

# TeleSculptor v1.1

User Guide



Matt Leotta

[matt.leotta@kitware.com](mailto:matt.leotta@kitware.com)

Kitware, Inc.  
1712 Route 9, Suite 300  
Clifton Park, NY 12065

June, 2020

## Table of Contents

Introduction .....	3
Branding: Origins and Destinations .....	3
TeleSculptor Installation .....	4
Views and Navigation.....	5
SketchUp Plugin Installation .....	7
User Workflows.....	9
Metadata Inspection.....	9
SketchUp Enhancement.....	10
Dense Automated 3D Models .....	10
Processing Steps.....	11
Create a New Project .....	11
Import a Video .....	12
Run End-to-End .....	13
Track Features.....	14
Estimate Cameras/Landmarks .....	16
Save Frames .....	17
Set Ground Control Points .....	17
Set 3D Region of Interest .....	19
Batch Compute Depth Maps.....	21
Fuse Depth Maps .....	22
Colorize Mesh .....	23
Export Data .....	25
Measurement Tool.....	27
Advanced Tools.....	28
Filter Tracks .....	28
Triangulate Landmarks.....	28
Refine Solution .....	28
Reverse (Necker).....	28
Align .....	29
Save Key Frames .....	29
Compute Single Depth Map.....	30
Compute Options.....	30
Ignore Metadata .....	30
Variable Lens.....	30
Fix Geo-Origin.....	30
Advanced Configuration Options.....	31
Changing Frame Sampling Rate .....	31
Modifying Algorithm Parameters .....	31
Printing all KLV metadata.....	32

## Introduction

Welcome to TeleSculptor. This software is free and open source software developed by Kitware under multiple SBIR contracts for DARPA and AFRL. The purpose of the software is to calibrate cameras and extract 3D information from video and images, especially aerial video. The software is designed to have different workflows that can either improve the manual modelling process in SketchUp or provide fully automatic 3D reconstruction.

TeleSculptor v1.1 marks the second official release of the software. There are several improvements to the application interface and algorithm robustness since the previous v1.0 release. Please refer to the release notes file for details on all the changes. While TeleSculptor continues to become more broadly useful across imaging scenarios, it is still optimized for the use case of aerial video flying a 360-degree orbit around a target object. TeleSculptor will give best results on a video clip containing one orbit around an object. We will improve robustness on more general imaging scenarios in future releases.

TeleSculptor is a cross-platform desktop application for Windows, Linux, and Mac. This document uses the Windows version of the application to illustrate usage, but everything should work nearly the same on other platforms.

## Branding: Origins and Destinations

TeleSculptor has its origins in a software project developed as “MAP-Tk”, the Motion-imagery Aerial Photogrammetry Toolkit. The original software was not an end user application but a collection of developer tools and libraries. As the software evolved it developed a graphical application that we now call TeleSculptor. At the same time, the software libraries of MAP-Tk were reorganized into a new, broader toolkit called KWIVER (Kitware Image and Video Exploitation and Retrieval). So, the original MAP-Tk has dissolved away leaving behind a TeleSculptor application powered by KWIVER. However, some uses of the name MAP-Tk may persist for historical reasons.

## TeleSculptor Installation

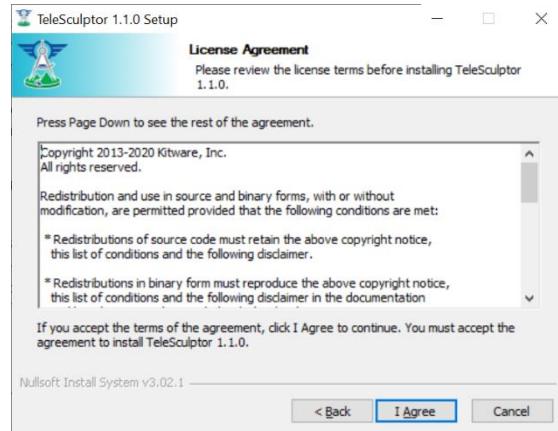
The TeleSculptor software uses a standard installer package on Windows. To install, double click the installer icon and step through the installation steps as shown in the images below. This will require administrative privileges. The name of the 64-bit Windows installer is **TeleSculptor-1.1.0-Windows-AMD64.exe**.



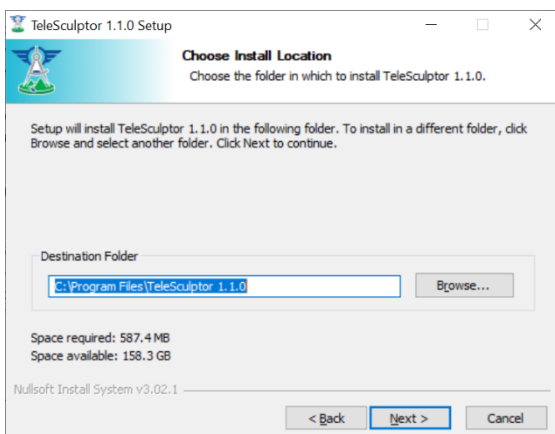
TeleSculptor-1.1.0-Windows-AMD64.exe



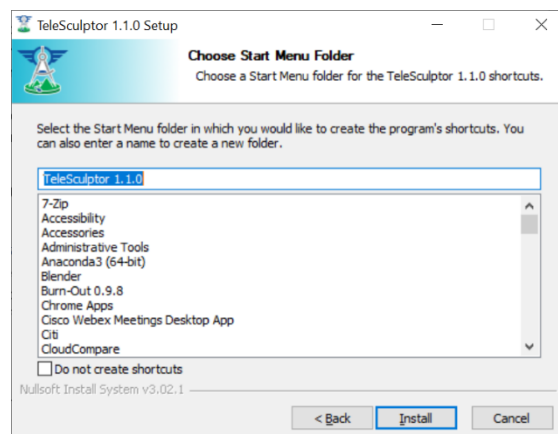
1. Click "Next" to begin the setup



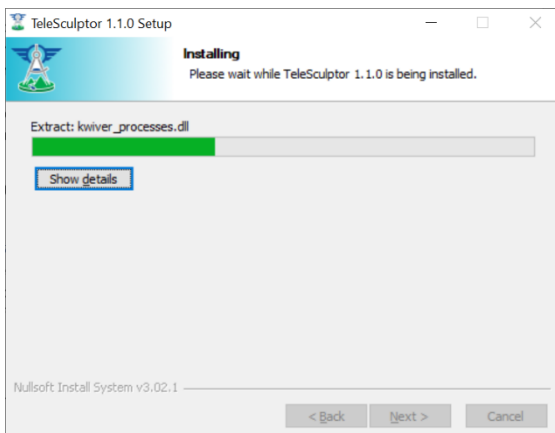
2. Click "I Agree" to accept the BSD license



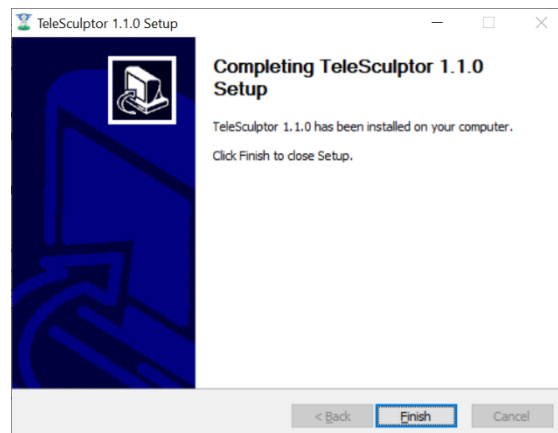
3. Click "Next" to accept the default destination



4. Click "Next" to accept the Start Menu location

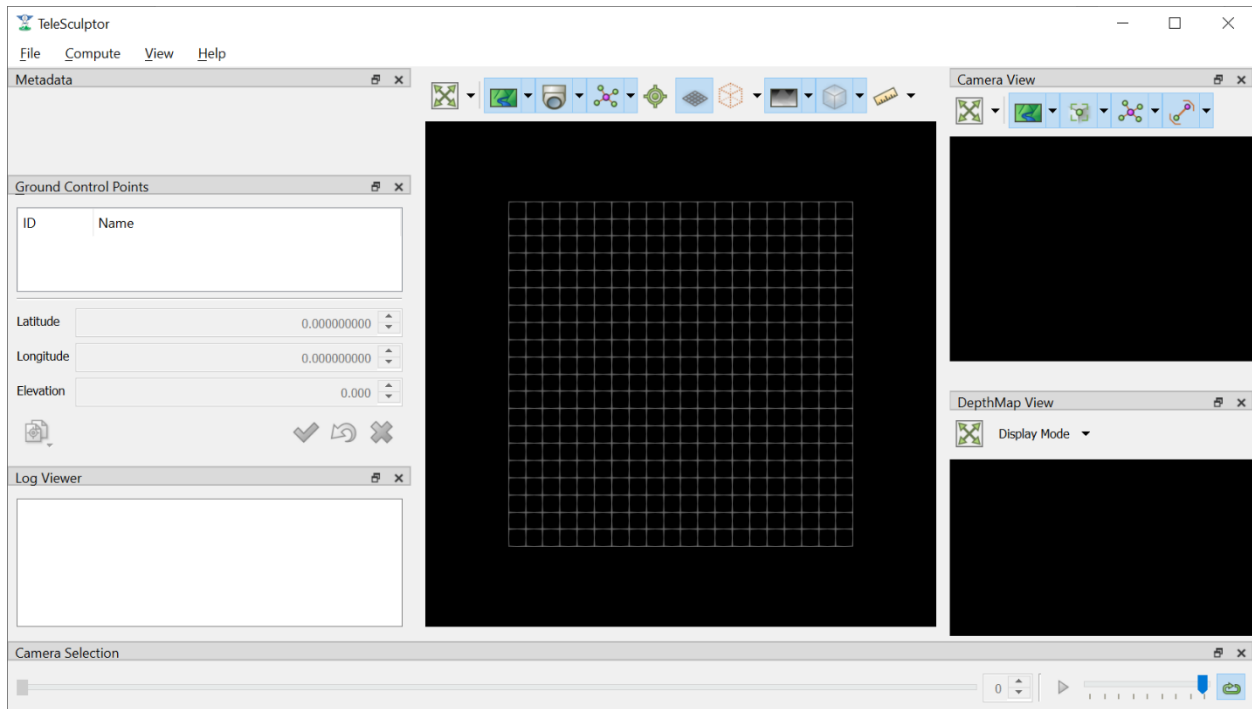


5. Wait for files to install



6. Click "Finish" to complete

To run the application, find TeleSculptor in the Start Menu and click the icon. The program will open with an appearance as shown below.



*The TeleSculptor Application when first opened.*

Once the application is open you can access additional documentation about the various features in the User Manual which is opened from the *Help* menu or by pressing the F1 shortcut key. The User Manual will open in your default web browser. The User Manual does not yet provide step-by-step instructions, as this document does, but it does provide more detailed descriptions of each of the buttons and menu options.

## Views and Navigation

The interface of TeleSculptor is made up of several viewing panes. The primary pane is the 3D world view in the center that shows 3D reconstruction results and camera poses. All other view panes are optional and can be closed, rearranged, or popped out into new windows. Closed panes can be reopened by selecting them under the *View* menu. The size and configuration of these panes is saved when the application closes and restored when opened again. The secondary panes are

- **Camera Selection** – shows a time slider, current frame number, and playback controls
- **Camera View** – shows the current video frame and 2D geometry overlays
- **Depth Map View** – shows the most recent depth map as a 2D image
- **Metadata** – shows the metadata values associated to the current frame
- **Ground Control Points** – shows a list of ground control points and their geodetic location
- **Log Viewer** – shows the log output from running algorithms


The 3D world view has controls for navigating 3D space (e.g. pan, rotate, zoom) while 2D image views have controls for navigating 2D space (e.g. pan, zoom). The controls for these views are as follows


### 3D Navigation

- **Rotate about scene center** – Left click and drag
- **Rotate about camera axis** – Hold Ctrl and left click and drag
- **Pan** – Middle click and drag or hold Shift and left click and drag
- **Zoom** – Right click and drag or scroll wheel
- **Select new scene center** – double left click on scene location

### 2D Navigation

- **Pan** – Middle click and drag or hold Alt and left click and drag
- **Zoom** – Right click and drag or scroll wheel

At any time, the Reset View button () above the pane will reset view to show all the visible data. The keyboard shortcut “R” has the same effect. In some views the drop-down menu under Reset View provides additional view options. In the 3D world view you can set the camera to the primary coordinate viewing directions (Top, Left, Right, Front, Back). You can also toggle between using a perspective and orthographic view. The 3D world view also has “Zoom to Landmarks” option under the Reset View button which is like Reset View, but only considers landmarks. This is a good way to focus on the landmarks or other 3D scene products while keeping the camera frustums visible.

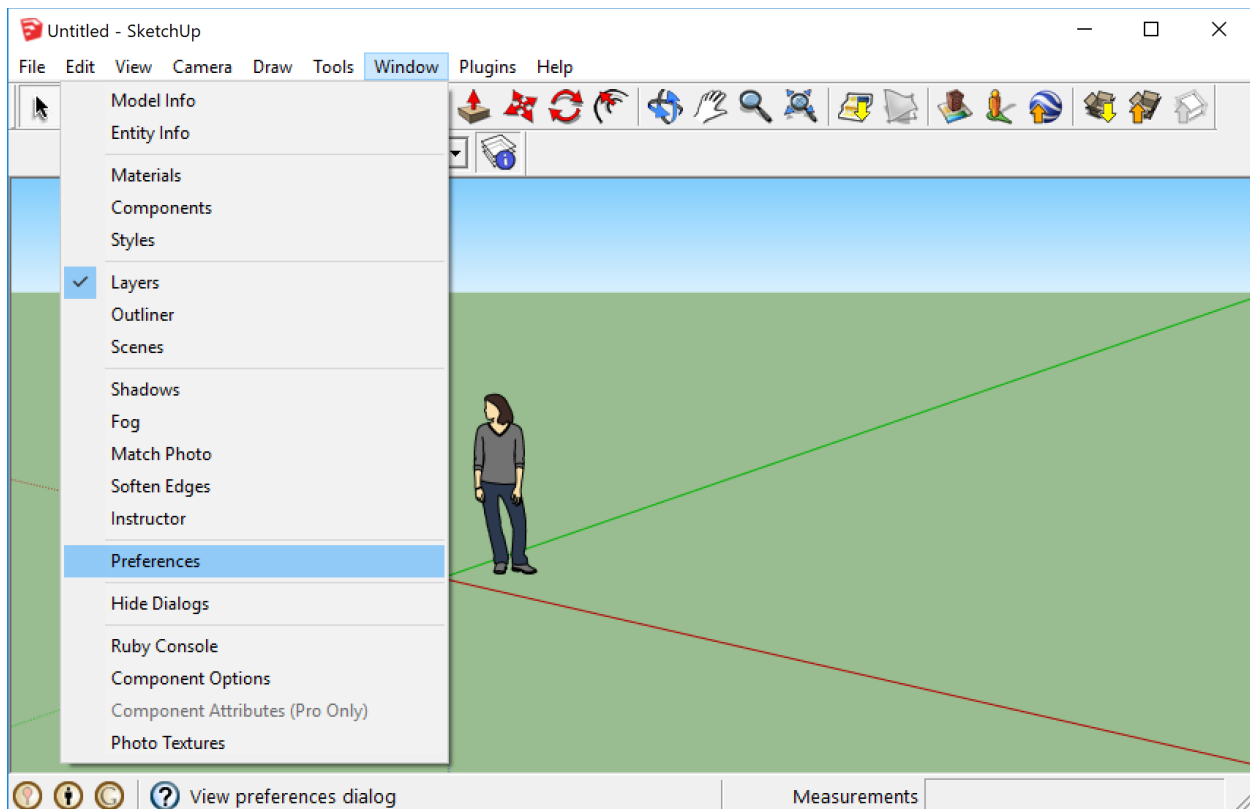
A grid drawn in the 3D world view provides an additional frame of reference for spatial orientation. You can toggle the grid visibility with the grid button () above the world view. This grid is always on a horizontal plane, initially Z=0. The size of the grid is dynamic and adjusts to the scale of the scene data. Grid cells do not represent real units. The height of the plane also adjusts to match the height of estimated landmark points. Additional coordinate axes can be enabled with the *World Axes* item in the *View* menu. The world axes scale to all visible data (including cameras) and the numbers and grid lines on the world axes do represent real world units.

## SketchUp Plugin Installation

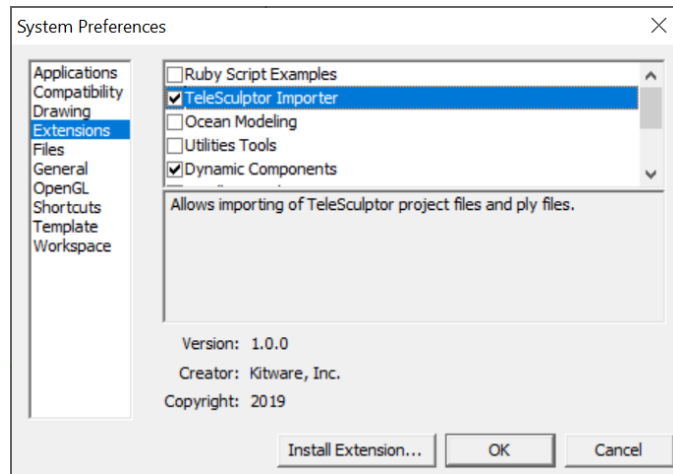
The TeleSculptor application also comes with a plugin for SketchUp that allows SketchUp to read TeleSculptor project files. This is not installed automatically. To install the SketchUp plugin, first locate the plugin in your TeleSculptor installation. If TeleSculptor is installed in the default location, you will find the plugin at

**C:\Program Files\TeleSculptor 1.1.0\share\telesculptor\1.1.0\plugins\sketchup\kw\_telesculptor.rbz**

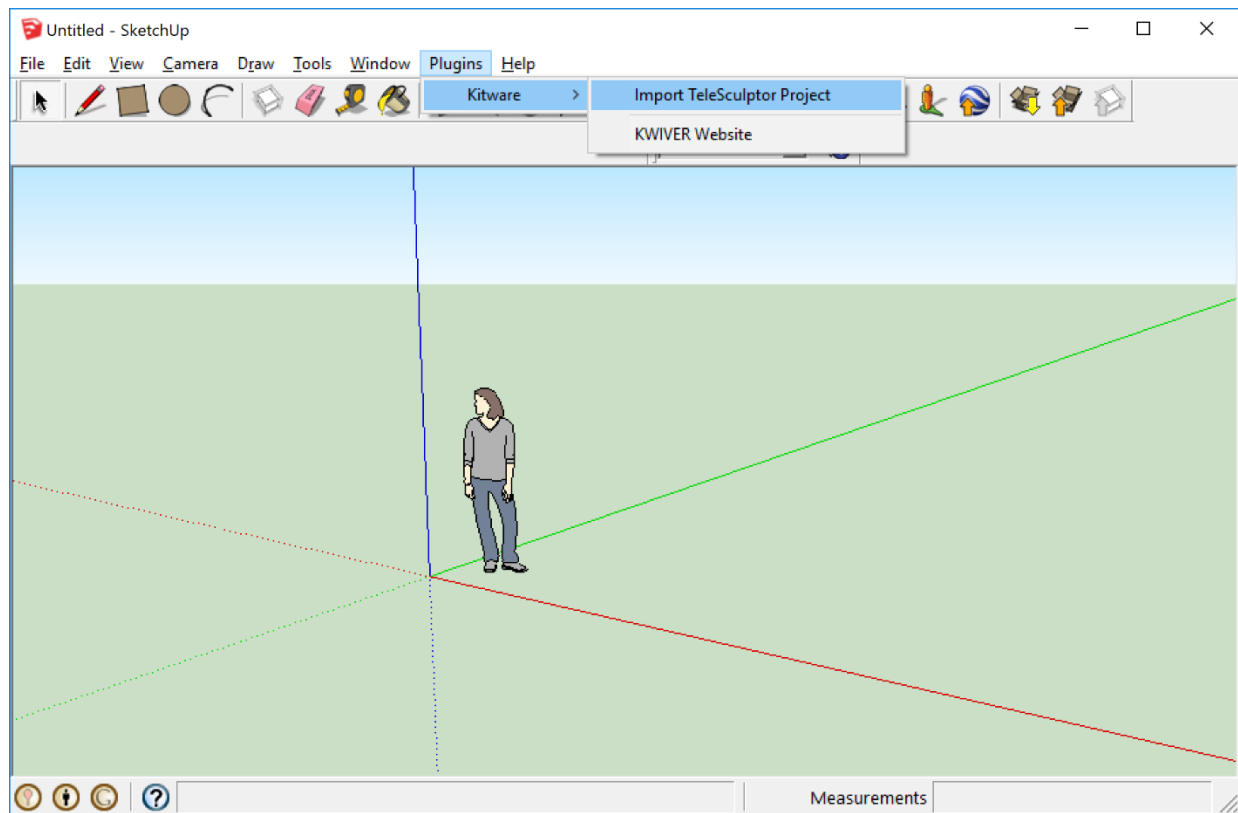
To install this plug in SketchUp, first open SketchUp. Next navigate to the *Window->Preferences* menu as shown in the figure below. Within the *System Preferences* dialog, click on *Extensions* in the menu on the left. Now click the *Install Extension* button at the bottom of the dialog. Use the file open dialog to locate the kw\_telesculptor.rbz file at the location above and click the *Open* button. Administrative privileges are needed to complete the installation. Once installed “TeleSculptor Importer” will appear in the Extensions list and the box next to it should be checked to activate the plugin. The plugin may not be fully active until the SketchUp application is closed and re-opened.



*Access the SketchUp Preferences to install the TeleSculptor plugin.*



*TeleSculptor Importer installed and activated in the SketchUp Extensions list in System Preferences.*



*The Import TeleSculptor Project option in the Plugins menu after installing the extension.*



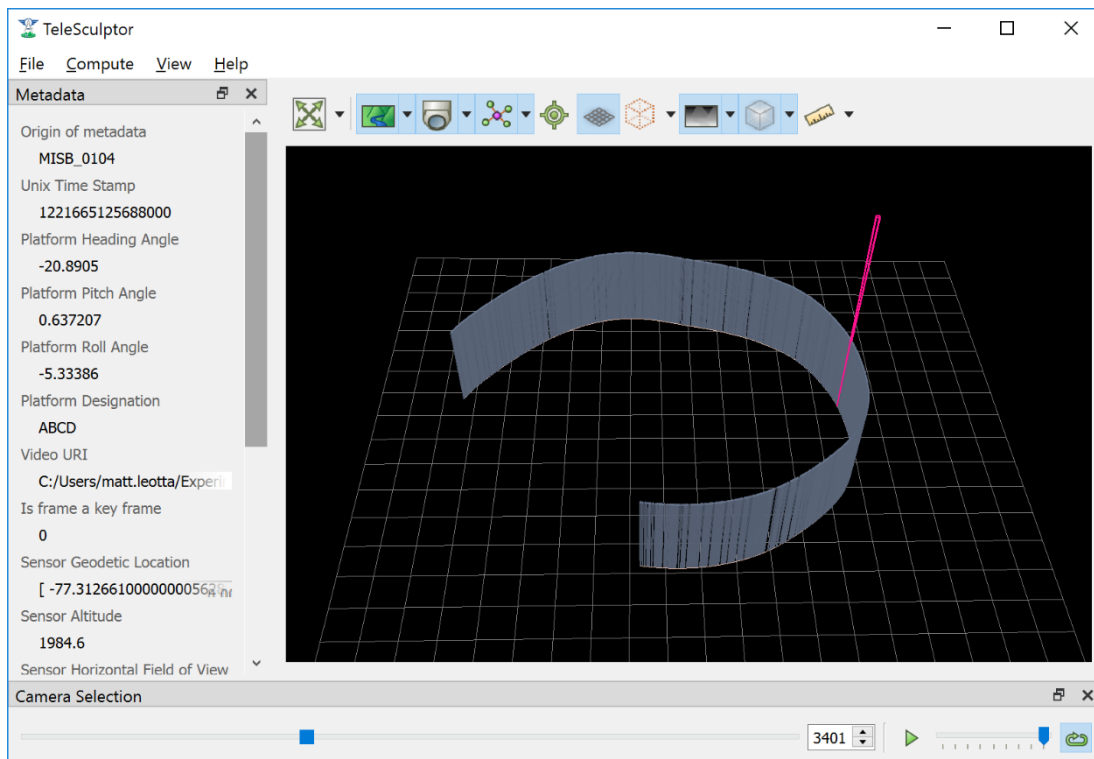
## User Workflows

This section describes the workflows a user should follow when using the TeleSculptor software. There are several ways to use the software depending on whether the end goal is to produce camera models for import into SketchUp, to create a fully automated 3D models from FMV, or simply to inspect the accuracy of the KLV metadata provided with an FMV clip. The available workflows are described below. The details of each processing step are described in the following Processing Steps section.

### Metadata Inspection

This is the simplest capability provided by the TeleSculptor application. When loading a video, the application will read all the KLV metadata (assuming MISB 0601 or 0104 standards) and convert metadata pertaining to the camera location and orientation into 3D visualization of these cameras. This is useful for checking if a video has camera pose metadata and validating that the cameras are pointing at the correct location. This visual check can help find gross errors in metadata that are difficult to interpret by just looking at the raw metadata values.

A metadata panel on the side of the user interface shows all KLV fields associated with the current frame. Values in this panel will update as the video is played or scrubbed. If no metadata is available for a given frame the values are all set to “(not available)”. In the example shown below we see a video with MISB 0104 metadata. The platform path is well represented, but this video has only platform orientation and is missing sensor orientation, so the metadata cameras are all rendered pointing up.



*Inspecting Metadata geometrically and with the metadata viewer pane.*

To inspect the metadata, follow the instructions for opening a video. There is no need to create a project file if no further processing is anticipated. When a video is opened it is automatically scanned for all metadata to build the 3D representation of the cameras. Scanning a video for metadata may take

several seconds or even minutes for very large videos. While TeleSculptor is scanning the video one can still playback the video or seek with the frame selection slider to preview the imagery and metadata values.

### SketchUp Enhancement

The goal of this workflow is to estimate accurate camera models for keyframes in the FMV and then import those keyframes and camera models into SketchUp to use the Match Photo interface to build 3D models in SketchUp by drawing directly on the images. The SketchUp workflow is described in a separate document. This document describes how to use TeleSculptor to produce the data that will be loaded into SketchUp. The processing steps are as follows

1. Create a New Project
2. Import a Video
3. Track Features
4. Estimate Cameras/Landmarks
5. Save Frames
6. Set Ground Control Points

### Dense Automated 3D Models

The goal of this workflow is to automatically estimate a dense 3D model of the scene without using SketchUp. The processing pipeline builds on the estimated cameras used in the SketchUp workflow. It estimates dense depth maps from multiple different viewing directions and then fuses the depth maps together into a unified 3D surface mesh. The processing steps are as follows

1. Create a New Project
2. Import a Video
3. Track Features
4. Estimate Cameras/Landmarks
5. Batch Compute Depth Maps
6. Fuse Depth Maps
7. Colorize Surface
8. Export Fused Mesh

If running both the SketchUp Enhancement and Dense Automated 3D Models workflows on the same video, there is no need to repeat the common steps (1-4). Simply run steps 5-8 on the existing camera and landmark estimation. Optionally, selecting a smaller 3D region of interest (ROI) before step 6 or 7 will limit the region of computation and compute results more quickly. By selecting “Run End-to-End” in the compute menu, TeleSculptor will automatically run steps 3-6 one after the other.

## Processing Steps

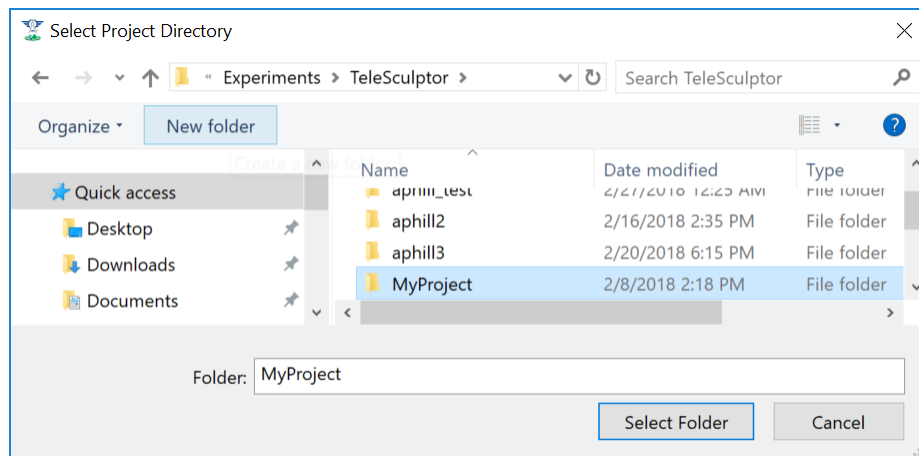
This section describes how to run the key processing steps and what each step does. The previous section describes which of these steps you should run, and in which order, depending on the desired goals. However, processing steps are generally run in the order listed below, with some steps only needed for one workflow or another.

### Create a New Project

The TeleSculptor application requires a working directory, also called the project folder, in which to save settings and algorithm results when processing a video. To create a new project use the *File->New Project* menu item or keyboard shortcut Ctrl+N. Create a new empty folder at the desired location and press the “Select Folder” button with that new folder highlighted.



*Create a new project from the File->New Project menu item.*



*Create a new project folder to store the results of processing.*

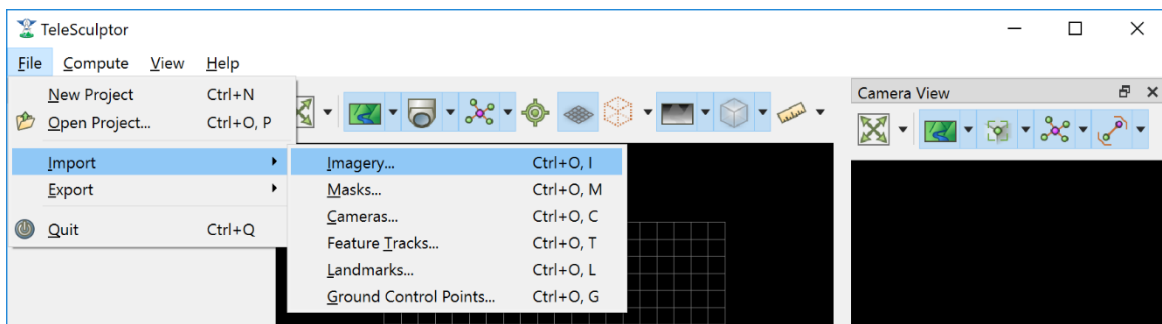
After creating the new project, the application will create a project file in the project directory with the same name and a “.conf” file extension. In the example shown in the figure above it will create **MyProject/MyProject.conf**. This conf file is known as the project file or the project configuration file. It contains the configuration settings for the project. To reload an existing project after closing the application, use *File->Open* and select this project configuration file.

The user must create or open a project before running any of the tools in the Compute menu. The application can open videos and other files for inspection without a project but cannot process the data without a project. The recommended workflow is to create a project first and then open a video, however, if the project is created after opening a video the open video will be added to the new project.

Projects can have any name and refer to a video file at any location on the computer. That said, for better organization we recommend giving the project folder the same name video clip that will be processed. If it is possible to move the video clip into the new project folder *before* opening the video the project will be relocatable. That is, if the source video is in the project folder you can move the project folder to another location or another computer and it will still load. If the video is outside the project folder the absolute path will be recorded, but this link could become broken if the video or project is moved later.

## Import a Video

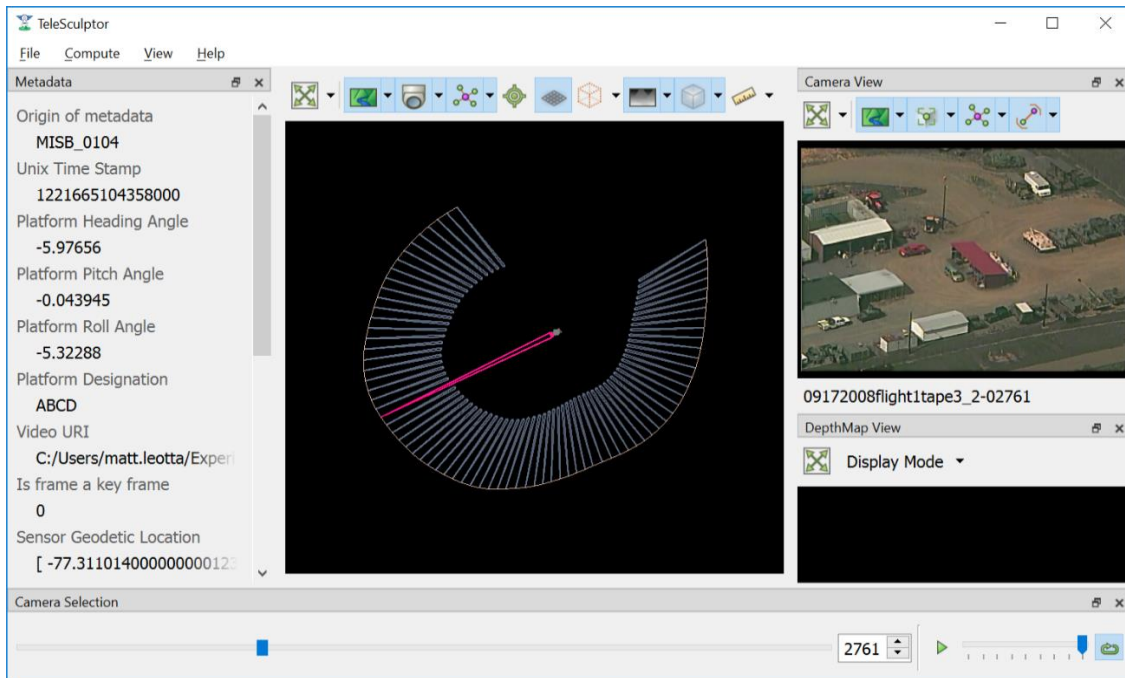
To import a video clip, use the *File->Import->Imagery* menu item. In the Open File dialog box browse to select the file to open and then press the Open button. This same menu item can be used to open a list of image paths in a text file. Advanced users can also open intermediate data files, like cameras and landmarks, from the import menu, but these use cases are not covered in this document. Masks are another special type of import. These are also a video or image list but contain black and white images with black indicating which pixels of the image to ignore. Mask images are particularly useful for videos with burned-in metadata as a heads up display (HUD). Kitware has a separate tool call Burn-Out for estimating these mask videos for videos with burned in metadata.



*Import a video clip, image list, or other data.*

Once a video is selected to open, the application will scan the entire video to find all metadata. This may take several seconds or even minutes for very large videos. If scanning the video takes too long or it is known that the video does not contain metadata it is okay to cancel the metadata scanning using the “Cancel” option in the Compute menu. If metadata scanning is canceled, then no metadata will be used in any subsequent processing steps.

The first video frame will appear in the Camera View in the upper right. The World View (left) will show a 3D representation of the camera path and camera viewing frustums if relevant metadata is found. When images are shown in the 3D world view, they are always projected using the active camera model onto this ground plane (indicated by the grid). The image button (🌐) toggles visibility of images in each view with an opacity slider in the drop-down menu below the button. For images to appear in the world view an active camera model is required. An active camera model is a camera model for the currently selected video frame.



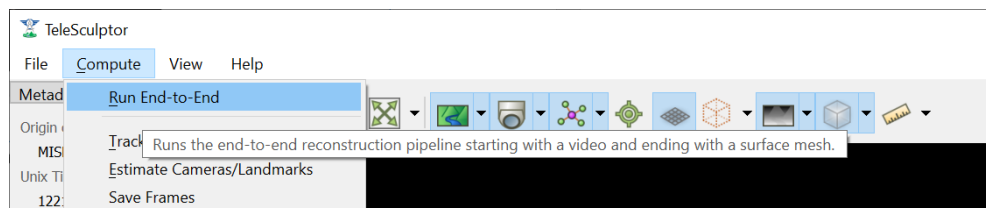
*A video with KLV metadata opened in the application.*

As can be seen in the image above. The aircraft flight path, as given in the metadata, is shown as a curved line in the 3D World View. A pyramid (frustum) is shown representing the orientation and position of the camera at each time step. The active camera, representing the current frame of video, is shown in a different color and is longer than the others. If the video is played with the play button in the lower right, the video frames will play in the Camera View and the active camera will updated in the World View.

The visibility of cameras is controlled by the cameras button (📷) above the world view. In the drop-down menu underneath the cameras button you can individually set the visibility, size, and color of camera path, camera frustums, and active camera.

## Run End-to-End

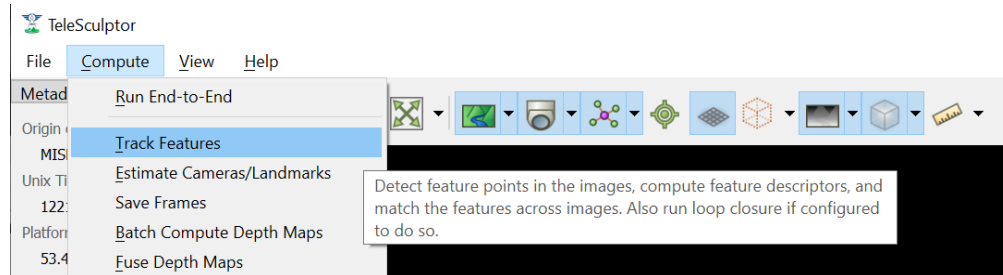
Run End-to-End is a new feature in TeleSculptor v1.1.0. Rather than waiting for user input to run each of the primary processing steps, Run End-to-End automatically runs Track Features, Estimate Cameras/Landmarks, Batch Compute Depth Maps, and Fuse Depth Maps. These steps are run sequential in this order. See details of these steps in their respective sections below.




*Run End-to-End Option in the Compute menu.*

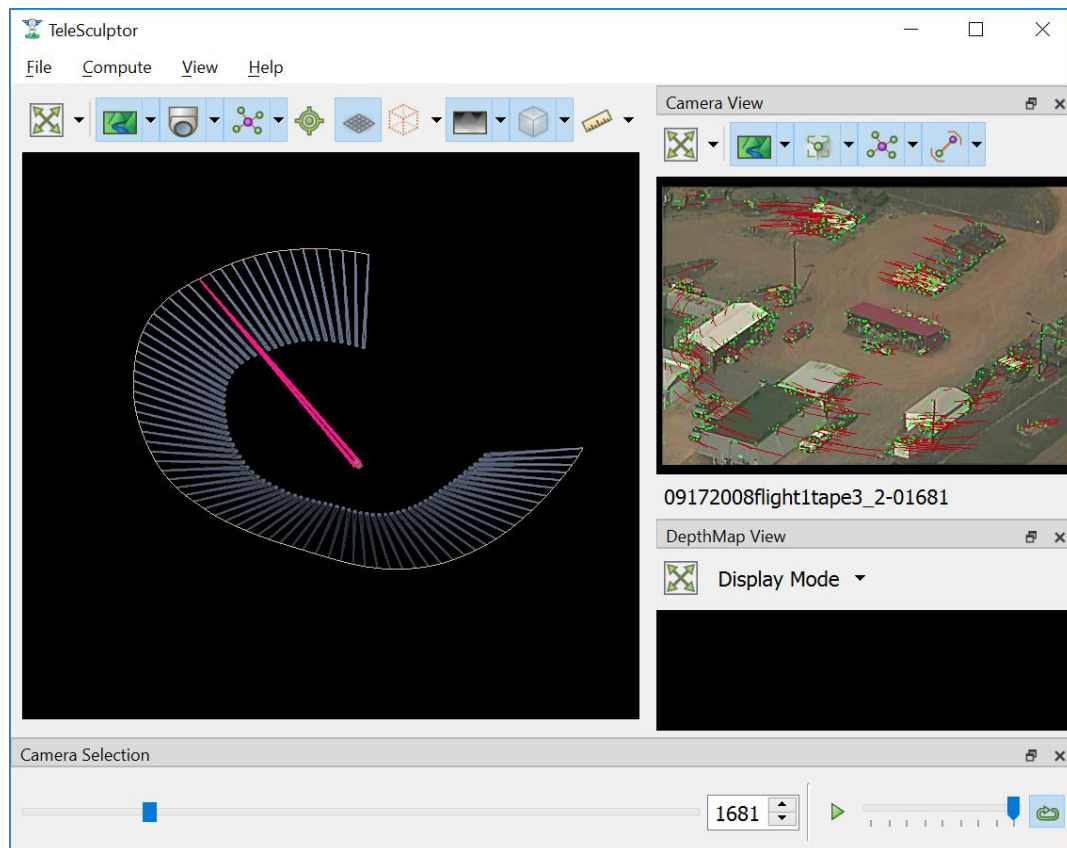
## Track Features

The Track Features tool detects visually distinct points in the image called “features” and tracks the motion of those feature points through the video. This tool is run from the *Compute* menu and is enabled after creating a project and loading a video.



*Run the Track Features algorithm from the Compute menu.*

When the tool is running it will draw feature points on the current frame and slowly play through the video tracking the motion of those points as it goes. These tracks are visualized as red trails in the image below. These colors are fully customizable. The feature tracks button (  ) above the Camera View enables toggling the visibility of tracks. The drop-down menu under this button has settings for the display color and size of the feature track points and their motion trails.



*The Track Features algorithm producing tracks on a video.*

The feature tracking tool will start processing on the active frame and will run until the video is complete or until the tool is canceled with the *Cancel* option in the *Compute* menu. We recommend

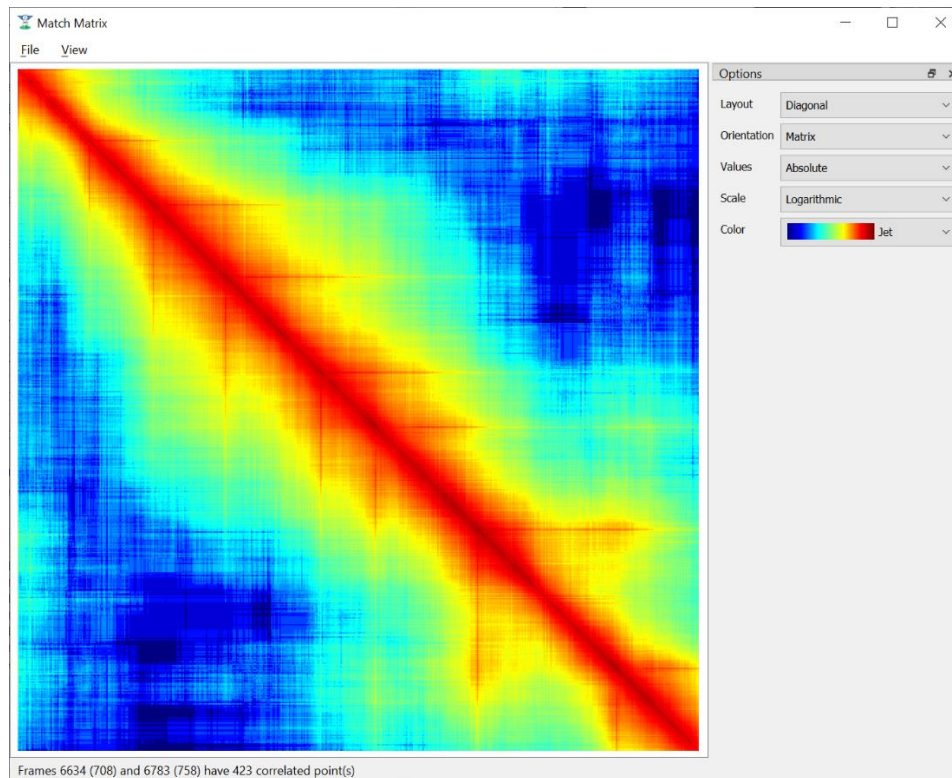


starting with the video on frame 1 and letting the algorithm process until complete for a video clip containing approximately one orbit of the UAV above the scene. However, it is possible to process subsets of the video by scrubbing to a desired start frame before running the tool and then hitting the cancel button after reaching the desired end frame.

When this, or any other, tool is running all other tools will be disabled in the *Compute* menu until the tool completes. Most tools also support exiting early with the *Cancel* button to stop at a partial or suboptimal solution.

Note that to limit redundant computation this tool does not track features on every frame of video for long videos. Instead the algorithm sets a maximum of 500 frames (configurable in the configuration files) and if the video contains more than 500 frames it selects 500 frames evenly distributed throughout the video. Once feature tracking is done, tracks will flicker in and out when playing back the video due to frames with no tracking data. To prevent this flickering select *Tracked Frames Only* from the *View* menu. With this option enabled, playback is limited to frames which include tracking data.

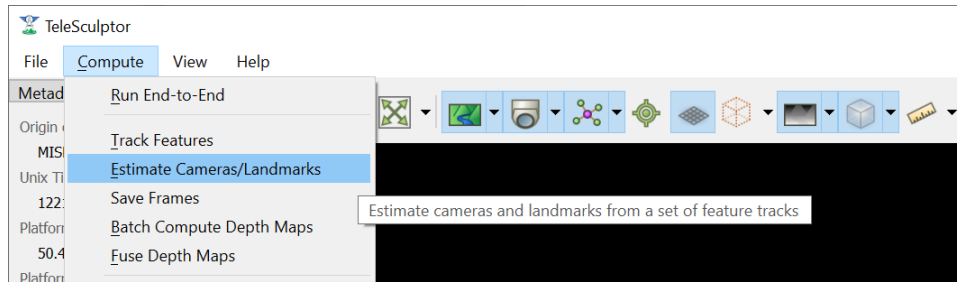
More technical users who want to understand the quality of feature tracking results may wish to use the *Match Matrix* viewer under the *View* menu (keyboard shortcut M). The match matrix is a symmetric square matrix such that the value in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column is the number of features in correspondence between frames  $i$  and  $j$ . The magnitude of these value is colored with a color map and placing the mouse cursor over a pixel prints the actual number of matches in the status bar. Typically, a match matrix has strong response down the diagonal (nearby frames) that drops off as you move away from the diagonal. Flight paths that make a complete orbit should see a second weaker off-diagonal band where the camera returns to see the same view again.



*The Match Matrix view of feature tracking results.*

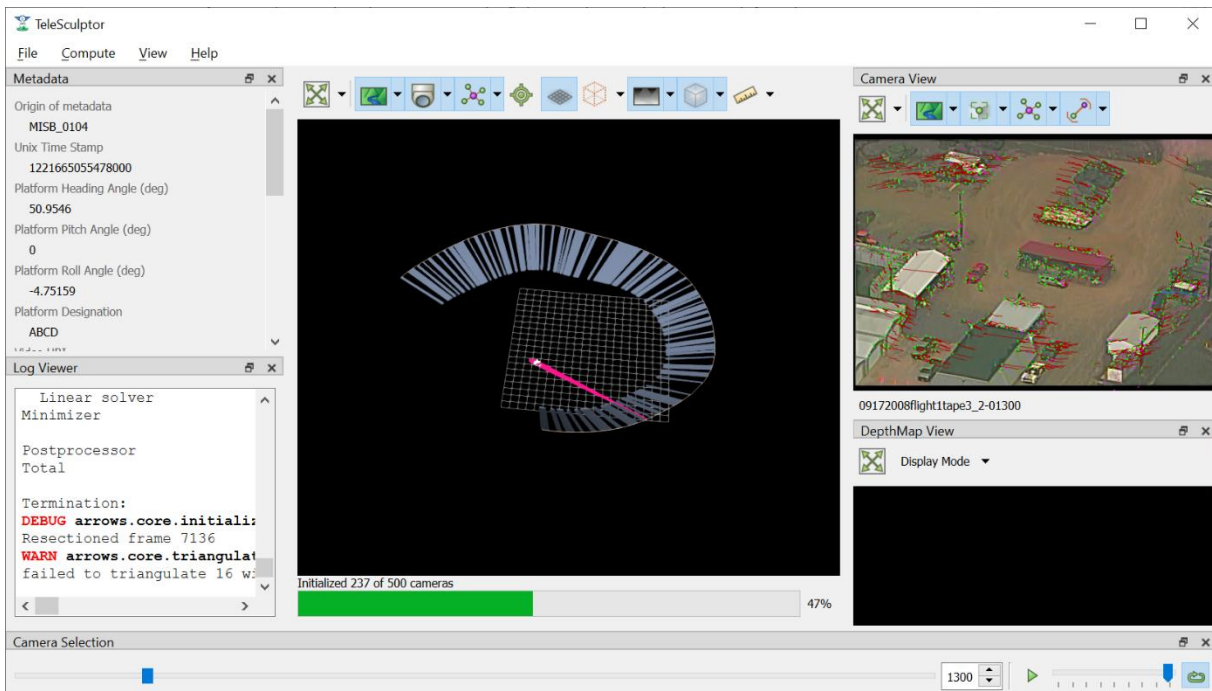
## Estimate Cameras/Landmarks

The Estimate Cameras/Landmarks tool in the Compute menu uses structure-from-motion algorithms to estimate the initial pose of cameras and the initial placement of 3D landmarks. It also uses bundle adjustment to jointly optimize both cameras and landmarks. These algorithms use the camera metadata as constraints and initial conditions when available. The algorithm will try to estimate a camera for every frame that was tracked and a landmark for every feature track.



*Run the Estimate Cameras/Landmarks algorithm from the Compute menu.*

The solution will start with a sparse set of cameras and then incrementally add more. Live updates will show progress in the world view display. During optimization, the landmarks will appear to float above (or below) the ground plane grid because the true elevation is typically not near zero. Once the optimization is complete, a local ground height is estimated, and the ground plane grid is moved to meet the landmarks. This ground elevation offset is recorded as part of the geo-registration local coordinate system.



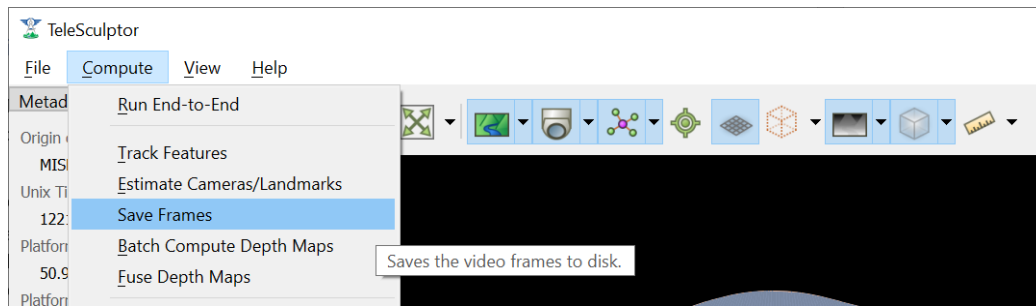
*Estimated landmarks, colored by height, shown in both the world and camera views.*

Once landmarks are computed their visibility can be toggled with the landmarks button (📍) in both the world and camera views. The drop-down menu under the landmarks button allows changing the size and color of the landmarks including color by height or by number of observations.



## Save Frames

The *Save Frames* tool is quite simple. It simply steps through the video and writes each frame of video to disk as an image file. These image files are stored in a subdirectory of the project directory. Saving frames only requires an open video and an active project. It can be run at any time. Like the feature tracking tool, it plays through the video as it processes the data and can be cancelled to stop early. The primary purpose for saving frames is for using them in the TeleSculptor / SketchUp workflow. SketchUp can only load image images, not video. So, this step produces the image files that are needed when loading a project file into SketchUp.

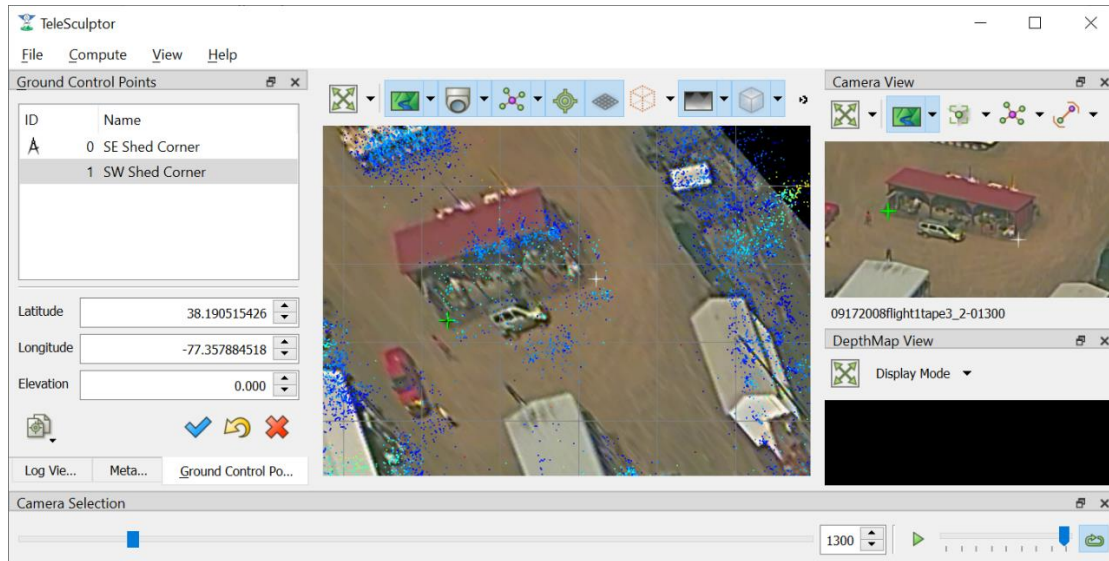


*Save the loaded video frame as image files on disk*

## Set Ground Control Points

Ground Control Points (GCPs) are user specified 3D markers that may be manually added to the scene for a variety of purposes. These markers are completely optional features that may be used to provide meaningful guide points for modeling when exporting to SketchUp. GCPs are also used to estimate geo-localization of videos with metadata or to improve geo-location for video with metadata.

To add GCPs press the GCP button (📍) above the 3D World View. To create a new GCP hold the *Ctrl* key on the keyboard and left click in either the 3D World View or the 2D Camera View. A new GCP will appear as a green cross in both views. Initial points are currently dropped into the scene along the view ray under the mouse cursor at the depth of the closest scene structure. If the initial depth of a GCP is not accurate enough it can be moved. To move a point, left click on the point in either view and drag it. Points will always move in a plane parallel to the image plane (or current view plane). It helps to rotate the 3D viewpoint or scrub the video to different camera viewpoints to correct the position along different axes. Holding *Shift* while clicking and dragging limits motion to single coordinate axis. The axis of motion is the direction which has the most motion in the initial mouse movement. Once additional points are added (with *Ctrl* + left clicks) the active point is always shown in green while the rest are shown in white. Left clicking on any point makes it active. Hitting the *Delete* key will delete the current active GCP.

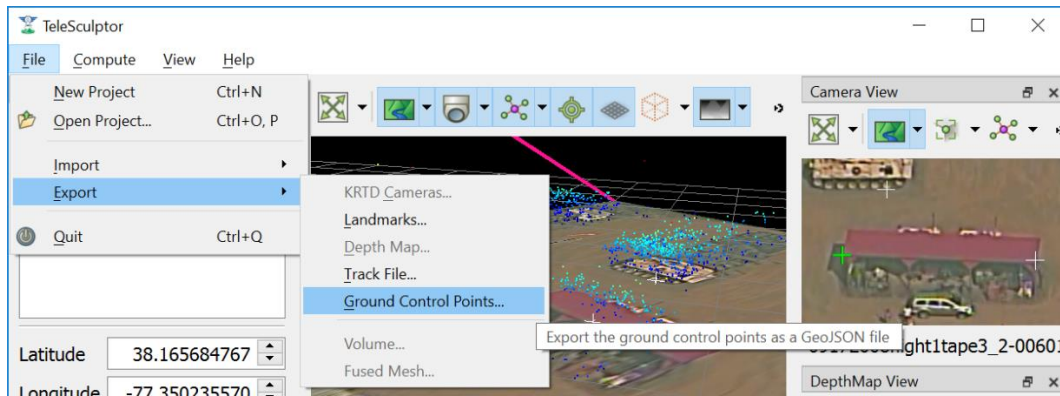


*Setting ground control points, the active point is shown in green.*

The Ground Control Points pane provides a way to select and manage the added GCPs. The pane lists all added points and allows the user to optionally assign a name to each. The GCP pane also shows the geodetic coordinates of the active point, and these points can be copied to the clipboard in different formats using the copy location button (📋). If the value of the geodetic coordinates is changed that GCP becomes a constraint and is marked with an icon (Ⓐ) in the GCP list. Constrained points will keep fixed geodetic coordinates when the GCP is moved in the world space. A constraint can be removed by pressing the reset button (↺). Once at least three GCP constraints are added with geodetic coordinates, the apply button (✓) can be used to estimate and apply a transformation to geolocalize the data. While three GCPs are the minimum, five or more are recommended. The transformation will be fit to all GCP constraints. After applying the GCPs, all cameras, landmarks, depth maps, and GCPs are transformed to the new geographic coordinates. Currently the mesh fusion is not transformed because the integration volume is axis-aligned. Instead the fusion results are cleared and need to be recomputed.

It is often helpful to compute a depth map (or even fused 3D model) before setting GCPs to provide additional spatial reference in the 3D view. It is possible set GCPs entirely from the 2D camera views by switching between video frames and correcting the position in each. However, this is more tedious. When the 3D position is correct the GCP should stick to the same object location as the video is played back.

GCPs are currently not saved automatically. To save the GCP state use the *File->Export->Ground Control Points* menu option and create a PLY file to write. This file path is cached in the project configuration, so GCPs are automatically loaded when the project is opened again. They are also automatically loaded when importing the project configuration into SketchUp.

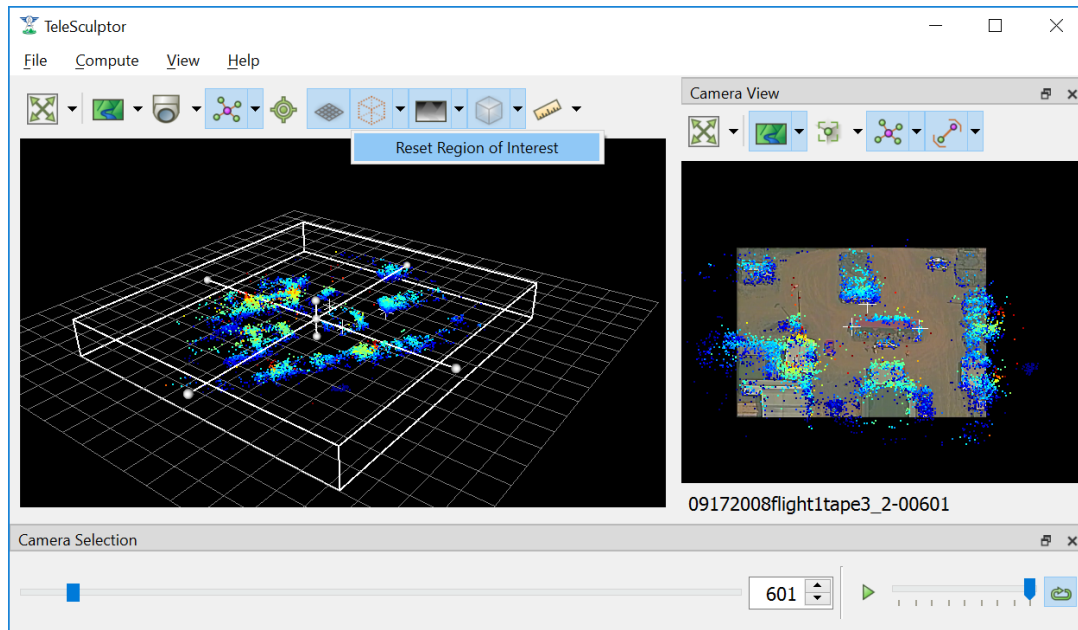


*Export the ground control points.*

### Set 3D Region of Interest

Before running dense 3D modeling operations, it is beneficial to set a 3D region of interest (ROI) around the portion of the scene that is of interest. This step is optional. By default, a ROI is chosen to enclose most of the 3D landmark point that were computed in the triangulate landmarks step. Some outlier points are rejected when fitting the ROI and the estimated ROI is padded to account for missing data. The default ROI is generally sufficient for further processing, but may be larger than necessary. The advantage of picking a smaller ROI is a significant reduction in compute time and resources. Furthermore, the quality and resolution of the result often improves when focusing on smaller subset of a large scene because we can focus more compute resources on that location.

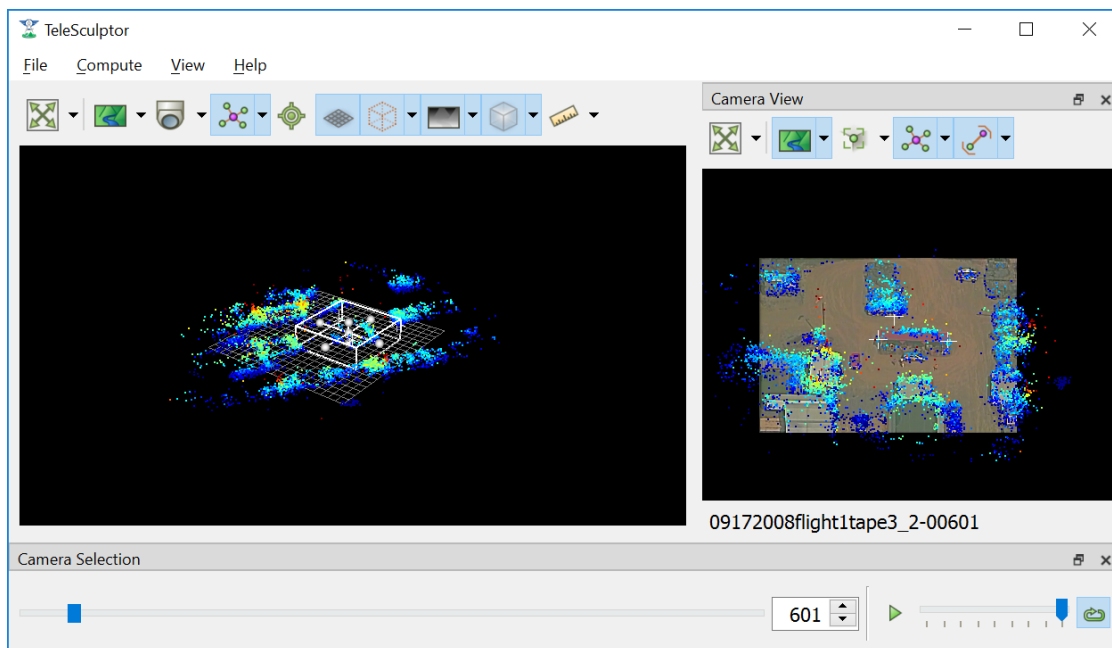
To see and manipulate the 3D ROI click the 3D ROI button (📦) above the 3D World View. A 3D axis-aligned box is shown which contains the set of 3D landmarks. Inside the box are axis lines along the center of the box in each of the three coordinate directions. At the ends of these lines are spheres which act as manipulation handles. Left click on any of these handles and drag to reposition the corresponding face of the box in 3D. Left click the sphere at the center of the box and drag to translate the entire box. A middle click (or *Ctrl* + left click or *Shift* + left click) and drag anywhere in the box has the same translation effect as using the center handle. A right click and drag will scale the box uniformly about its origin. Note that the ground plane grid will adjust size relative to the ROI size.



*The default 3D bounding box is fit to the sparse landmarks and is often bigger than needed.*

A good practice is to set the bottom of the ROI box just below the ground and the top just above the tallest part of the structure. Likewise, set the sides to be just a bit outside the object of interest. It may be difficult to determine the bounds accurately from the sparse landmarks. A good strategy is to start with a slightly larger guess, then use the *Compute Single Depth Map* advanced tool to compute a single depth map. The depth map gives more detail which helps pick a tighter box. Then use *Batch Compute Depth Maps* to compute the additional depth maps with the revised box.

To reset the ROI to the initial estimated bounding box, use the *Reset Region of Interest* option in the drop-down menu under the ROI button.

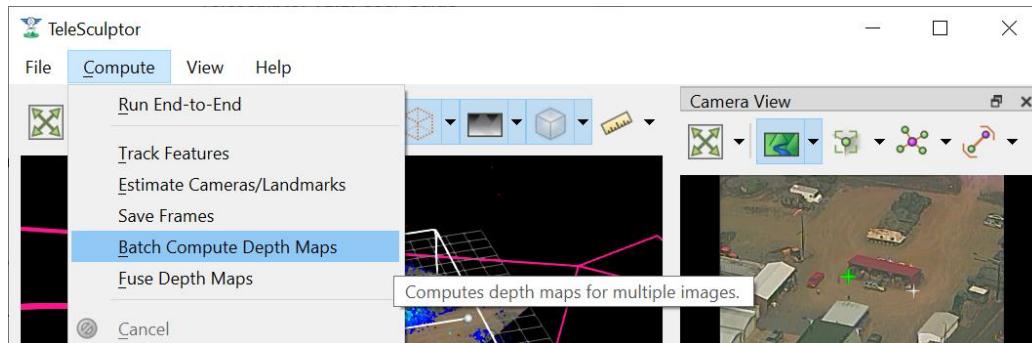


*Setting a tighter bounding box around one structure of interest.*


## Batch Compute Depth Maps

Computation of dense depth maps is part of the fully automated 3D reconstruction pipeline. Several depth maps are needed to compute a final 3D result. Running the *Batch Compute Depth Map* tool from the *Compute* menu will estimate depth maps (2.5D models) for twenty different frames sampled evenly through the video. To compute a depth map only on the current frame, see the *Compute Single Depth Map* option in the advanced menu.

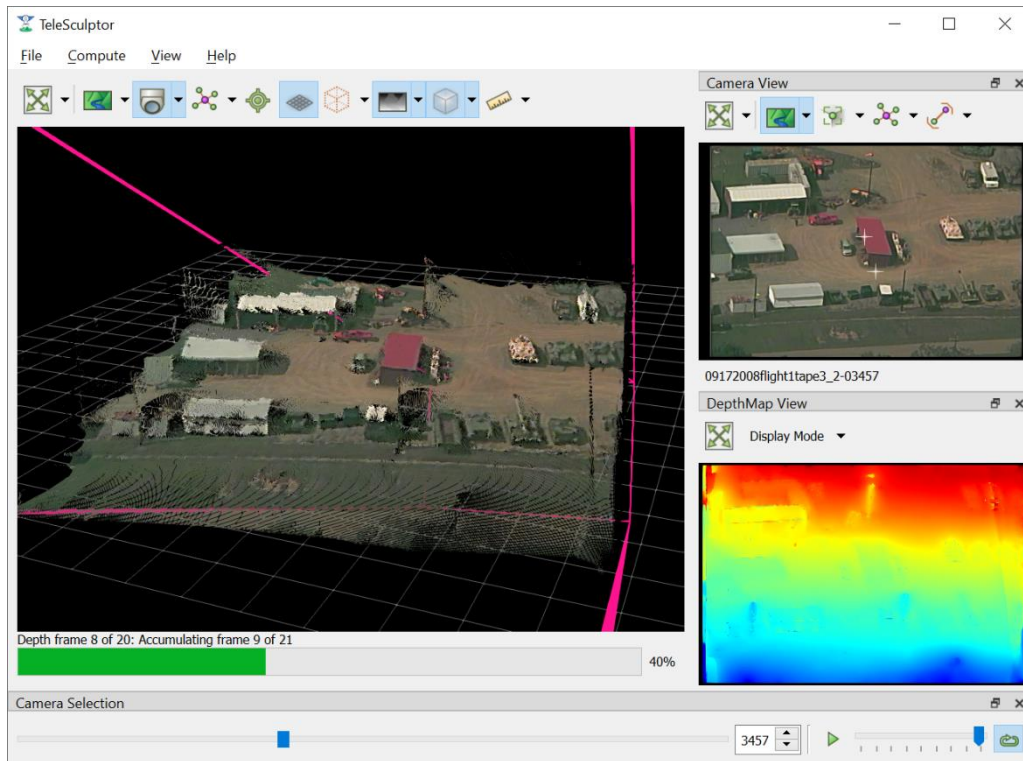
This algorithm requires very accurate camera and landmark data resulting from the previous *Estimate Cameras/Landmarks* step above. Furthermore, cameras models on multiple frames in nearby positions are required. By default, the algorithm uses the ten frames before and ten frames after each selected depth frame for reference.



*Compute dense depth maps on key frames*

The results of depth map estimation are shown in two ways. In the world view the depth maps are shown as a dense colored point cloud in which every image pixel is back projected into 3D at the estimated depth. Use the Depth Map button (  ) to toggle depth point cloud visibility. The second way depth maps are visualized is as a depth image in the Depth Map View. Here each pixel is color coded by depth and the color mapping is configurable.

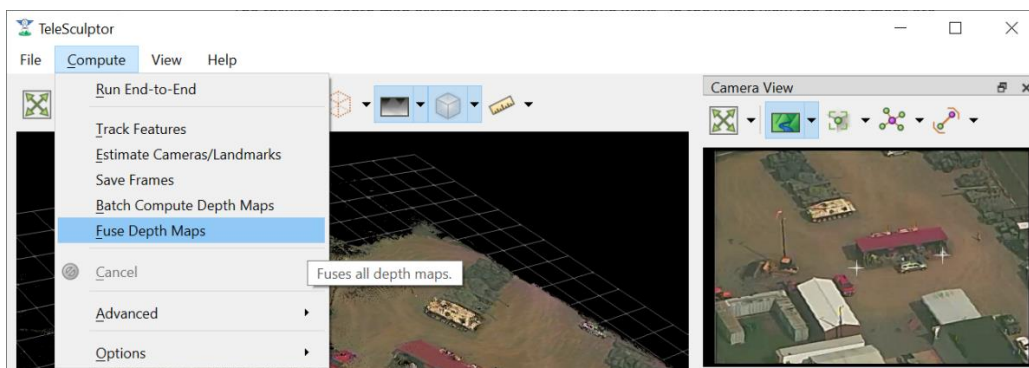




*Results of depth map computation*

## Fuse Depth Maps

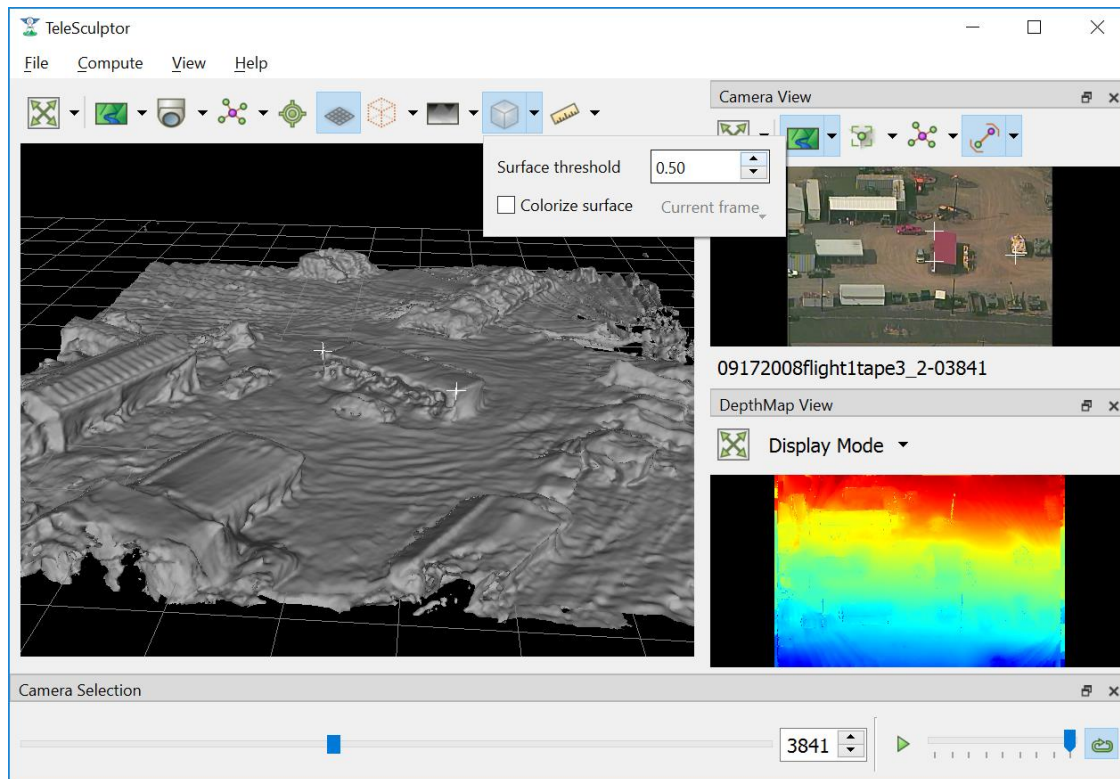
After computing depth maps in batch, or manually computing multiple depth maps, the next step is to fuse them into a consistent 3D surface. Running *Fuse Depth Maps* from the *Compute* menu will build an integration volume and project all depth maps into it for fusion. This integration step requires a modern CUDA capable Nvidia GPU (Requires at least Nvidia driver version 396.26). The size of the integration volume and the ROI covered is determined by the same ROI box used in the depth map computation. This processing step runs in only a few seconds and may cause lag in the display during this time due to consumption of GPU resources. Once the data is fused into a volume, a mesh surface is extracted from the volume.



*Fuse the depth maps into a consistent mesh.*

To toggle the view of the fused mesh, press the Volume Display button (📦) above the 3D World View. The surface mesh can be fine-tuned if desired by adjusting the surface threshold in the drop-down menu

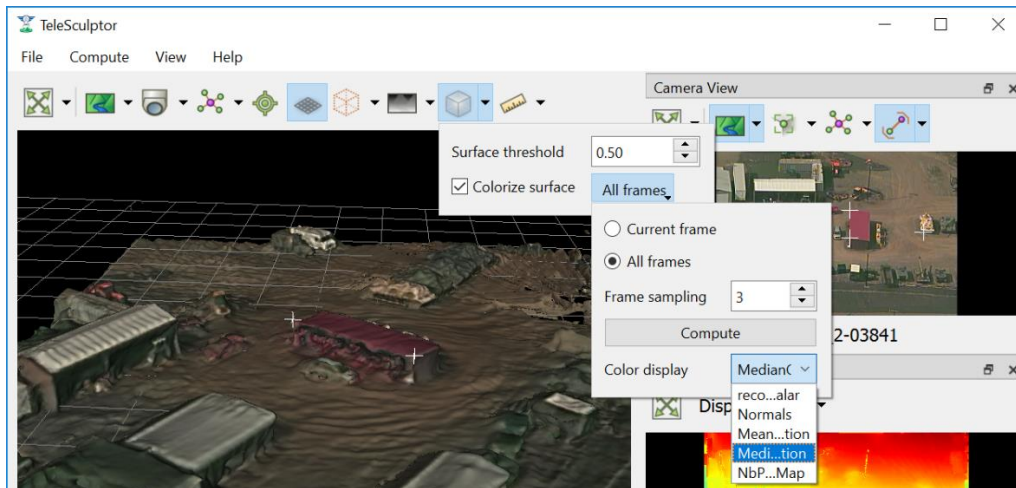
under the Volume Display button. Setting the threshold slightly positive (e.g. 0.5) often helps to remove unwanted outlier surfaces that tend to appear in areas with only a few views.



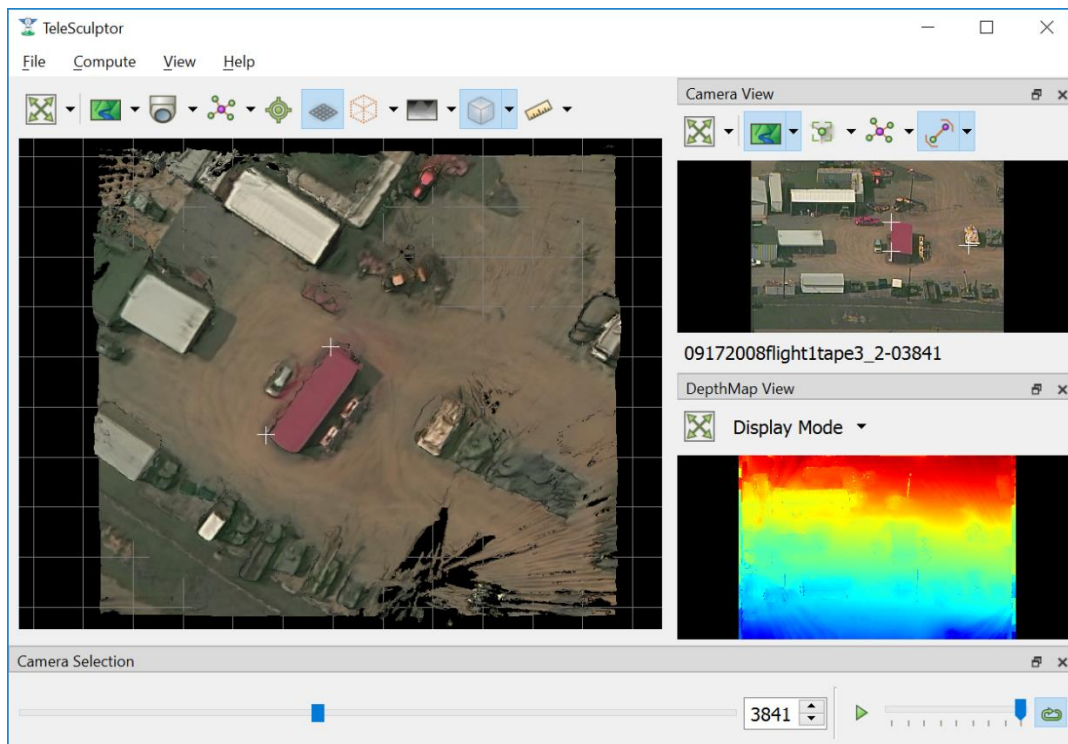
View the fused mesh, adjust surface threshold if desired.

### Colorize Mesh

The fused mesh is provided initially in a solid grey color. To add color, use the drop down menu under the Volume Display button ( ). Check the *Colorize surface* box to enable color. There are two options to color the mesh. The *Current frame* option always projects the current frame onto the mesh and the color updates when you play back the video. The *All frames* option estimates a static mesh coloring by projecting multiple images onto the surface and combining them. The *Frame sampling* combo box allows configuration of how frequently to sample frames for coloring. Smaller sampling uses more frames for better color but more computation time. Press the Compute button to compute mesh color (note: a progress bar is not yet implemented for this step). When complete, the *Color display* option can be changed without needing to recompute color. The recommended color display option is *MedianColoration*, however, *MeanColoration* is often quite similar. There are also special colorization options to gain insight into the data. The *Normals* option colors the mesh by surface normal direction, and the *NbProjectedDepthMap* option colors by the number of depth map views that observed each part of the surface.

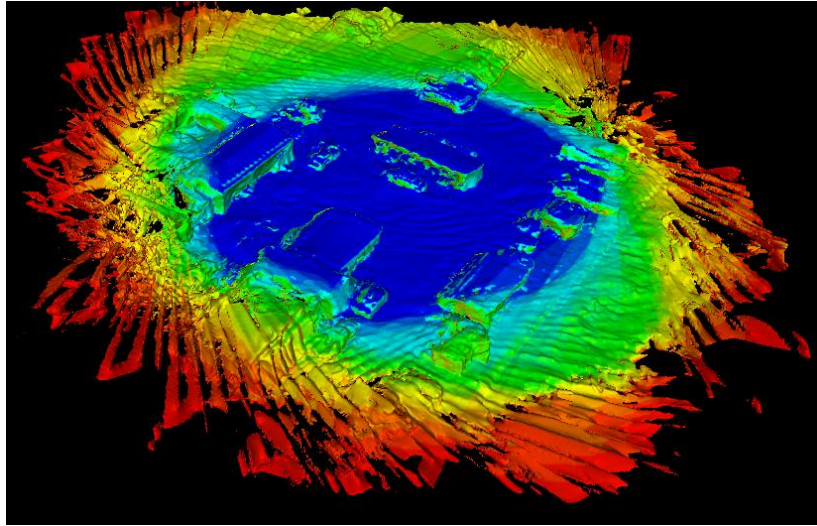


*The mesh colorization menu options.*



*A fused mesh colored by the mean of multiple frames.*

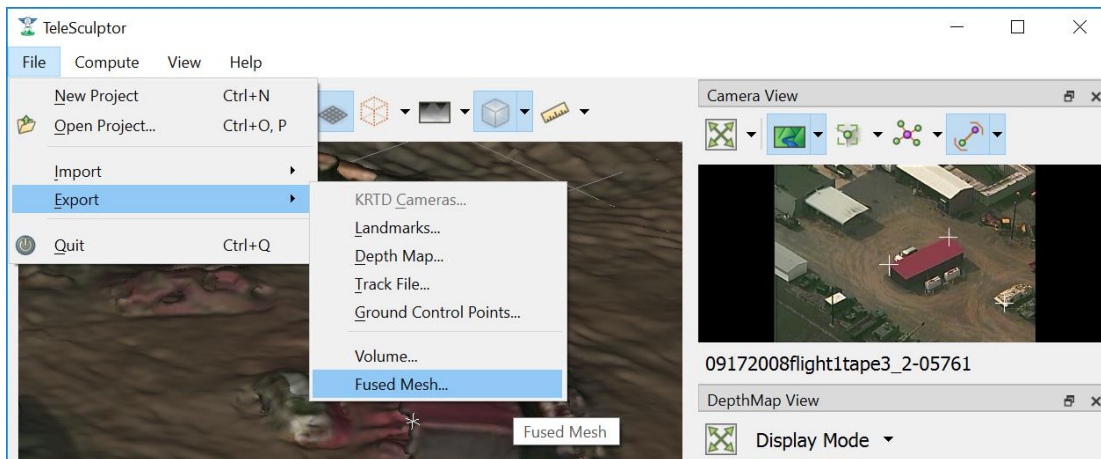




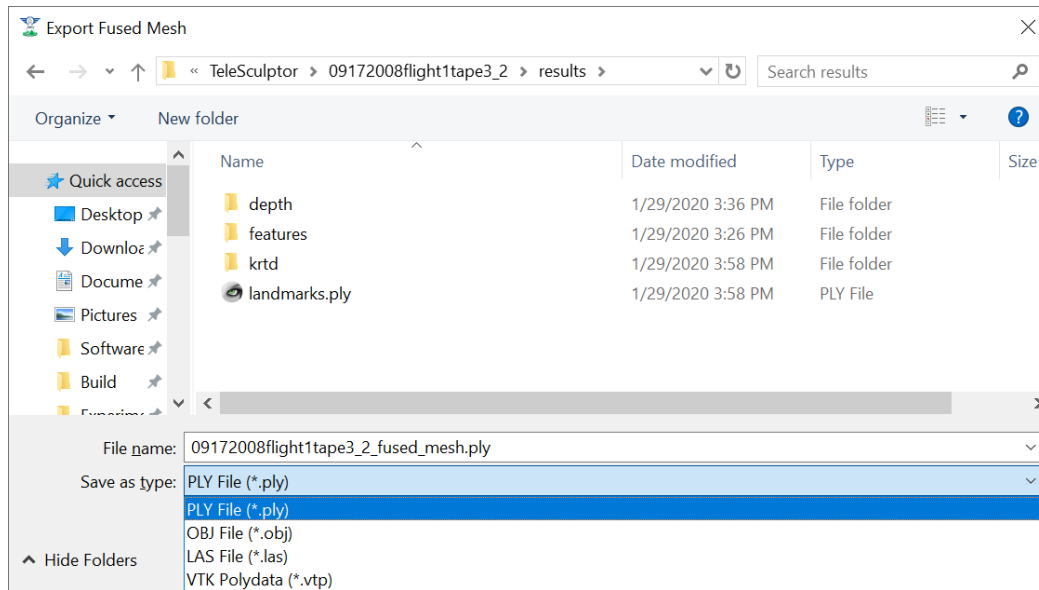
*A fused mesh colored by the number of views that see each part of the surface.*

## Export Data

To export the finale colorized mesh for use in other software, use the *File->Export->Fused Mesh* menu item. This will provide a file dialog to save the model as a mesh in standard PLY, OBJ, LAS, or VTP file formats. The LAS file format will only save the dense mesh vertices as a point cloud and does include geo-graphic coordinates. The other formats save the surface mesh but only in local coordinates. Note that all formats (except OBJ) will also save RGB color on the mesh vertices. This color matches whatever display options are currently set.

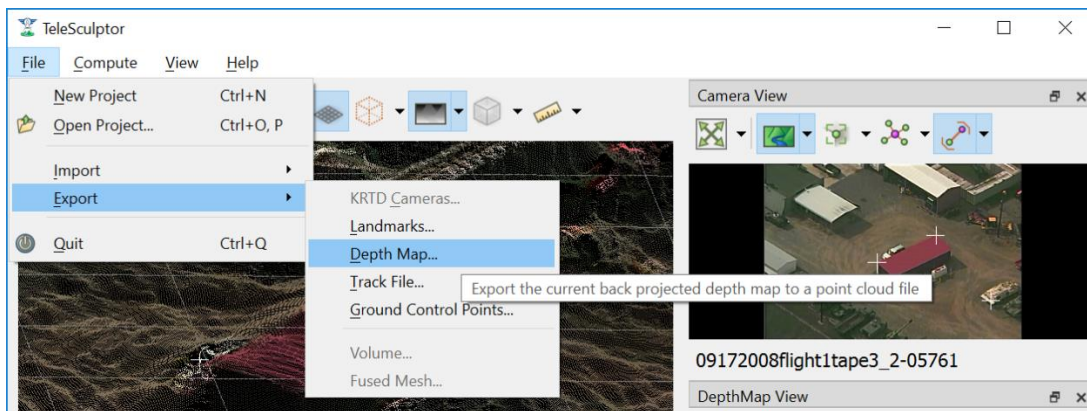


*Save a colorized mesh as a PLY, LAS, or VTP file.*




*Change the Save as type to export as LAS.*

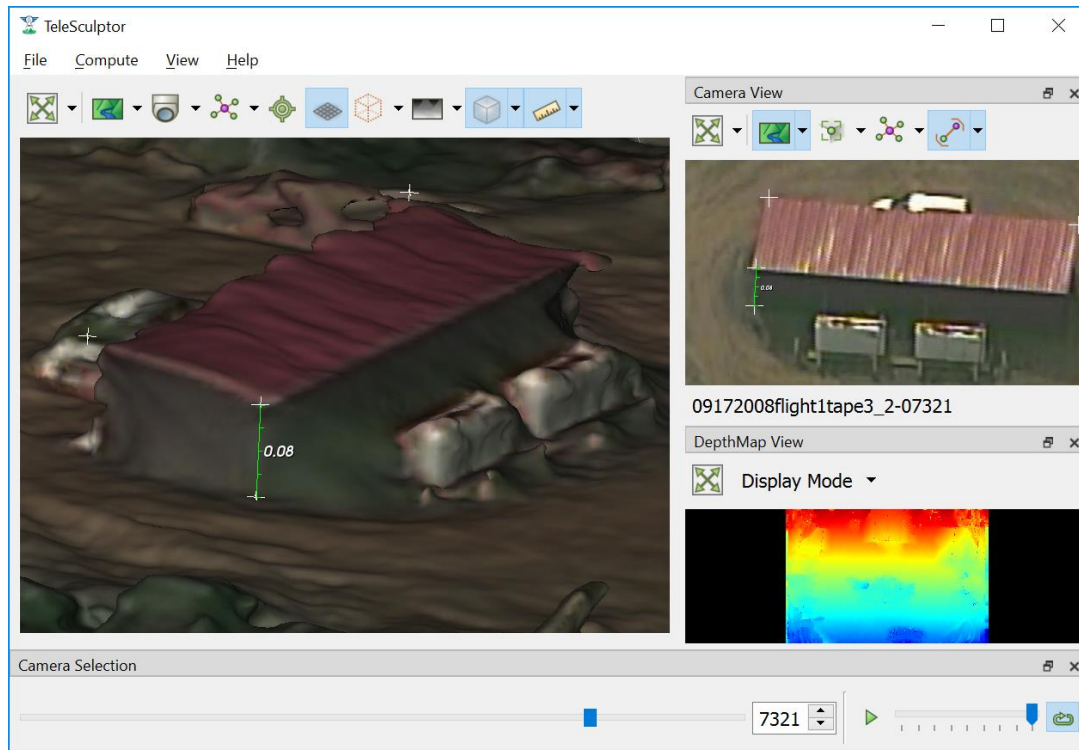
To export the active 2.5D depth map for use in other software, use the *File->Export->Depth Map* menu item. This will provide a file dialog to save the model as an RGB colored point cloud in the standard PLY or LAS file formats.



*Save a computed depth map as a point cloud in PLY or LAS formats.*

## Measurement Tool

The measurement tool () allows the user to measure straight line distance in world coordinates. Placing the end points of the ruler uses a similar interface to placing GCPs, and each end point can be adjusted independently just like GCPs. The number displayed next to the green line in both world and camera views represents the distance in world space. If geo-spatial metadata is provided the measurements are in units of meters. Without metadata (as in the example below) the measurements are unitless. The ruler can be drawn and adjusted in either the world or camera views. Often it is easier to get more accurate alignment in the image space. As with GCPs, if the ruler sticks to the correct location when playing back the video then the 3D coordinates are correct.



*Measuring a building height with the measurement tool.*

When measuring it is sometimes convenient to constraint the measurements to the horizontal or vertical directions. After the initial ruler is placed in the scene, click and drag one end point. If the Z key is held on the keyboard the moving point will be constrained to lie on a vertical axis through the point at the other end of the ruler. If either the X or Y keys is held, the moving point will be constrained to lie in a horizontal (X-Y) plane that passes through the other ruler point. Each of these constraints is indicated by an indicator as shown below.



*Vertical Constraint (hold Z)*



*Horizontal Constraint (hold X or Y)*

## Advanced Tools

The following tools are under the Compute->Advanced menu. Most users should not need these tools, but they may come in handy on challenging data sets where the normal compute steps do not work ideally.

Filter Tracks  
 Triangulate Landmarks  
 Refine Solution  
 Reverse (Necker)  
 Align  
 Save Key Frames  
 Compute Single Depth Map

### Filter Tracks

This tool filters the set of tracks to find a reduced set of tracks that spans the same frame range. It tries to keep the longest, most stable tracks and throws out many short tracks. The goal is to make bundle adjustment more efficient by limiting the solution to the most important tracks. Since we now limit how many features are tracked to begin with, this tool does not usually provide a benefit.

### Triangulate Landmarks

*Triangulate Landmarks* attempts to create a 3D landmark position for each feature track by back-projecting a ray through the feature point locations in each image and intersecting these rays in 3D space. The triangulation is fast and should finish almost instantly, however it requires accurate camera models to work. The *Triangulate Landmarks* tool requires both feature tracks and cameras to work.

If the metadata for camera poses is very accurate one can bypass the Estimate Cameras/Landmarks step and directly triangulate landmarks. The triangulated positions will likely still be very noisy, but this can be improved by bundle adjustment using the *Refine Solution* tool below.

### Refine Solution

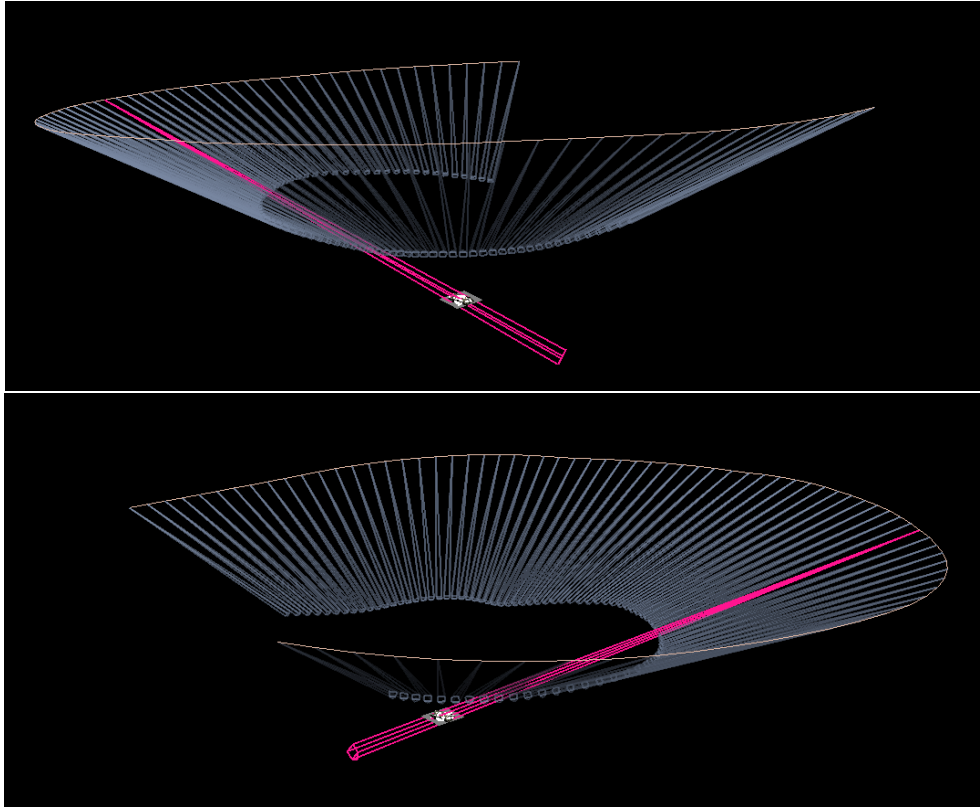
Refining the solution optimizes the calibration and pose of the cameras as well as the locations of the 3D landmarks in a process known as bundle adjustment. This tool requires tracks, landmarks, and cameras to run. It adjusts the parameters such that the bundle of rays used to triangulate each point meets more precisely at a single location. Bundle adjustment is already run as part of *Estimate Cameras/Landmarks*, but this option allows it to be run directly. Direct refinement is useful for features that are directly triangulated from metadata cameras.

Running this algorithm can take some time. While it is running, the solution incrementally improves, and updated results are displayed as the solution evolves. Bundle adjustment tends to make very large corrections very quickly and then spend lots of time fine tuning the final solution to get it just right. The cancel option allows the user to exit the optimization early and keep the current state of progress. Cancelling is useful when the solution is taking too long to complete but appears to have found a reasonable solution. It helps get a solution more quickly but beware that a suboptimal solution will impact the quality of later 3D reconstruction stages. It is better to wait for completion when time is available.

### Reverse (Necker)

There is a special type of failure mode in camera calibration that only happens with very long focal length cameras. This failure mode happens because of a depth reversal ambiguity that occurs when perspective distortion is lost and the projection is nearly orthographic. The solution is bistable, just like to famous Necker Cube optical illusion. Under this “Necker Reversal” the one can invert the height of landmarks and flip cameras upside down and mirror them across the orbit to produce nearly identical

geometric projection. The *Reverse (Necker)* tool flips the data into this alternate configuration to rectify this invalid solution. Note that Necker Reversal is not a problem when geospatial metadata is available or when one can make assumptions about the orientation of the cameras (e.g. up in the world is up in the image). The Initialize Cameras/Landmark tries to automatically detect and correct for this ambiguity, so this the manual correction is rarely needed.



*Effect of Necker Reversal on camera orbits. Bottom cameras are flipped and upside down.*

## Align

The *Align* tool is for videos that do not have metadata. Without metadata the orientation, position, and scale (7 degrees of freedom) of the solution are completely undetermined. The solution floats freely in space. The *Align* tool attempts to align the data to a canonical coordinate system with the following properties: The centroid of the landmarks is aligned to the origin. The direction of minimal landmark variance is aligned with the Z axis with cameras on the positive Z side. The scale is set such that the variance in landmark positions from the origin is one. The origin is shifted along Z such that 90% of landmarks are above  $Z=0$  (ground plane estimation). This algorithm is also now run automatically as part of Initialize Cameras/Landmarks, so this manual tool is rarely needed.

## Save Key Frames

The *Save Key Frames* tool is the same as the Save Frames tool except that it only saves frames that are marked as key frames by the *Track Features* tool. Saving only key frames makes more sense than saving all frames for use in SketchUp. However, the selection of key frames is not currently reliable. Sometimes only one keyframe is selected. One could try this option first and then save all frames if not



enough keyframes are available. To preview the key frames select *Keyframes Only* from the *View* menu and play back the video.

### Compute Single Depth Map

The *Compute Single Depth Map* tool is the same as the *Batch Compute Depth Map* tool except that it only computes on depth map on the current frame. The tool provides live visualization of the intermediate results to visualize how the solution evolves over compute iterations. The initial estimated point cloud will be quite noisy and then will continue to improve with live updates shown in both views as the algorithm progresses. Much like the refine solution tool, the solution improves quickly at first and then spends a longer time fine tuning the solution. The cancel option will also end the optimization early for this tool.

### Compute Options

A few basic switches are now available under the Compute->Options menu to control the behavior of some of the algorithms. Each of these options corresponds to a boolean configuration value in the project file.



Checking these options will save that option in the project file. In the future, we will provide a visual interface to configure many more of the TeleSculptor algorithm options. For now these basic options are presented in the menu:

#### Ignore Metadata

When checked, this option causes the Initialize Cameras/Landmarks algorithm to ignore any metadata that was loaded with the video. This option is useful because it is often the case that the metadata is incorrect and negatively impacts the algorithm rather than helping it.

#### Variable Lens

By default, TeleSculptor assumes that all frames in a video are collected with the same lens and zoom setting. The intrinsic camera parameters are shared across the entire sequence. This assumption gives the best results as long as the assumption holds true. When the assumption does not hold, and the lens zooms or is changed in the middle of the sequence, checking “Variable Lens” will instruct TeleSculptor to estimate unique camera intrinsic parameters for each video frame.

#### Fix Geo-Origin

TeleSculptor automatically selects a local origin near the centroid of the data and records the geographic location of this origin point. When the data is updated by running algorithms that origin point is recalculated and may change. In some cases, there are benefits to specifying the geographic origin to use and keeping it fixed, for example, forcing two data sets to share a common local origin for easier comparison. Checking “Fix Geo-Origin” instructs TeleSculptor to keep the current geographic origin and not recalculate a new one.

## Advanced Configuration Options

Most users should not need these advanced features, but some may be interested.

### Changing Frame Sampling Rate

By default, all video frames are loaded when opening a video (though not all are processed in feature tracking). If the number of frames is too large to manage it is possible to read only every Nth frame by changing a setting in the project configuration. First close the application. Next open the project .conf file in a text editor like Notepad. Look for the following line:

**video\_reader:filter:output\_nth\_frame= 1**

Increase the number from 1 to 10 to sample every 10<sup>th</sup> frame, for example. Save the .conf file in the text editor and open that file again in the TeleSculptor application.

### Modifying Algorithm Parameters

TeleSculptor is a highly configurable application, though most of the configuration options are not yet exposed to the user interface. Each tool in the compute menu calls an algorithm from the KWIVER toolkit and each algorithm is configurable at run time. Algorithms can even be swapped out for other algorithms at run time. One can contribute a new algorithm without recompiling TeleSculptor by dropping in a new DLL and updating the configuration files. All this configurability is managed with configuration files. The project file is one example of configuration file, but there are also many default configuration files loaded by TeleSculptor at run time. The default configuration files for a standard install path are found in these two locations:

**C:\Program Files\TeleSculptor 1.1.0\share\telesculptor\1.1.0\config**

**C:\Program Files\TeleSculptor 1.1.0\share\kwiver\1.5.0\config**

TeleSculptor specific configurations are found in the first directory and these include configurations for KWIVER algorithms found in the second directory. It is recommended that you not modify these values, but instead copy some of these files into your project directory and modify the copies. TeleSculptor will load configuration files from the project directory first. Each of the tools in the *Compute* menu loads a configuration file when it is run. These have names starting with a “gui\_” prefix. For example:

Compute Menu Name	Configuration File Entry Point
Track Features	gui_track_features.conf
Estimate Camera/Landmarks	gui_initialize.conf
Save Frames	gui_frame_image_writer.conf
Batch Compute Depth Maps	gui_compute_depth.conf
Fuse Depth Maps	gui_integrate_depth_maps.conf

Most of these configuration files also reference other configuration files with an “include” statement. Configuration values for tools can also be added to the project file but copying the GUI configuration files into the project directory adds more flexibility because these files are reloaded each time the tool is run, which allows changing parameters between runs without loading the project.

As an example, consider changing the maximum number of frames to use in the feature tracker. First copy `gui_track_features.conf` into your project directory. Open this file and look for the following section:

```
# Parameters for the feature tracker
block feature_tracker
  include core_feature_tracker.conf

  # The maximum number of frames for the GUI to process.
  # The tracker will choose frames distributed over the video
  max_frames = 500
endblock
```

modify the line with “`max_frame = 500`” use a different value, such as 1000. Note that you could also make this change in the project file by appending the following line:

```
feature_tracker:max_frames = 1000
```

Note that the `max_frames` parameter is in the `feature_tracker` scope and scope must be specified either using `block/endblock` notation or with a prefix before a colon.

### Printing all KLV metadata

The TeleSculptor application loads KLV metadata and display it in a viewer, but there is no way to export this data in batch. However, the installer does provide the `kwiver` command line tool that has an applet that will print out all metadata in a video. This applet is called “`dump_klv`”. The default installation path is

**C:\Program Files\TeleSculptor 1.1.0\bin\kwiver.exe**

To run `dump_klv`, open up a command prompt (search for `cmd.exe` in the Start Menu). Then run

**“C:\Program Files\TeleSculptor 1.1.0\bin\kwiver.exe” `dump_klv video_file.mpeg`**

and replace “`video_file.mpeg`” with the path to the video file to process. This will print out all the metadata. To redirect the output to a file use:

**“C:\Program Files\TeleSculptor 1.1.0\bin\kwiver.exe” `dump_klv.exe video_file.mpeg > metadata.txt`**