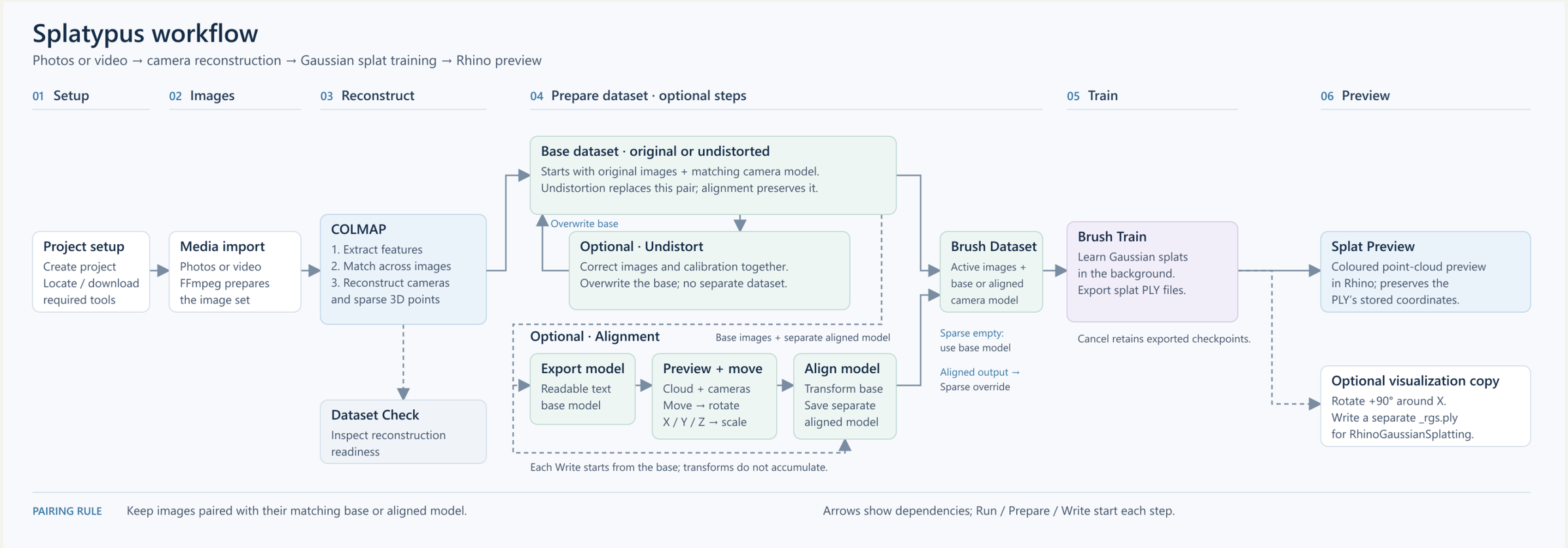
From media to models

Splatypus 0.2.0 - photogrammetry and Gaussian splat training workflows for Grasshopper.

A connected local workflow

Splatypus connects FFmpeg media import, COLMAP camera reconstruction and Brush training through one project. Inspect cameras and points, align the reconstruction, prepare the active dataset, train and inspect PLY exports.



Know the scope

A Gaussian splat viewport renderer is not included. Splat Preview shows colored points. Version 0.2.0 adds Quick Align, video camera playback, calibrated viewport capture and automatic backgrounds.

Separate Windows x64 and macOS packages are provided. Automatic Mac tool installation targets Apple Silicon. Consult README.md for package requirements and platform details.

Install and check

Install through Rhino Package Manager or use the package matching your platform.

Before installing

Use **Rhino 8.34.26223.11001 or newer with .NET 8** and Grasshopper. Rhino 7 and .NET Framework mode are unsupported. Choose Package Manager or manual installation and keep one installed copy. Back up your project media and save working definitions outside the installed package folder.

Install from Rhino

Run **_PackageManager**, search **Splatypus**. Choose the compatible package, install, then restart Rhino. Open Grasshopper, find the Splatypus tab and place **SP Hello**. Connect Status to a Panel. Expected: **SP is loaded: Ready**.

Manual ZIP installation

Find Grasshopper > File > Special Folders > Components Folder. Close Rhino. Unblock the downloaded ZIP on Windows if the checkbox is present, extract it and copy **both Splatypus.gha and Splatypus.Core.dll** into a Splatypus subfolder. Keep one copy and restart. Use the build for your operating system; update both files together.

External tools are a separate step

Use SP Download Tools with a writable Folder (or leave it blank for the per-user default). On Windows choose **CUDA=False** for CPU COLMAP or True for NVIDIA CUDA. Press Download with a Button. Wait for Ready=True and connect the returned FFmpeg, COLMAP and Brush paths to later Exe inputs. Startup checks do not certify GPU training.

Platform	Pinned tool versions
Windows x64	FFmpeg + ffprobe 8.0; COLMAP 3.11.1 CPU/CUDA; Brush 0.3.0.
Apple Silicon macOS	FFmpeg + ffprobe 7.1.1; COLMAP 3.11.1 CPU through conda-forge; Brush 0.3.0.

Automatic tool installation does not support Intel Macs or Windows ARM. Tools can also be supplied through manual native executable paths. See README.md, Installation and External tools, for detail.

Reconstruct the scene

Begin with a small, representative capture that you own or have permission to process.

1. Create the project

Place SP Project. Choose a writable Parent Folder and Project Name, set Create=True and use Root for downstream components. The project contains images, colmap, brush and output folders. Keep your original media separately.

2. Import media

Connect FFmpeg Exe and Root to SP Media Import. Set source and sampling options. The estimate updates when inputs change. If files inside a source folder change, use **Refresh media estimate** in the component right-click menu on Windows, or a Button on **Refresh Estimate** on macOS. **Format=0** is JPEG; **1** is PNG with existing alpha preserved. **Skip Duplicates** filters near-identical video frames and retains timing. Estimate is before filtering. Then press Run.

3. Run COLMAP

Connect COLMAP Exe and Root to **SP COLMAP Pipeline**. Extract, Match, Sparse and Export default to True; Undistort defaults to False. Press Run and follow Status/Progress. Alternatively, run the individual reconstruction components in sequence. **Matching:** 0 Exhaustive, 1 Sequential, 2 Spatial (complete position priors), 3 Auto. **Extract Size=3200** and **Features=8192** control extraction; changing them clears old matches.

4. Inspect and optionally refine

Use Camera Planes and Sparse Point Cloud. Sparse reconstruction has no automatic real-world scale. Optional undistortion updates images and calibration together. Alignment writes a separate model with transformed points and camera poses, preserving the base pair. Connect Aligned Sparse Model to Brush Dataset's Sparse Model Override when you want to train that branch.

One current image/model pair

Root -> SP Active Dataset -> SP Brush Dataset. Re-prepare after reconstruction, undistortion or changing the aligned override. Do not train while changing the dataset. See README.md, Reconstruct cameras with COLMAP, for the complete connection guide.

Capture tips: overlap photographs, vary viewpoints, use sharp exposures and retain visible detail. Motion, reflections and textureless regions are difficult. Exhaustive matching gets expensive as image count grows; more images do not guarantee a better result.

Align to Rhino geometry

Keep the base dataset as the source. Preview the alignment, then write the complete camera model.

1. Connect source geometry

Connect Sparse Point Cloud **Cloud** to **SP Quick Align / Cloud**, and Camera Planes **Planes** (optionally Centres) to **Cameras**. Both previews must use the same original active Dataset. Hide their upstream previews.

2. Pick three corresponding pairs

Double-click Quick Align. Pick a highlighted sparse point, then its matching native Rhino reference point. Repeat source/reference for all three pairs. Use distinct, widely separated features forming a triangle and Rhino object snaps. **Error** reports RMS fit distance in document units.

3. Refine or reset

Double-click again to refine the currently aligned result. Xform remains the complete alignment from original coordinates. **Esc** preserves the previous completed fit. **Reset** restores original coordinates; Undo and saved pairs are supported. Repick or Reset if the source cloud changes.

4. Add optional fine adjustments

Quick Align Cloud/Cameras -> **Geometry Translator / Geo**. Quick Align Xform -> **Previous**. Set CenPt in aligned coordinates, then adjust movement, rotation in degrees and positive scale. Previous records the earlier fit for saving without applying it again to the geometry.

5. Write and use the aligned model

Connect Translator **Transforms -> Align COLMAP Model / Xforms**, supply original Dataset and COLMAP Exe, then press Write. For Quick Align alone, connect its Xform directly. Do not flatten duplicate transform branches or add Quick Align twice. Write preserves the base and replaces colmap/aligned. Connect **Aligned Sparse Model** to Brush Dataset **Sparse Model Override**; use Aligned Text Model to inspect saved cameras. Rewrite after reconstruction or undistortion.

Train, export and preview

Brush is a separate local application with its own graphics-device requirements.

5. Prepare the Brush dataset

Connect Root to SP Active Dataset and its Dataset to SP Brush Dataset. Enable Prepare. Dataset links depend on the project files remaining available and the filesystem supporting links. Follow any OS link-permission error.

6. Start a short training test

Connect Brush Exe, the prepared Dataset and Root to SP Brush Train. Use a new Job Name to retain earlier exports. The default is 30,000 steps; start with a short test. Set Viewer=False for a basic background run. Run starts the job; Cancel stops it. **Resolution=1920** limits the training image side; **Max Splats=10,000,000** caps growth. Lower settings can reduce workload but are not a GPU-memory guarantee. Use a Button or reset Run to False.

7. Read the outputs

Exports and the full brush-training.log are in **output/<job name>**. Check Code and Status. **Replace=True removes that job's previous exports before training.** Completed exports can remain after cancellation.

8. Inspect a PLY

Select an exported binary little-endian PLY in SP Splat Preview. Max defaults to 150,000 points; lower it for faster interaction. Preview keeps stored coordinates and displays a decimated colored point cloud. It does not rasterize Gaussian splats.

Optional external rendering handoff

Rotate for Visualization writes an additional **_rgs.ply** rotated +90 degrees around X for a compatible external RhinoGaussianSplatting workflow. It preserves the original PLY. Splatypus does not include or license that third-party renderer.

If training fails, review the log, verify the selected GPU/driver with the pinned Brush build, reduce the input workload and test Viewer=False. CPU COLMAP selection does not make Brush a CPU trainer.

Replay camera tracks

Follow reconstructed video cameras through the visible Rhino scene. No movie export is performed.

1. Read extracted video frames

Connect Active Dataset **Dataset -> SP Video Frames / Dataset**. Frames has one ordered branch per source video, excluding still photographs. Videos lists the corresponding zero-based indices. Times and Registered show timing and camera availability.

2. Select camera coordinates

Leave Model empty to prefer an existing aligned COLMAP text model, otherwise the base model. Status shows the selected folder. Use Model to choose a specific text model or project root. Write alignment before playback if the scene uses aligned coordinates.

3. Play, pause and seek

Connect **Tracks -> SP Video Playback / Tracks** without flattening or selecting one item. Choose Video Index, activate a Rhino model viewport and set Play=True. Set False to pause; change Time to seek. Preview=True applies the sought view while paused.

4. Control motion and restore the view

Speed=1 follows recorded timing, 0.5 is half speed and 2 is double speed. Loop repeats with a cut to the start. Smooth=True softens position/rotation jitter; False follows original positions with straight segments. Press **Restore View** to stop and return to the saved view, then toggle Play off before restarting.

Timing and missing cameras

New imports retain timestamps even after duplicate filtering. Keep ffmpeg-media.json with the project. Older imports can supply their original mixed files/folders to Media and use a default two-frames-per-second pace. There is no FPS or Refresh socket. Missing camera poses interpolate between available poses; a track with no registered camera cannot play.

Camera outputs use X right, Y up and negative Z forward. Playback adapts framing to the viewport aspect ratio and does not reproduce lens distortion or off-centre principal points. Pause playback before View Capture.

Capture and retrain

Capture known camera views of the Rhino scene, then train without repeating camera reconstruction.

1. Prepare the scene and frames

Put cameras, splats and design geometry in the same coordinates. Load Aligned Text Model in Camera Planes if using saved alignment. Connect **Camera Planes / Frames -> SP View Capture / Frames**, not its Planes surfaces. Flatten frame branches. Hide guides, camera surfaces and inspection clouds; pause other camera animation.

2. Choose a fresh project Root

Use a separate destination root with empty images and colmap folders. Existing data is rejected. Defaults: maximum **1920 x 1280**, Rendered, 35 mm equivalent Lens, white background, **Wait=1 second**. Size is reduced to fit the visible viewport; Status reports the actual calibrated dimensions. Frames do not contain original lens calibration.

3. Optionally estimate the backdrop

Connect the coordinate-matched PLY to **SP Auto Background**, plus the same Frames, Lens and Aspect (Width / Height). Supply Transform if the displayed splat was moved. For a gradient: **Bottom -> Background**, **Top -> Background Top**. For solid fill: BG -> Background only. Preview inspects the fill; wire colors explicitly for capture. Keep external splat relighting off.

4. Capture a small batch first

Activate an unlocked model viewport and press Run. Keep it visible; do not navigate, resize, switch views or start other commands. Keep windows and the cursor away. Increase Wait for a heavy renderer. ScreenCaptureToFile must support the renderer; Raytraced/progressive modes are unsupported. The original view is restored afterwards.

5. Train the captured dataset

Inspect the PNGs. Connect **Dataset -> Brush Dataset**, enable Prepare, then connect its Dataset to Brush Train. Use the capture Root and leave Sparse Model Override empty. Images, COLMAP cameras, transforms.json and brush/dataset are written directly under Root. Brush can initialize this camera-only dataset. Cancellation retains saved images; incomplete outputs are withheld. Use a fresh Root for another batch.

Keep the work traceable

Use logs and small reproductions to diagnose problems without exposing private project data.

Common checks

No tab: check Rhino/.NET runtime, both plugin files, platform and duplicate installations. **Run does nothing:** reset False or use a Button. **Download fails:** check Folder permissions and HTTPS access to GitHub/CDN; Apple Silicon also needs conda-forge. **Slow preview:** lower Max points.

Logs and cancellation

Set Terminal=True before starting a job for live output where supported. Closing the terminal does not cancel processing; use Cancel. Review Status, Code and saved logs. Keep independent backups before replacement or refinement.

Local processing and privacy

The inspected Splatypus code has no publisher telemetry, login or automatic project upload. Images/models/logs are local. Download Tools sends normal network requests to upstream providers only when triggered. Rhino/Food4Rhino and external tools have their own policies. Logs can disclose usernames, file paths and arguments; redact them before sharing.

Credits and licenses

Splatypus is licensed under MIT, copyright 2026 Zeming Li. FFmpeg licensing depends on the selected build (LGPL with possible GPL parts). COLMAP itself is BSD-3-Clause; Brush 0.3.0 is Apache-2.0. Dependencies are separately licensed. These tools are downloaded separately and are not bundled in the plugin package. Host and tool names remain with their owners; no endorsement is implied.

Support and further reading

Contact lizeming2021@gmail.com. Website: https://zeming.li/. Include Splatypus/Rhino/.NET/OS versions, CPU/GPU, tool versions, settings, Status/Code and a redacted log excerpt. See LICENSE.txt and NOTICE.txt for copyright, use terms and the public support contact.

README.md contains the full workflow, example setup, troubleshooting, privacy and third-party credits. Examples/Splatypus_Workflow.gh is the included Grasshopper definition.

Primary references: McNeel packaging; FFmpeg licensing; COLMAP license; Brush license.