTF format with additional data that can be used by Kintsugi 3D Viewer.

In most programs, when the glTF model exported by Kintsugi 3D Builder is imported, the “metallicity” representation (described below, under “Shading modes”) will be used. With some manual configuration, the “reflectivity” representation can be used in Sketchfab (as described below, under “Uploading to Sketchfab”). Kintsugi 3D Viewer is the only viewer at the time of writing that supports the “basis” material representation.



Options available when exporting include:

- Texture format: PNG, JPEG, and TIFF. Because JPEG only supports three color channels (no alpha), if JPEG is selected, another format (PNG) will be used for exporting packed weight maps.
- Export lower resolution copies: Generates lower resolution versions of the textures, often useful for web distribution.
    - Minimum texture resolution: The lowest resolution generated when “export lower resolution copies” is selected.

If Kintsugi 3D Viewer is installed alongside Kintsugi 3D Builder, the “Open Kintsugi 3D Viewer once complete” option can be used to automatically open the exported model in Kintsugi 3D Viewer.

Launch Kintsugi 3D Viewer

This menu option can be used to launch Kintsugi 3D Viewer, automatically opening the model contained within the current project to see what the model looks like in that environment.

Environment menu

Relighting

Enable this option to turn on relighting for the object. By default, Kintsugi 3D Builder maintains the “flash-on-camera” configuration for virtual lighting, resulting in an entirely viewpoint-dependent algorithm for rendering the 3D model. To simulate arbitrary lighting, the “relighting” toggle can be enabled, which may be particularly useful for previewing results after processing textures. It should be noted that the “image-based” shading modes will incur a significant performance hit when turning on relighting.

Show light widgets

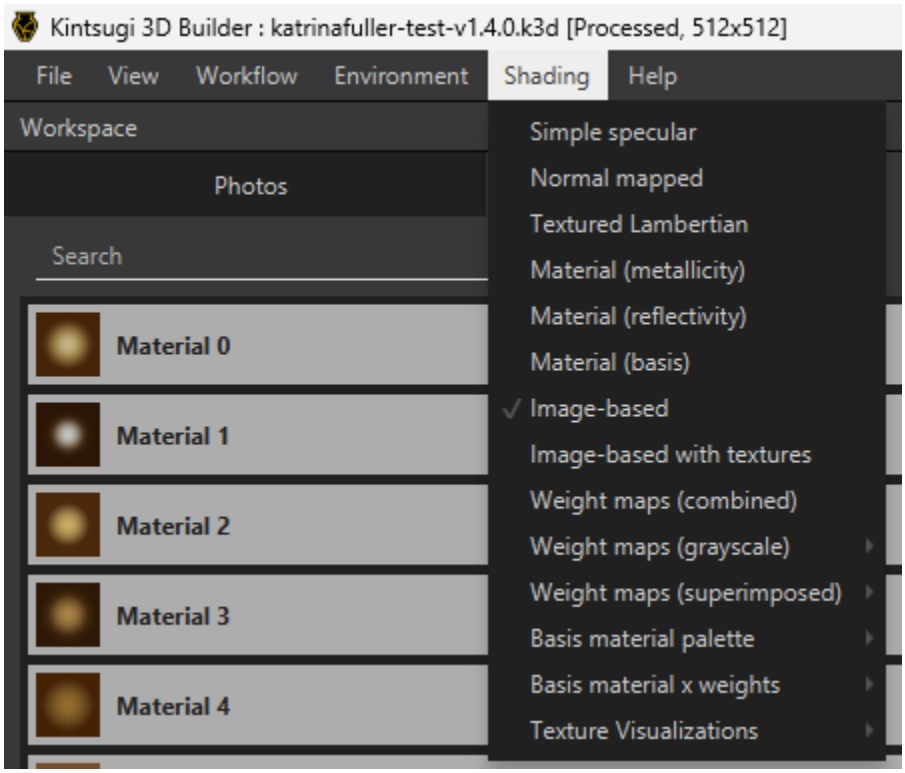
Enable this option to cause light widgets to be rendered that allow for direct manipulation of the light sources from within the 3D viewport. This has no effect if relighting is disabled.

Scene window

Opens the scene window, described later in this document.

Shading modes

Kintsugi 3D Builder contains several built-in shading models for different purposes.



Simple Specular

This will render the object with a simple shiny gray “plastic” material so that characteristics of the geometry can be clearly seen.

Normal Mapped

This will render the object with the same gray plastic material as Simple Specular, but using the current normal map, either defined in the material (.mtl) file for the imported 3D model, or optimized using the “process textures” task. If a normal map has not been defined, then this behaves the same as Simple Specular. For objects imported directly from photogrammetry that contained a baked normal map, prior to running “process textures,” this will be the normal map that the photogrammetry software generated.

Textured Lambertian

This will render the object as a non-specular material (like clay), using the current diffuse color texture, either defined in the material (.mtl) file for the imported 3D model, or optimized using the “process textures” task. For objects imported directly from photogrammetry, prior to running “process textures,” this will be the texture that the photogrammetry software generated.

Material (metallicity)

This will render the object using a monochrome “metallicity” texture in combination with a roughness texture. These two monochrome textures can be packed (optionally accompanied by an occlusion map generated by photogrammetry or 3D modelling software) into an “ORM” (occlusion / roughness / metallicity) texture, which is supported by the glTF standard, and thus is usable in many game engines and online viewers. Kintsugi 3D Builder compresses the basis materials and weight textures down to this more universal representation as part of the “process textures” task. Because of this, the “metallicity” representation is generally less accurate than “basis” or “reflectivity,” but may still be acceptable for certain types of materials. Prior to running “process textures,” this shading mode is disabled.

Material (reflectivity)

This will render the object using an RGB specular reflectivity texture in combination with a roughness texture. Like the “metallicity” representation, these textures are a compression of the basis materials and weight textures, also produced by the “process textures” task, but have a higher fidelity than the “metallicity” representation due to the ability to represent specular reflectivity as an arbitrary RGB color at each pixel. The quality of this representation is often close to the quality level of the “basis” material representation. It can be uploaded to Sketchfab using its “specular” workflow as discussed in a later section. Prior to running “process textures,” this shading mode is disabled.

Material (basis)

This will render the object using the actual optimized basis materials produced by the “process textures” task. This is the highest quality result, and can be used by Kintsugi 3D Viewer. Prior to running “process textures,” this shading mode is disabled.

Image-based

This will render the object with image-based rendering as described in prior publications.[11] [12] [13] [14] This is the default rendering mode and is useful for visualizing the projections of the “preview resolution” photographs onto the 3D model. It can also be used as a rendering mode in its own right – as it uses the original photographs, it is even closer to the original object in some sense. However, it has much higher resource usage than the material shading modes and as such cannot use the full resolution of the original images. It also has less relighting flexibility, which

[11] Tetzlaff, M. and Meyer, G. “Image-Based Relighting using Environment Maps. *IS&T Archiving Conference*. 2017, 23-27.

[12] Tetzlaff, M. and Meyer, G. “Using Flash Photography and Image-Based Rendering to Document Cultural Heritage Artifacts. *Eurographics Workshop on Graphics and Cultural Heritage*. 2016, 137-146.

[13] Berrier, S.; Tetzlaff, M.; Ludwig, M.; Meyer, G. “Improved Appearance Rendering for Photogrammetrically Acquired 3D models.” *Digital Heritage International Congress*. 2015, 255-262.

[14] Buehler, C., Bosse, M., McMillan, L., Gortler, S., & Cohen, M. “Unstructured Lumigraph Rendering.” *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 2001, 425-432.

may cause visual artifacts for very shiny objects when animating lights or attempting to simulate environment-based lighting, i.e. an HDRI.

Image-based with textures

This will render the object with image-based rendering, using the “reflectivity” representation of the material to refine the results when “relighting” is enabled (more accurate Fresnel effect and geometric attenuation). Prior to running “process textures,” this shading mode is disabled.

Weight maps (combined)

This visualization shows the first eight weight maps used to “paint” the optimized materials from the basis material palette onto the 3D model. Each map is given a distinct color for easy differentiation.

Weight maps (grayscale)

Each of these visualizations shows an individual weight map used to “paint” a specific material from the optimized basis material palette onto the 3D model.

Weight maps (superimposed)

Each of these visualizations shows an individual weight map used to “paint” a specific material from the optimized basis material palette onto the 3D model, rendered over the “Material (basis)” shading mode for context.

Basis material palette

Each of these visualizations shows the entire 3D model rendered homogeneously with a single material from the optimized basis material palette.

Basis material x weights

Each of these visualizations shows a specific material from the optimized basis material palette, only in locations on the object where its corresponding weight map indicates that it is to be used.

Texture visualizations

Each of these options allows a single texture map to be rendered on the 3D model in isolation. Texture maps that can be selected for this include:

- Normal map
- Albedo map – used primarily in the metallicity material representation to represent color.
- Roughness map – used primarily in the metallicity and reflectivity material representations to approximate the microfacet distribution.
- Metallicity map – used exclusively by the metallicity material representation to derive specular and diffuse color from a single albedo color.
- Diffuse color map – used primarily in the reflectivity and basis material representations to represent diffuse (Lambertian) reflection.

- Specular color map – used primarily in the reflectivity material representation to represent the color of specular highlights and reflections.

In addition, an “error map” is available that shows where on the model the highest error was observed after optimizing the materials.

Help

Documentation

A link to this documentation can be found under the “Help” menu.

View Log

Open the console log to see in-depth information about Kintsugi 3D Builder’s processing steps and other details.

3D viewport navigation

The 3D viewport is where the mouse and keyboard are used to produce direct manipulation of the camera, environment, lights, and object. Each mouse button and key has a different meaning in the context of the 3D viewport. **In general, the left mouse button performs rotation actions, while the right mouse button performs move and scale actions.** A summary of all the keyboard and mouse controls is given below. The documentation which follows will then describe in detail the 3D viewport controls for each combination of mouse button and modifier keys (Shift, Ctrl, Alt).

Mouse and keyboard control summary / quick reference

Camera orbit **Camera pan**	**LMB (left mouse button)** **RMB (right mouse button)**
Look at point	**Ctrl-Shift-RMB**
Camera twist **Camera dolly**	**Alt-LMB** **Alt-RMB**
Rotate environment	**Shift-LMB**
Focal length	**Shift-RMB**
Object rotation **Object translation** **Object twist**	**Ctrl-LMB** **Ctrl-RMB** **Ctrl-Alt-LMB**
Environment brightness **Background brightness**	**Arrow keys** **Shift + Arrow keys**
Toggle visible lights **Toggle light manipulators**	**L** **Ctrl-L**

Vantage point controls

Clicking and dragging without any keys pressed modifies the vantage point of the camera, the point and direction in space from which it is looking at the object. The left and right mouse buttons have two different functions that modify the vantage point in different ways:

Camera orbit

To rotate the camera around the object being rendered in order to see it from different vantage points, simply **click and drag using the left mouse button.** The camera always rotates around a point in the center of the screen. If you go over the top of the object with vertical rotation, you may observe that the camera “flips around” in order to stay right-side up.

Camera pan

To pan (translate) the camera and change the point at which the camera is looking, **click and drag using the right mouse button.** You can also **hold down Ctrl+Shift and right-click** on a point on the object to place that point in the center of the screen.

Alt key - Camera axis controls

Clicking and dragging while holding down the Alt key provides a mechanism for manipulating the camera on a fixed axis. The left and right mouse buttons have provide two different functions that each operate on a fixed camera axis:

Camera twist

Clicking and dragging side-to-side using the left mouse button with the Alt key pressed changes the “twist” or orientation of the camera, effectively rotating the entire screen.

Camera dolly

Clicking and dragging up and down using the right mouse button with the Alt key pressed moves the camera towards or away from the object being rendered. This effectively makes the object appear bigger on the screen.

Ctrl key - Object controls

The Ctrl key is generally used for actions that apply to the active object pose rather than to the active camera. This means that the camera, the environment and the lights will all remain fixed with respect to each other while the object is manipulated. The controls with the Ctrl key pressed are otherwise approximately analogous to the camera controls:

Object rotation

To rotate the object, **click and drag using the left mouse button with the Ctrl key pressed.** The object always rotates around a center point, which by default is the centroid of the triangle mesh.

Object translation

To change the center point about which the object rotates, you can move the object by **clicking and dragging using the right mouse button with the Ctrl key pressed.**

Object twist

To rotate the object in the screen plane, click and drag using the left mouse button with the Ctrl and Alt keys pressed simultaneously.

Shift key - Miscellaneous controls

The Shift key is used for two miscellaneous controls:

Rotate environment

Clicking and dragging using the left mouse button with the Shift key pressed rotates the environment. The environment can only be rotated horizontally.

Focal length

Clicking and dragging up and down using the right mouse button with the Shift key pressed changes the focal length of the camera (which determines its field of view). Like the camera dolly tool, this will cause the object to appear bigger or smaller in the screen. Unlike the dolly tool, however, this will not change the perspective distortion of the object as the camera is simply viewing more or less of the scene from a fixed vantage point rather than physically moving closer or further away from the object.

Key controls

There are also several key controls that can be used independent of the mouse:

Environment brightness

The **arrow keys** change the brightness of the environment. This affects the total amount of light the environment casts on the object, as well as the appearance of the environment in the background if no backplate or background color is specified. The **up** and **down** keys change the brightness by a large increment, while the **left** and **right** keys change it by a small increment.

Background brightness

While **holding down the shift key, the arrow keys** change the brightness of the background rather than the environment. This will affect whatever is being displayed behind the object, whether it is an environment map, a backplate, or a solid color. However, it does not affect how much light the environment casts on the object. As a result this will potentially make the rendering no longer physically consistent when the environment map is displayed in the background, but may still be helpful for producing an aesthetically pleasing result. The **up** and **down** keys change the brightness by a large increment, while the **left** and **right** keys change it by a small increment.

Toggle lights

Pressing the **L key** toggles whether or not point light sources are rendered visually in the scene along with the object.

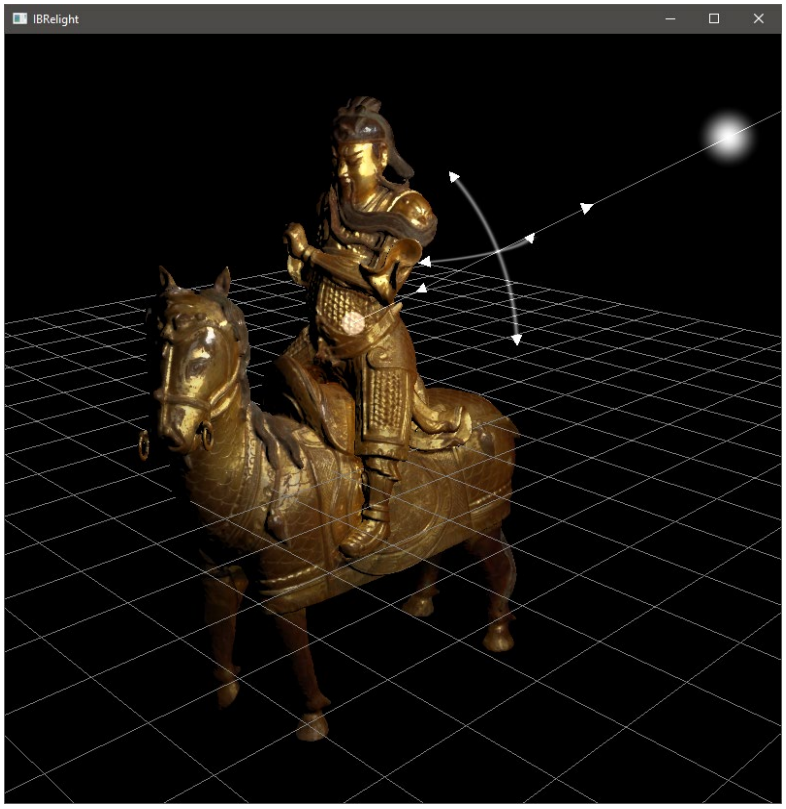
Toggle light widgets

Pressing the **L key while holding down the Ctrl key** toggles whether or not manipulator widgets are displayed to enable the movement of light sources in the 3D viewport.

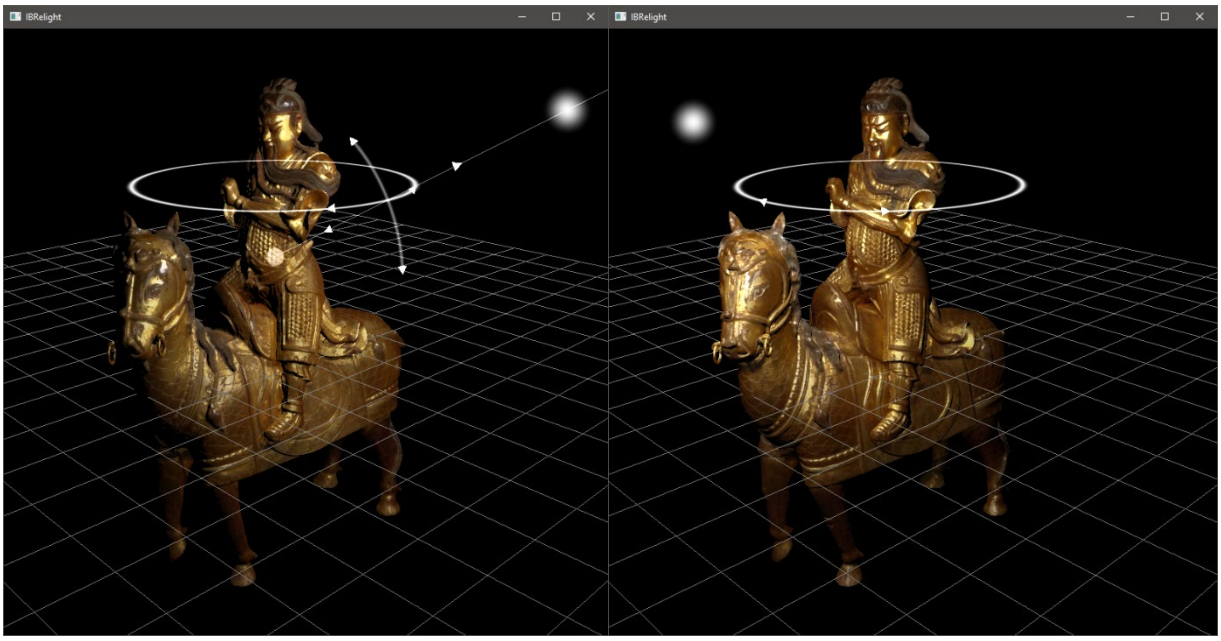
Light manipulation

Light manipulation is performed via a 3D widget that is rendered in the 3D viewport. Each point light source has its own widget, which consists of three double arrows that intersect at a point that is in the same direction as the light. There is also another point to which this widget is

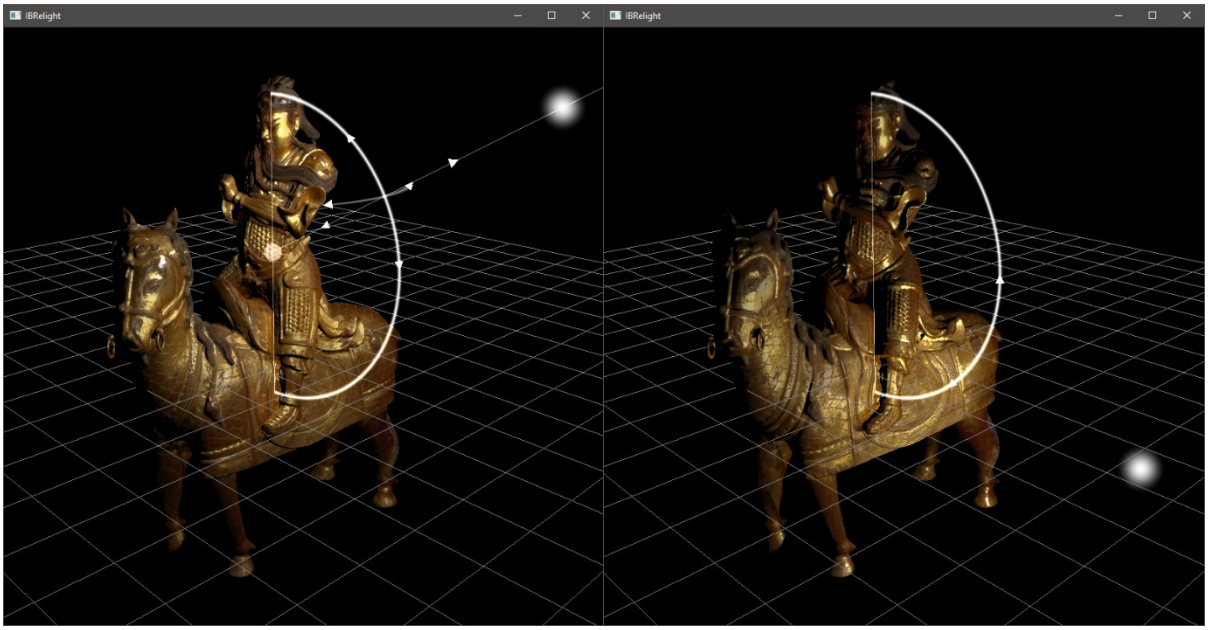
connected, rendered as a circle, which controls the center point which the light source rotates around.



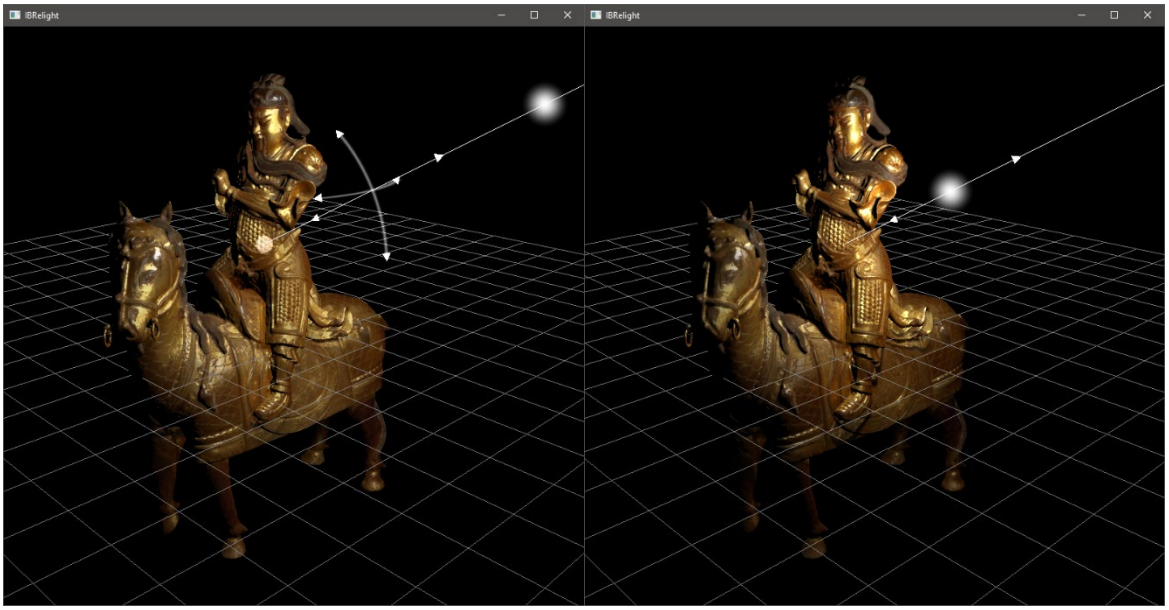
Two of the double arrows are curved. The horizontal arc is used to manipulate the azimuth of the light source. Hovering over one of the rotation arcs will show a complete circle around the center point. Clicking on the arc will cause the other arrows to disappear, and dragging will move the light source around this circle.



The vertical arc is used to manipulate the inclination of the light source. Hovering over this arc will show a semicircle from the top of the object to the bottom of the object. Clicking on the arc and dragging will move the light source up and down along this semicircle.

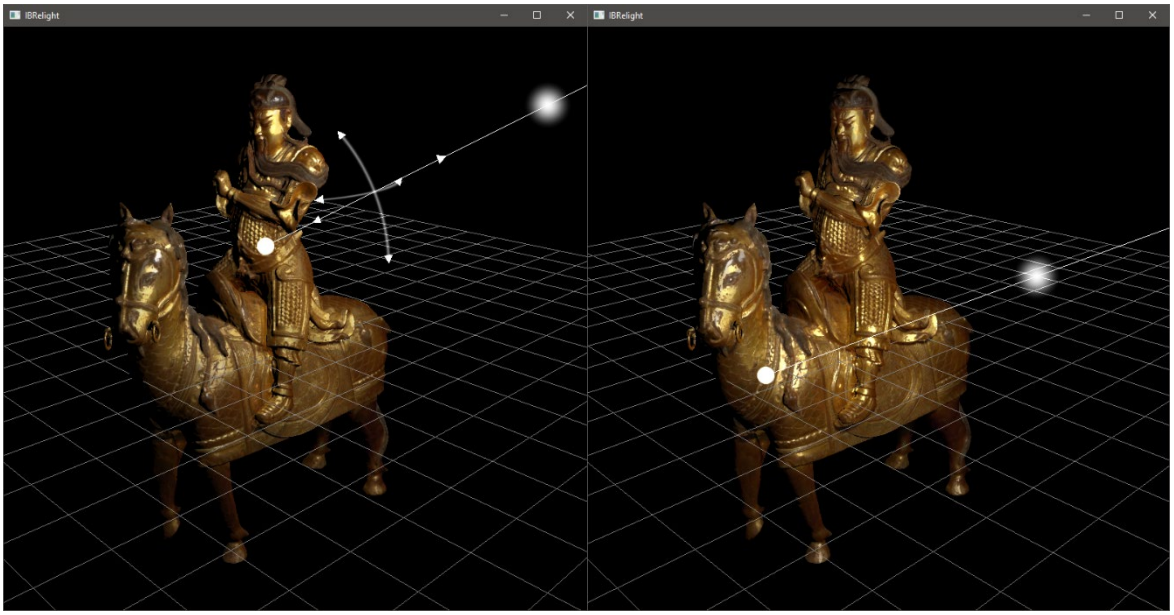


The third double arrow lies along the line between the light source and the center point. Hovering over this line or the arrowheads on the line will highlight it, and clicking and dragging will move the light source closer to or further away from the object.



The small circle to which the light manipulator is attached is the center point, which affects the behavior of all the other manipulations. Hovering over this circle will highlight both it and the line to the light source, and clicking and dragging will move both the center point and the

attached light source. The center point can only be moved to a point on the surface of the object being rendered using the 3D viewport.



Scene window

While the 3D viewport provides controls for modifying the camera, environment, lights and the pose of the object being rendered, the Scene Window provides the additional functionality of being able to create different “presets” for each of these and and then quickly switch through different viewing and illumination conditions. It also provides a mechanism for fine tuning the properties of each camera, environment, light, and pose by directly editing their numerical values.

Camera Panel

Camera list

The Camera Panel contains a list of the saved cameras in the scene. When a camera is selected in this list, it becomes the active viewport in the 3D viewport.

For a selected camera, the following options are available:

New: Duplicates the selected camera and adds it as a new camera in the list. The new camera is automatically selected and becomes the active viewport.

Rename: Assign the camera a new name.

Lock / Unlock: When a camera is locked, the indication “(L)” is displayed before its name. While locked, a camera cannot be modified again until it is unlocked.

Delete: Permanently deletes the camera from the list.

The rest of the Camera panel is used to modify properties of the active camera.

Look at point

This modifies the point in 3D space where the camera is directed (the center of the viewport). The 3D grid can be turned on under the Viewport menu to get a sense of what these units mean. The grid lies in the XY plane, and each square of the grid is 0.1 unit by 0.1 unit in size.

Position and Orientation

These settings affect in what direction the camera is looking and how far away from the object it is.

Azimuth: The horizontal orientation of the camera.

Inclination: The vertical orientation of the camera.

Distance: The distance from the camera to the “look at” point.

Twist: Affects what direction is “up” for the camera. A twist of +180 or -180 degrees will turn the camera completely upside down.

Intrinsic Properties

Field of view and focal length: Affects how much of the scene the camera can "see." These two properties are two different ways of specifying this camera characteristic. A larger field of view, or a smaller focal length, will allow the camera to see more, but with more distortion near the edges of the screen. A smaller field of view, or a larger focal length, will allow the camera to see less, but with less distortion.

Environment Panel

Environment list

The Environment Panel contains a list of the saved environments in the scene. When an environment is selected in this list, it becomes the active environment in the 3D viewport.

For a selected environment, the following options are available:

New: Duplicates the selected environment and adds it as a new environment in the list. The new environment is automatically selected and becomes the active environment in the 3D viewport.

Rename: Assign the environment a new name.

Lock / Unlock: When an environment is locked, the indication "(L)" is displayed before its name. While locked, all of the other options in the panel for this environment are disabled until it is unlocked.

Delete: Permanently deletes the environment from the list.

The rest of the Environment Panel is used to modify properties of the active environment.

Lighting

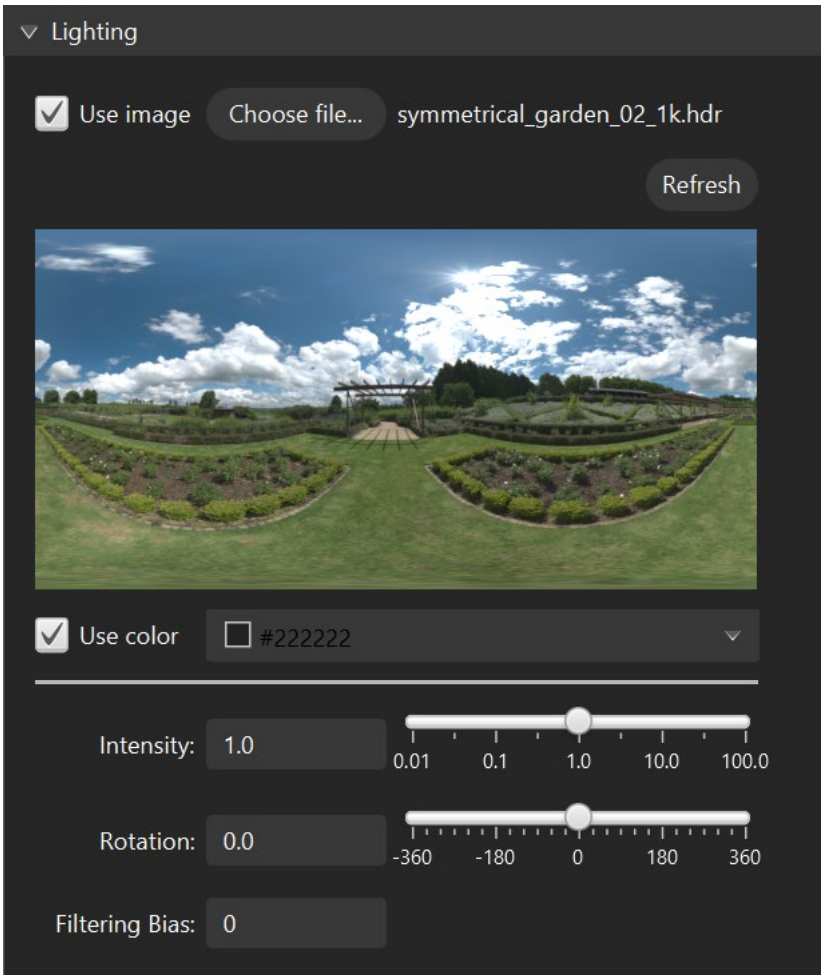
This section defines how the environment illuminates the object. This is determined by the combination of whether "Use image" and "Use color" are checked within this section. There is also an intensity slider to control the brightness of the environment, and a rotation slider to control its orientation.

Use image and use color: The environment map will be used to illuminate the object. The color specified will be used to tint the environment map.

Use image, don't use color: The environment map will be used to illuminate the object without tinting.

Don't use image, use color: The specified color will be used as an ambient light for the scene. The ambient light intensity can be scaled using the intensity slider for the environment.

Don't use image or color: No environment illumination will be applied to the object.



Loading an environment map: To load an environment map, click the button labelled "Choose file…" Currently, only Radiance HDR environment maps are supported. If you have an environment map in another format, you will need to use an image manipulation utility like ImageMagick to convert it into Radiance HDR. Kintsugi 3D Builder assumes that an environment map is stored as a panorama unless its name ends with "_zvc.hdr", in which case it interprets it as a cubemap in a "cross format."

> **Note:** If you intend to switch quickly between environment maps, you may want to reduce their resolution to minimize loading time when switching. Although you will see a reduction in the quality of the background at a lower resolution, Kintsugi 3D Builder can accurately illuminate the object from environments at a very low resolution (i.e. 256x128 pixels).

Refresh: The refresh button will reload the environment map from the disk, and can be used to quickly update the environment after editing it using another software program.

Intensity: The intensity slider in the Lighting panel controls the brightness of the environment map. This affects the total amount of light the environment casts on the object, as well as the appearance of the environment in the background if no backplate or background color is specified.

Rotation: The rotation slider in the Lighting panel controls the orientation of the environment map.

Filtering Bias: This parameter adjusts the amount of filtering that is applied to the environment map when using it for relighting with an “image-based” shading mode. This setting has no effect for other shading modes. Values above zero will potentially reduce directional aliasing effects that are perceptible when interacting with the 3D model, but will also cause the object to be illuminated more diffusely. Values below zero will potentially reduce the diffuseness of the illumination, but may increase directional aliasing effects.

Background

This section determines what is displayed behind the object. This is determined by the combination of whether “Use image” and “Use color” are checked within this section.

Use image and use color: The specified image will be used as a background and will be tinted by the specified color.

Use image, don’t use color: The specified image will be used as a background without tinting.

Don’t use image, use color: The specified color will be used as a background color.

Don’t use image or color: The background will be whatever is set under the Environment map section.

Loading a backplate image: To load a backplate image, click the button labelled “Choose file.” Currently, only BMP, JPEG, PNG, and GIF files are supported.

Refresh: The refresh button will reload the backplate image from the disk, and can be used to quickly update the backplate after editing it using another software program.

Intensity: The intensity slider in the Background panel controls the brightness of the background rather than the environment. This will affect whatever is being displayed behind the object, whether it is an environment map, a backplate, or a solid color. However, it does not affect how much light the environment casts on the object. As a result this will potentially make the rendering no longer physically consistent when the environment map is displayed in the background, but may still be helpful for producing an aesthetically pleasing result.

Ground plane

This section determines whether a ground plane is rendered underneath the object in 3D space.

Use ground plane: Toggle whether the ground plane is visible.

Color: The color of the ground plane.

Height: The location of the ground plane on the y-axis.

Size: The scale of the ground plane on the x- and z-axes.

Lights Panel

Light groups

Kintsugi 3D Builder maintains a list of saved light groups. A light group consists of up to four point light sources, each of which is defined by a position, orientation, color, and intensity. Clicking on a light group selects it and makes it the active lighting in the 3D viewport. To create a new, empty light group, click the button labelled “New” under the word “Group” in the “Light Groups” section.

For each light group, the following options are available:

Rename: Assign the light group a new name. The name must be unique across all saved cameras and light set. The name field will automatically be populated with a unique ID number when a new light set is created.

Lock/Unlock: When a light group is locked, the indication “(L)” is displayed before its name. The lights in a locked light group cannot be modified until the group is unlocked.

Delete: Removes the light group from the list permanently.

Lights in a light group

For each light group, up to four light sources will be available. Each light in a group is indicated by an “X” in the table of light groups. Clicking on this X will select the light so that you can perform actions on it or edit its properties.

The following actions are available for a light source:

New: Duplicates the currently selected light source, if there are less than four lights in the active light group. The light will be shifted slightly from the original so that it is not directly on top of the original, but will be identical in all other respects.

Lock/Unlock: Locked lights have their “X” replaced with an “L” in the light group table. While locked, a light source cannot be modified again until it is unlocked.

Delete: Removes the light source from the light group permanently.

The rest of the Lights Panel is used to modify properties of the currently selected light source.

Target point

This modifies the point in 3D space where the light is directed. This will be the point in space that the light rotates around when manipulating it, and the point that it moves towards or away from when modifying its “distance” attribute. When rendering with shadows, it will also affect the direction in which it casts light. The 3D grid can be turned on under the Viewport menu to get a sense of what these units mean. The grid lies in the XY plane, and each square of the grid is 0.1 unit by 0.1 unit in size.

Position

These settings affect the position of the light source in space.

Azimuth: The horizontal orientation of the light source relative to its target point.

Inclination: The vertical orientation of the light source relative to its target point.

Distance: The distance from the light source to the target point.

Light Properties

Intensity: How bright the light source is.

Color: The color of the light source.

Spot Size: Controls the size of the "spot" cast by the light by specifying the angular extent to which it casts light, in degrees.

Taper: Controls how gradually the spotlight transitions from full intensity to no intensity. A value of 0.0 will cause a sharp cutoff, while a value of 1.0 will cause the intensity to gradually decrease over the entire spot area.

Object Poses Panel

Pose list

The Object Poses Panel contains a list of the saved poses in the scene. When a pose is selected in this list, it becomes the active pose in the 3D viewport.

For a selected pose, the following options are available:

New: Duplicates the selected pose and adds it as a new pose in the list. The new pose is automatically selected and becomes the active pose.

Rename: Assign the pose a new name.

Lock / Unlock: When a pose is locked, the indication "(L)" is displayed before its name. While locked, a pose cannot be modified again until it is unlocked.

Delete: Permanently deletes the pose from the list.

The rest of the Object Poses panel is used to modify properties of the active pose.

Center

This modifies the point in 3D space that is used as the center of the object when rotating it. The 3D grid can be turned on under the Viewport menu to get a sense of what these units mean. The grid lies in the XY plane, and each square of the grid is 0.1 unit by 0.1 unit in size.

Rotation

These settings affect the orientation of the object.

Rotate Y: This rotation is applied first. It rotates the object around the vertical axis of the world coordinate system. In some sense, it is analogous to the "azimuth" property of cameras and lights.

Rotate X: This rotation is applied second. It rotates the object forwards and backwards, as if "tipping it over" in a direction defined by the first rotation (Rotate Y). In some sense, it is analogous to the "inclination" property of cameras and lights.

Rotate Z: This rotation is applied third. It twists the object along an axis determined by "Rotate Y and Rotate X" and is analogous to the "twist" property of cameras.

Scale

Scale: This is a uniform scale applied to the object to make it bigger or smaller.

Settings

Cache settings

Clear Cache: This will delete all files in the cache for all Kintsugi 3D Builder projects. This includes both the preview images generated when creating a project, and the texture-space chunks used for texture processing. All files in the cache can be regenerated when needed from the original source photographs (assuming those files are still available).

Lighting settings

Fresnel effect: Checking this will enable the Fresnel effect, which causes highlights to get brighter and more white when the object is lit from behind. This feature works best after completing the "process textures" task.

Shadows: Enable this option to cause the object to cast shadows on itself.

Physically-based masking/shadowing: Use a physically-based geometric attenuation (microfacet masking/shadowing) equation (currently, this is the V-cavity model). This will improve the accuracy of the object's appearance when lit from the side or from behind and can be used for either metallic objects or glossy objects with an accurate diffuse texture map. For non-metallic objects, if the only texture map available is the one computed by Metashape / Reality Capture, this should be left unchecked.

Relighting: Alias for the "Relighting" toggle in the Environment menu.

Show light widgets: Alias for the "Show light widgets" toggle in the Environment menu.

Photo projection settings

Limit blended views per pixel: When disabled, all preview photographs are blended as described by Berrier et al. [15], rather than the "nearest neighbor" selection described by Buehler et al. [16] This should not affect the results of high-resolution texture processing.

Weight exponent: The 'alpha' value from the weight function used when "Limit blended views per pixel" is disabled. Increasing this causes more views to be used at each point but will introduce blurriness. Decreasing it will force the views to have a negligible effect sooner and results in sharper images. Has no effect if "Limit blended views per pixel" is enabled, or the shading mode is set to anything other than "image-based" or "image-based with textures." This should not affect the results of high-resolution texture processing.

> **Isotropy factor**: When set to a value greater than zero, this parameter causes photographs that are far away from a desired but possibly non-existent photograph and

[15] Berrier, S.; Tetzlaff, M.; Ludwig, M.; Meyer, G. "Improved Appearance Rendering for Photogrammetrically Acquired 3D models." *Digital Heritage International Congress*. 2015, 255-262.

[16] Buehler, C., Bosse, M., McMillan, L., Gortler, S., & Cohen, M. "Unstructured Lumigraph Rendering." *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 2001, 425-432.

which are expected to exhibit similar appearance based on certain physical assumptions to have a higher blending weight when rendering. This factor should always be set to a value less than one or undesirable artifacts may appear. This feature is somewhat experimental. Has no effect if “Limit blended views per pixel” is enabled, or the shading mode is set to anything other than “image-based” or “image-based with textures.” This should not affect the results of high-resolution texture processing.

System memory settings

Limit memory usage: This can be enabled to limit the amount of memory used by Kintsugi 3D Builder. This can prevent system crashes for systems with low RAM, but may increase the likelihood for Kintsugi 3D Builder to crash if more memory than the specified cap is required to perform a computational task.

Visual settings

Preview Image Width and Height: Sets the resolution of the “preview” images that are used for the “image-based” shading modes. Has no effect if the shading mode is set to anything other than “image-based” or “image-based with textures.” Changing this option will not have an effect until a new project is loaded, and may result in a longer load time the next time each project is loaded as the preview images are regenerated from the full-resolution source images.

Image Compression (recommended on): Apply texture compression to the preview images as they are being loaded. This will reduce memory usage significantly but may result in some visual artifacts. This should not affect the results of high-resolution texture processing. Has no effect if the shading mode is set to anything other than “image-based” or “image-based with textures.” Changing this option will not have an effect until a new project is loaded.

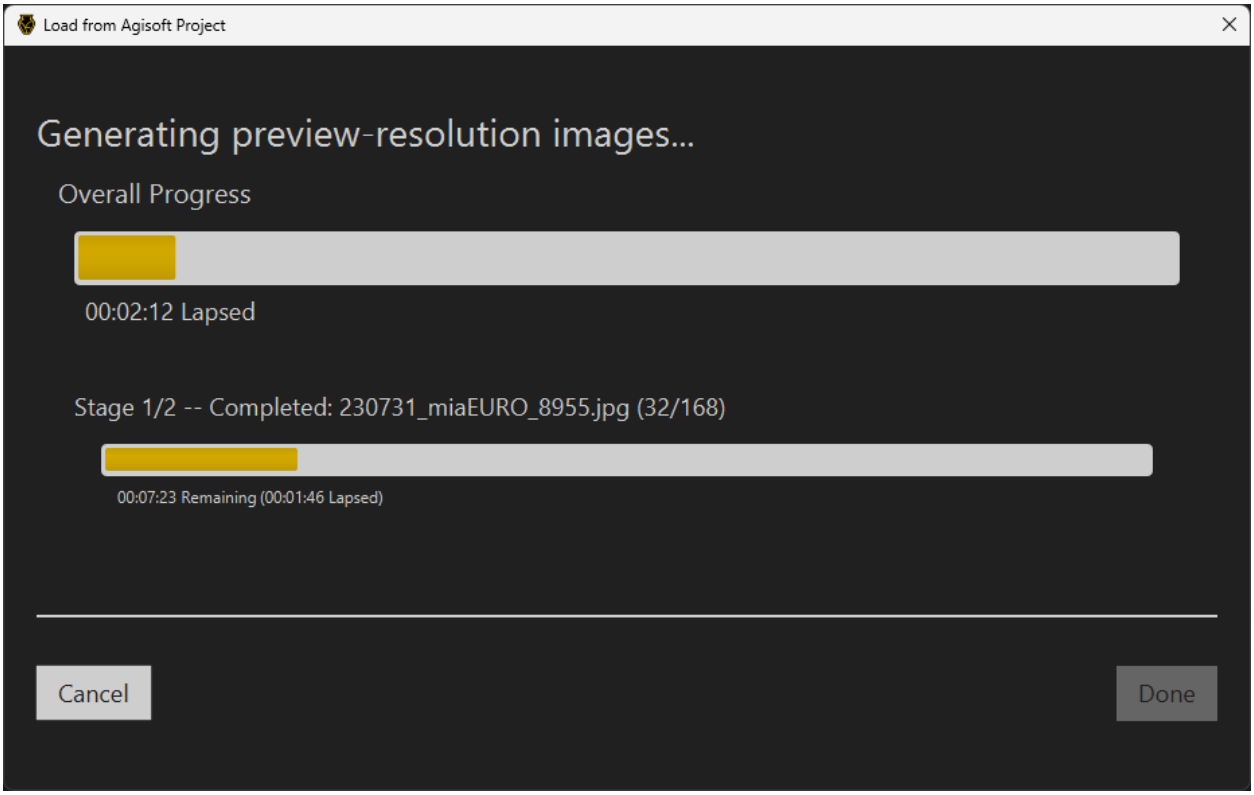
Preload Visibility and Shadow Testing (recommended on): If the model has some parts that could cast a shadow on itself or otherwise obstruct the surface then this option will generate “depth maps” to help it render properly. If the model does not have these features you may safely uncheck this box and reduce memory usage. This should not affect the results of high-resolution texture processing. Has no effect if the shading mode is set to anything other than “image-based” or “image-based with textures.” Changing this option will not have an effect until a new project is loaded.

> **Width and Height** (recommended: 512x512): The resolution of the depth maps. Only applied. Has no effect if the shading mode is set to anything other than “image-based” or “image-based with textures,” or if the “Preload Visibility and Shadow Testing” option is disabled. Changing this option will not have an effect until a new project is loaded.

Mipmaps (recommended on): Generate mipmap levels for every preview image. This increases memory usage but may help eliminate some visual artifacts that are the result of image resampling. Has no effect if the shading mode is set to anything other than “image-based” or “image-based with textures.” This should not affect the results of high-resolution texture processing. Changing this option will not have an effect until a new project is loaded.

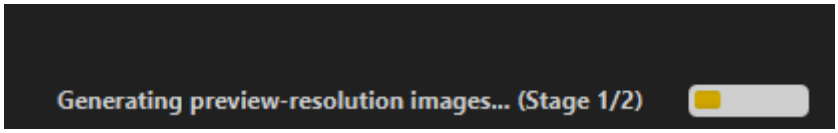
Reduce Viewport Resolution: This reduces the resolution of the rendering by half to help speed up rendering up to four times the framerate. This is particularly meant to help with the image-based shading modes on High-DPI displays and should be enabled if you have a retina display.

Progress dialog



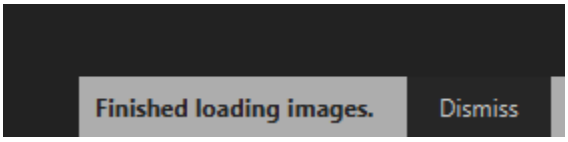
Upon starting a task, the progress bars window will appear. From this window, you will find several pieces of information, including elapsed times, a loading time estimate for the current stage, and the status of your task's completion. You can cancel your task if desired, or minimize the window without cancelling.

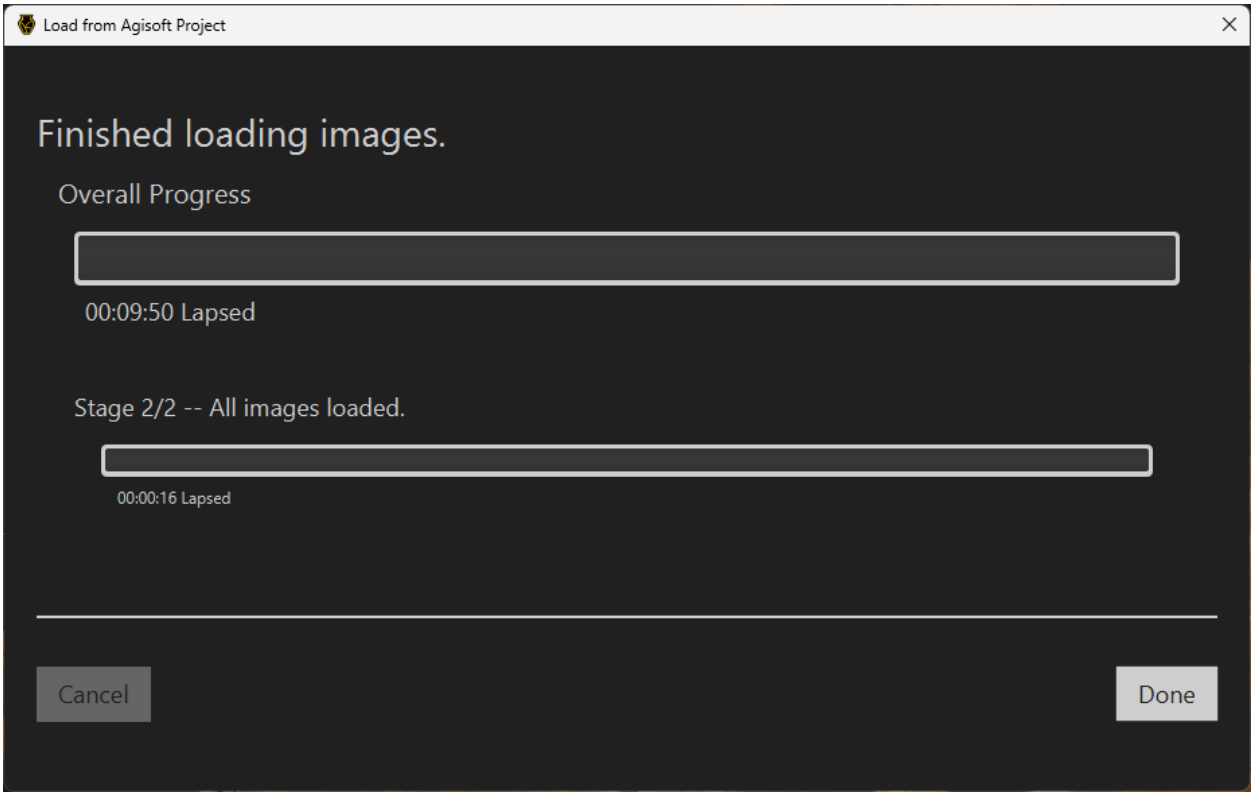
Note: After hitting the "Cancel" button, the task may take several seconds to cancel. This is expected behavior, as Kintsugi 3D Builder will only cancel a task after completing one of its sub-tasks.



Upon minimizing the task, a miniature progress bar will appear at the bottom of the window. Click this bar to reopen the progress bars window.

When the task completes, both versions of the progress bars will change color to indicate so. Click “Dismiss” or “Done” to hide the corresponding modal.

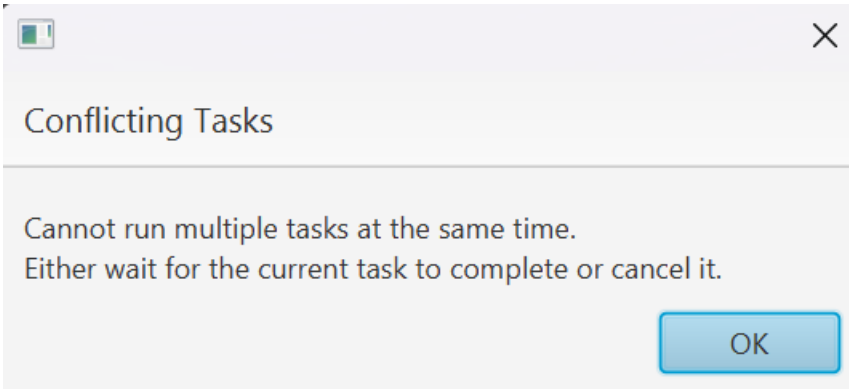




Some tasks cannot run at the same time. These tasks include:

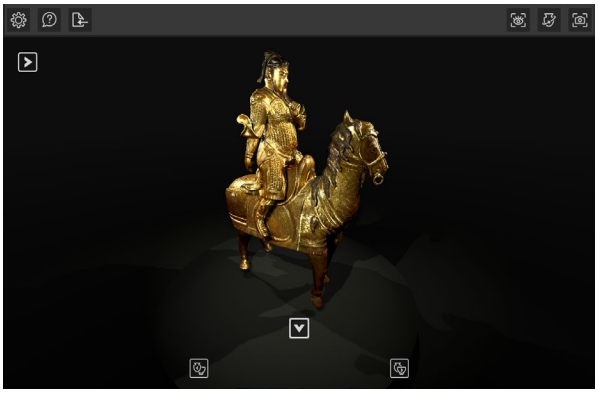
- Creating or loading a project
- Exporting a project (or sample images such as Resample or Orbit Animation)
- Processing textures
- Any other task which would open the progress bars modal

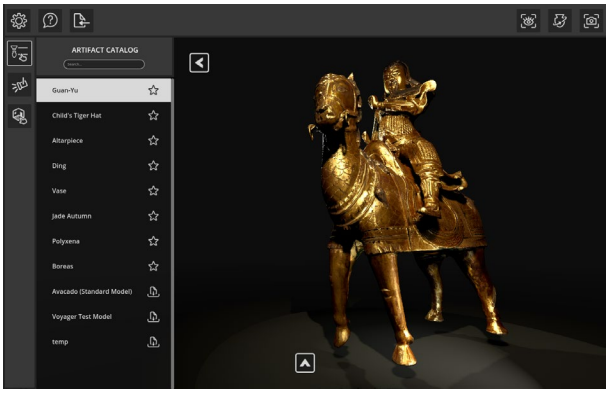
If you attempt to run two conflicting tasks at the same time, a popup will prevent you from doing so.

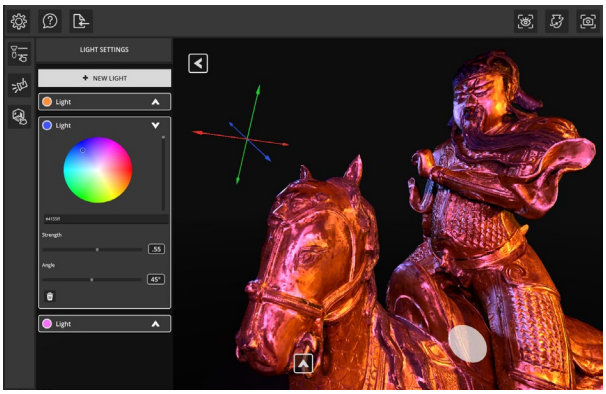


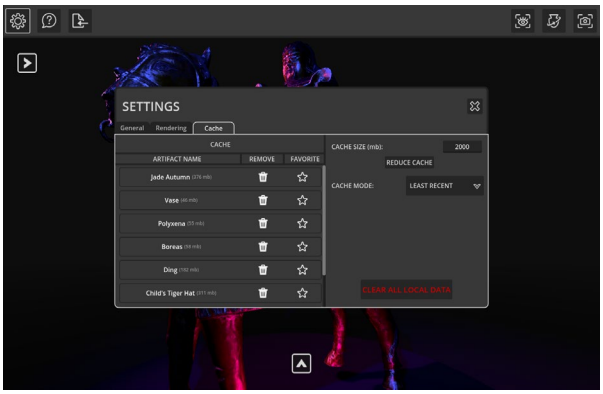
Kintsugi 3D Viewer

Kintsugi 3D Viewer is the sibling project to Kintsugi 3D Builder, also available as free and open-source software, and included with Kintsugi 3D Builder by default. Kintsugi 3D Viewer is the only viewer that will currently be able to support the “basis” material representation produced by Kintsugi 3D Builder. Kintsugi 3D Viewer can either open local files, or download static content from an institutional server. Web, PC, and Mac builds of Kintsugi 3D Viewer are available, and mobile (Android / iOS) builds are in development.







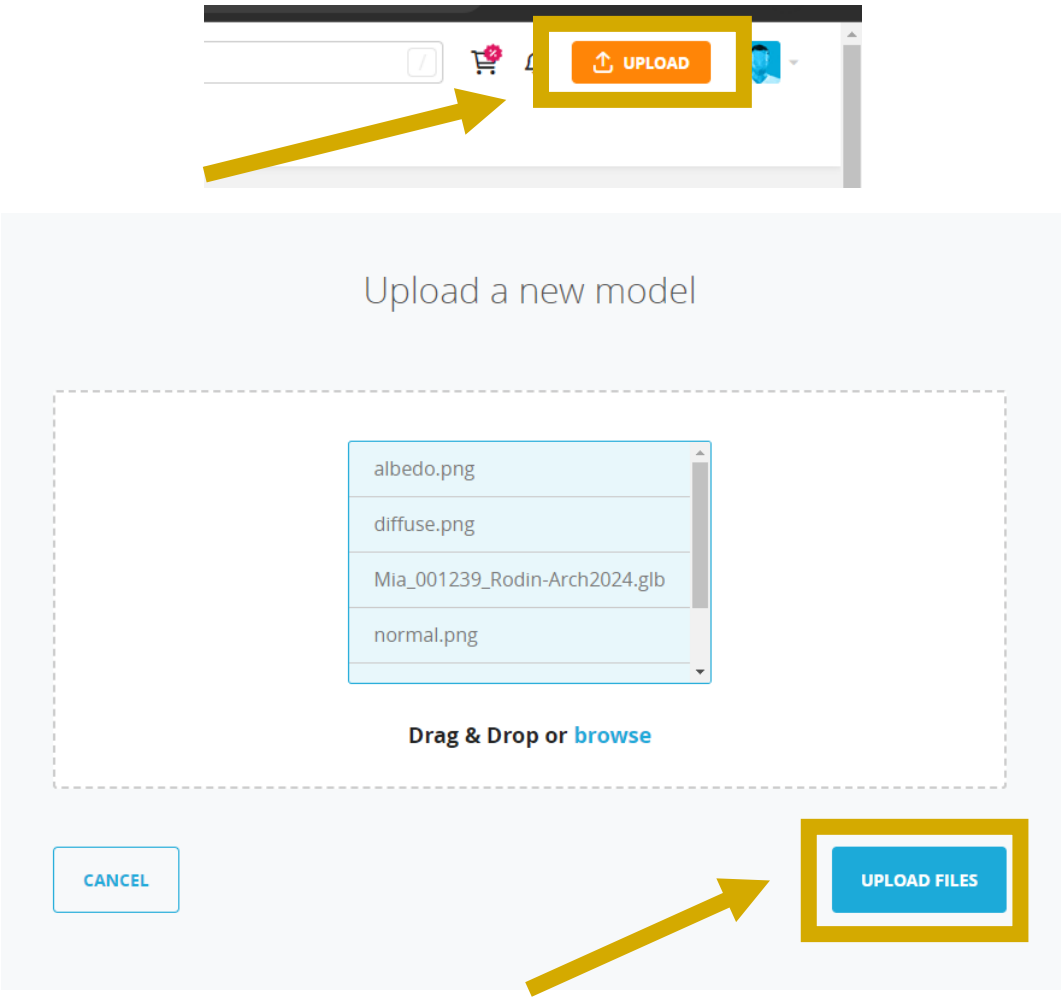


Uploading to Sketchfab

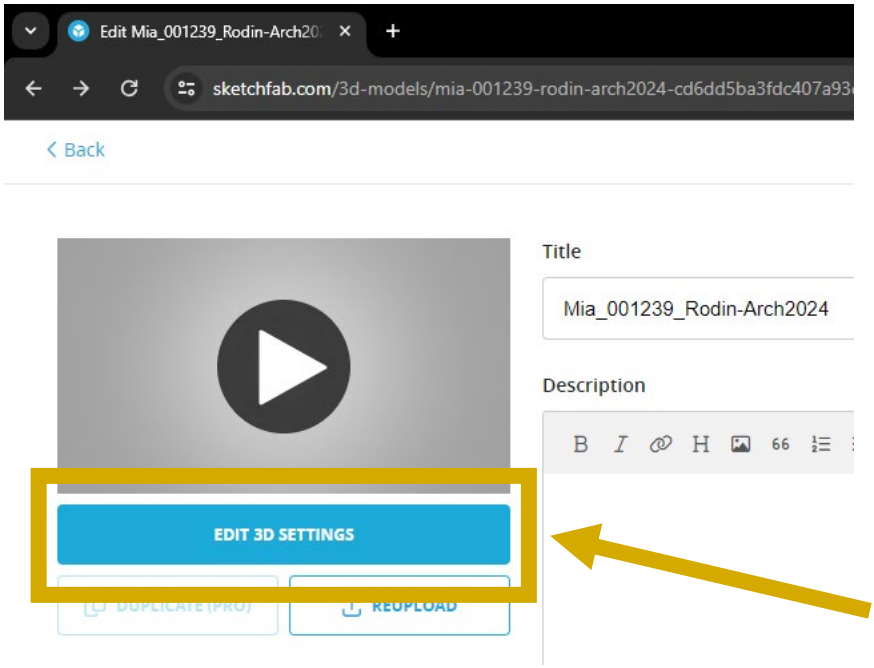
Kintsugi 3D Builder is able to generate textures that can be used effectively in Sketchfab. However, some care must be taken when uploading to get the best results.

First, it is important to simultaneously upload the .glb model along with the albedo, ORM, diffuse, specular, and normal map textures produced by the “Export glTF” task. This can be done by dragging and dropping all the files simultaneously into the “Upload” screen on Sketchfab.

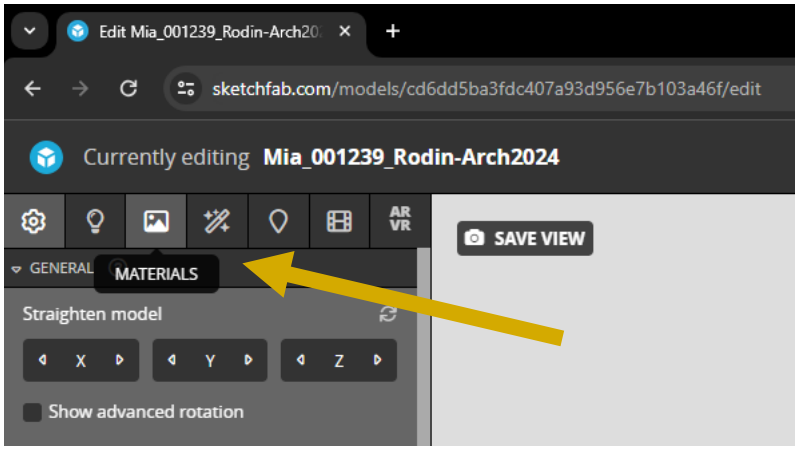
WARNING: Uploading loose textures may cause them to be flipped vertically. If you need to replace textures after uploading, you will need to flip them vertically using Photoshop before uploading to Sketchfab.

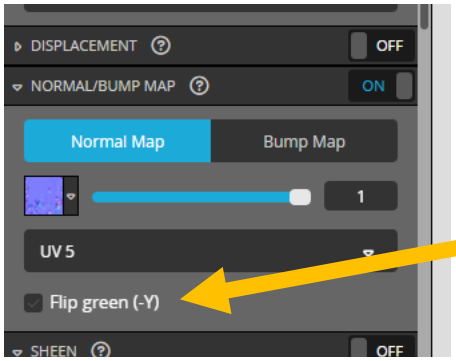


After uploading files, Sketchfab will automatically set up the model using the metallicity representation. To preview this and edit the material configuration, click on the “EDIT 3D SETTINGS” button that appears after uploading files:

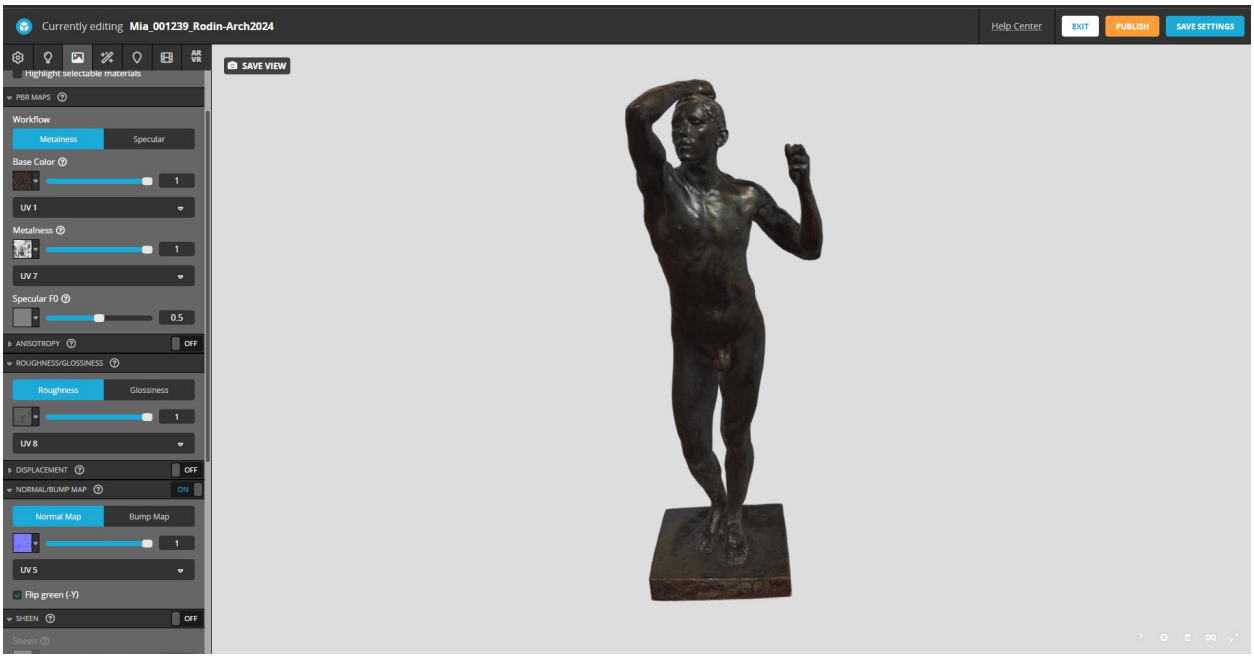


However, Sketchfab's internal coordinate system requires the green ("Y") channel of the normal map to be flipped. This can be done by checking the "Flip green (-Y)" option under "NORMAL/BUMP MAP."

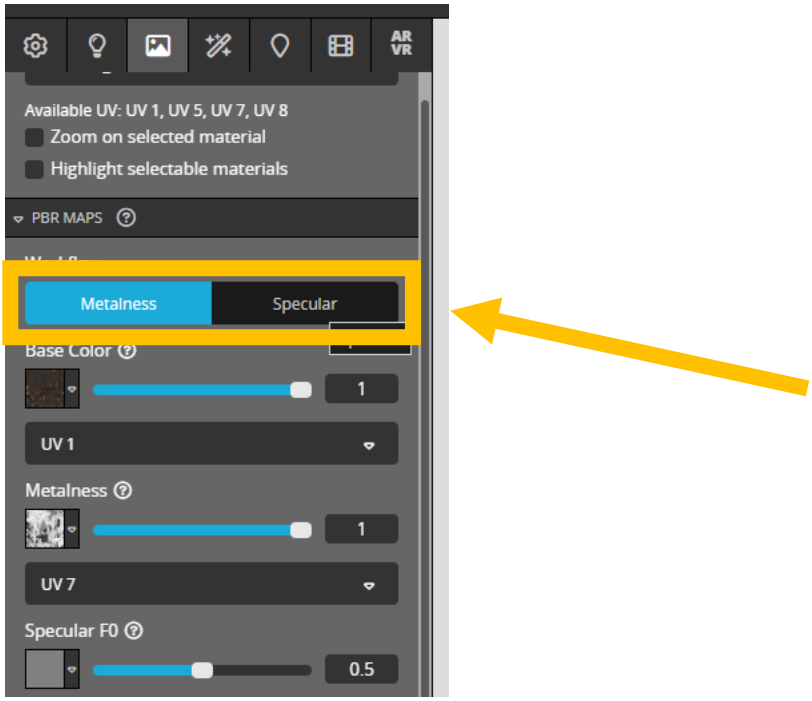




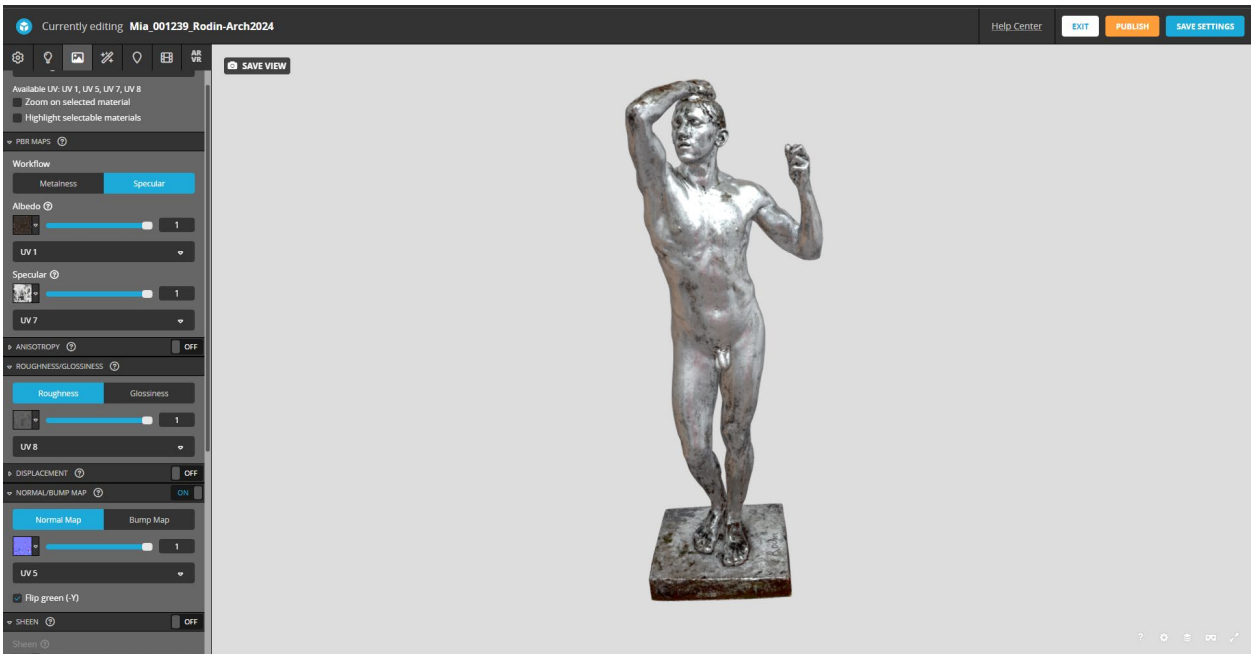
This should result in a reasonable rendering using the "metallicity" representation.



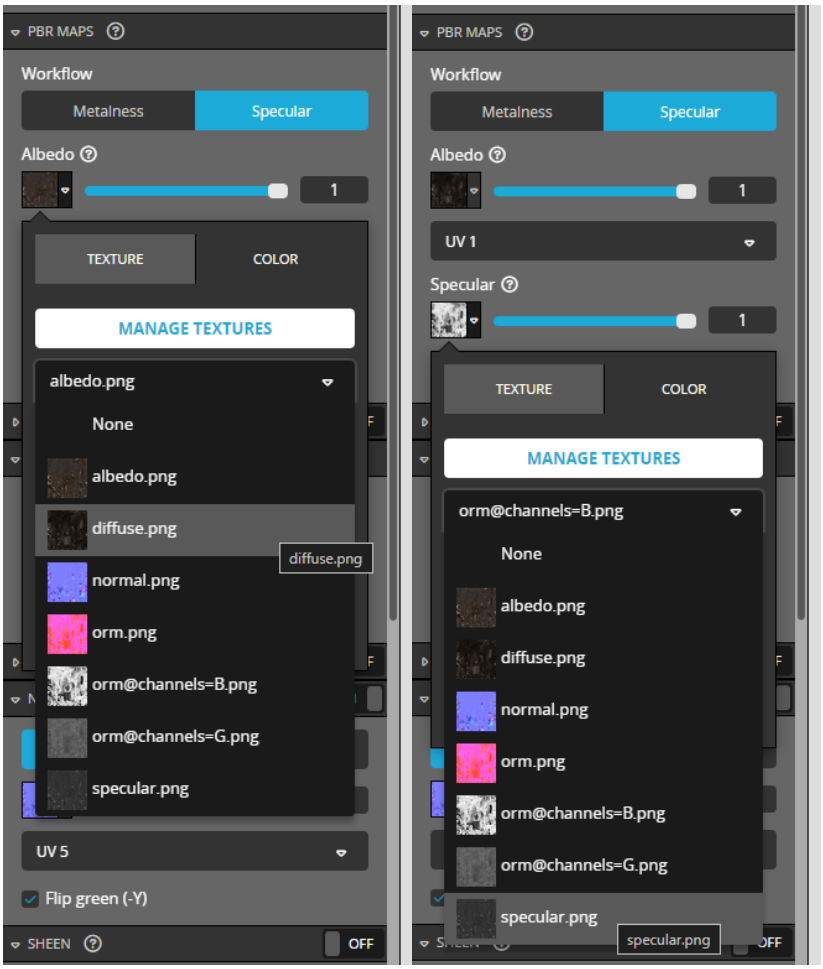
To switch to use Kintsugi 3D's "reflectivity" representation, change the "Workflow" toggle from "Metalness" to "Specular."



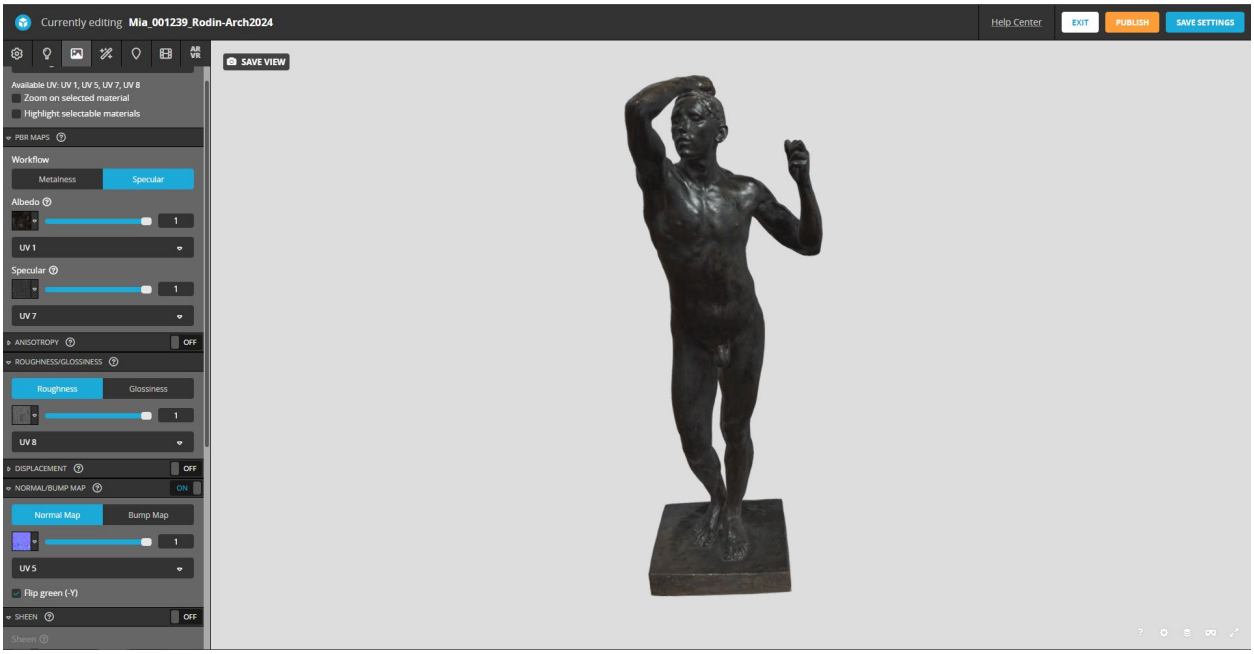
The model will not look right immediately after making this change, as we still need to tell Sketchfab which textures to use for this workflow.



First, click on the texture thumbnail for “Albedo,” and from the dropdown where it probably says “albedo.png,” select “diffuse.png.” Next, click on the texture thumbnail for “Specular.” A dropdown should say “orm@channels=B.png.” Sselect “specular.png.”



After doing this, the “reflectivity” representation should be accurately rendered in Sketchfab.



Importing to Blender

Models processed in Kintsugi 3D Builder can be imported into Blender. For best results, the surface shader in Blender should be set to "Specular BSDF" rather than "Principled BSDF." This workflow essentially uses the “reflectivity” material, comparable to the Sketchfab upload workflow for material accuracy. This only works with the EEVEE renderer.

In general, given an exported model and textures from Kintsugi 3D Builder, the textures should be set up in Blender as follows:

- diffuse.png -> Base Color
- specular.png -> Specular
- roughness.png -> Roughness
- normal.png -> Normal

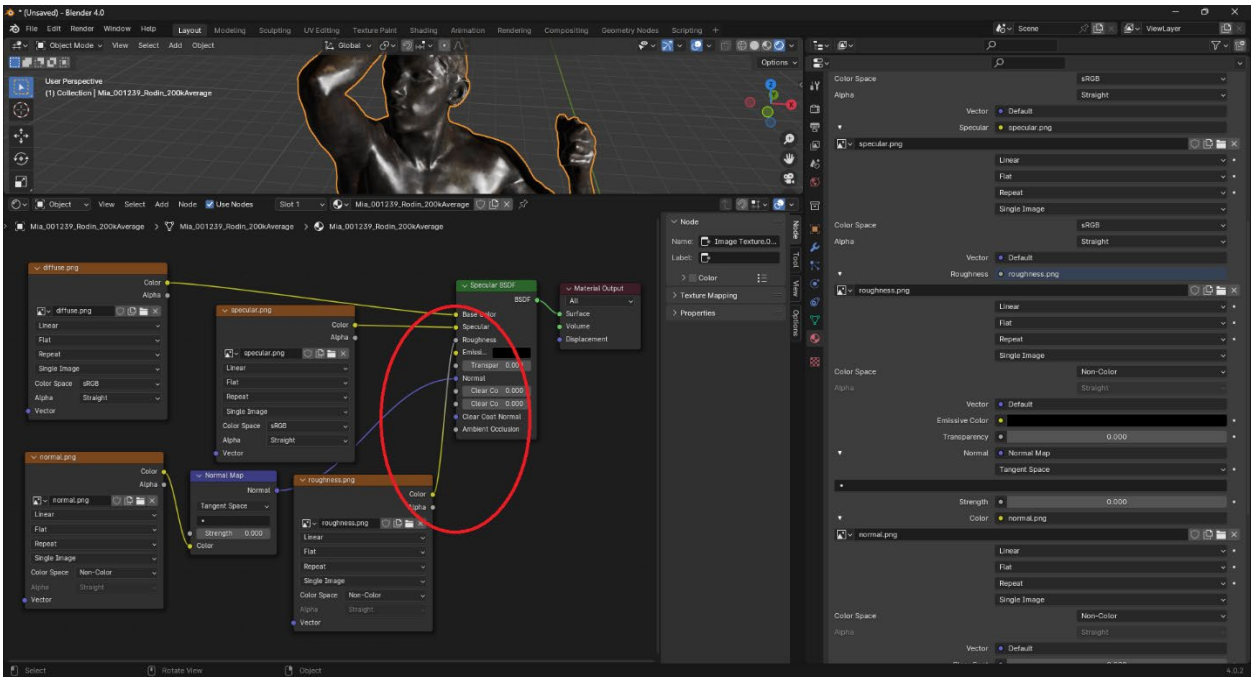
However, there are few important adjustments to make besides just hooking those up.

First, each texture must have its color space set correctly. They should be as follows:

- diffuse.png: sRGB
- specular.png: sRGB
- normal.png: Non-Color
- roughness.png: Non-Color

Second, by default, Blender will try to use the alpha channel of roughness.png for the Roughness map. This must be fixed in Blender using the node-based "Shader Editor." In that context, connect the "Color" pin from roughness.png to the "Roughness" pin on the Specular BSDF node (by default it will use the Alpha pin instead).

Here's what it should all look like (with the Roughness fix circled):



Troubleshooting

Black model

From time to time, bugs in the code of a 3D rendering application can result in a pitch-black model against a gray background. If you experience this in Kintsugi 3D Builder after loading an object with no errors, try switching to Shading > Simple Specular. You should see the 3D shape with a shiny gray “plastic” appearance. If this is the only mode in which something appears on the screen, it is possible that your Metashape or Reality Capture project has an attribute that Kintsugi 3D Builder does not yet support. If you provide us with the files you were trying to load, that will help us determine what about the project was not supported so that we can eventually add support for it.

You should also make sure that you reset the model transformation before exporting your cameras and model files from Metashape (Model > Transform Object > Reset Transform in Metashape) as model transformations are known to be problematic for Kintsugi 3D Builder in some circumstances.

Memory considerations

The program will run out of video memory and may hang or crash if you try to load too many images. If you think you might be running out of memory, here are some things you can try:

Options that will ***not*** affect the quality of the processed textures and 3D model:

- **Make sure image compression is enabled** under Settings > Visual Settings.
- **Reduce the preview image resolution** under Settings > Visual Settings.
- **Disable “preload visibility and shadow testing”** under Settings > Visual Settings.
- **Disable mipmaps** under Settings > Visual Settings.

Options that ***will*** affect the quality of the processed textures and 3D model.

- **Reduce the number of triangles in your model:** The images most likely will take up more memory than the model, but if you don’t have very much memory to work with and every little bit counts, this will free up a small amount of memory. If you are building models for distribution over the web (i.e. Sketchfab or Kintsugi 3D Viewer), you probably want to be using 64k-poly models anyways for web optimization.
- **Reduce the number of views:** If you are still running out of memory, there may simply be too many views enabled. Around 500 views or less is a good place to start, but you may need to go lower on less powerful hardware. You can open the model back up in Metashape or Reality Capture and disable views in there. The tradeoff is that these views will not be available for texture processing.

Performance considerations

In addition to the memory requirements, the renderer also requires significant processing power from your graphics card. You may find it runs slow on lower-end hardware, particularly in the image-based shading modes. Here are some things you can do to speed up the renderer.

Options that will ***not*** affect the quality of the processed textures and 3D model:

- **Make sure relighting is disabled (under the Environment menu):** Relighting significantly impacts performance and is not needed for a basic workflow.
- **Switch the shading mode to one that is not "image-based" (under the Shading menu)** Image-based shading is considerably more performance-intensive than other shading modes. However, it is important to note that the image-based shading modes are the only ones at present that effectively preview the projection of the photos onto the 3D model prior to the long texture processing step.
- **Zoom out/make the window smaller:** The fewer pixels you are trying to render at once the faster it will be so zoom out and make the model smaller to increase performance.
- **Try reducing viewport resolution** (under Settings > Visual Settings) to cut the number of pixels in half in both dimensions. While this option is intended for high-DPI displays it always has a significant effect on rendering speed.

Options that ***will*** affect the quality of the processed textures and 3D model.

- **Reduce the number of triangles in your model:** While the renderer is generally pixel bound and not geometry/vertex bound, it may still prove useful to reduce the faces from the default model generated by Metashape or Reality Capture. If you are building models for distribution over the web (i.e. Sketchfab or Kintsugi 3D Viewer), you probably want to be using 64k-poly models anyways for web optimization.
- **Reduce the number of views:** Reducing the number of views not only reduces the memory requirements, but also makes the renderer run faster. Around 500 views or less is a good place to start, but you may need to go lower on less powerful hardware. You can open the model back up in Metashape or Reality Capture and disable views in there. The tradeoff is that these views will not be available for texture processing.

Photography special cases

A photographed piece may contain structural supports such as poles, back supports, etc. In order for these parts to not be projected onto false surfaces and distort the main textures, remove these parts of the model last, after you have created the Kintsugi 3D textures. Depending on the accuracy of the photogrammetric reconstruction, Kintsugi 3D may be able to use the geometry of the supports to mask out the supports from the photographs when building the textures.

For developers

The shaders for Kintsugi 3D Builder are provided as text files. The software loads these text files when it runs. On Windows, these files are loose files in a “shaders” subdirectory within the installation directory. On MacOS, they are encapsulated in the app bundle. The F11 key can be used to reload and recompile these shaders at runtime after a model has loaded. This can be used to rapidly experiment with and debug modifications to the shaders.

It is also possible to use the Kintsugi 3D architecture to run arbitrary shader programs. The “Generic” export option under the File > Export submenu provides this utility. This feature requires a fragment shader written in GLSL, and can optionally take a custom vertex shader as well, but also includes several built-in vertex shaders. Documentation on the specific functions and variables made available in these shaders by Kintsugi 3D Builder will be forthcoming.