DYNOSCAN3D USER MANUAL

1 Software

1.1 GitHub File Descriptions

Name	Location	Description
DSLR_Capture.py	DynoScan3D/DSLRImageCapture	Automated image projection and capture via a DSLR camera
DSLR_Capture_Integrated.py	DynoScan3D/DSLRImageCapture	Automated image projection and capture via a DSLR camera, integrated with the turntable
ArduinoControls.ino	DynoScan3D/Arduino Script	Arduino script for turntable controller
PointCloudDistance.py	DynoScan3D/PointCloudDistances	Script to get distance distributions for point clouds
SaveGrayCode.m	DynoScan3D/GrayCode	Generate and save gray code images
AndroidImageCapture (Folder)	DynoScan3D	Legacy image projection and capture scripts using an android phone
Drawings (Folder)	DynoScan3D	Engineering drawings for all manufactured components
Resources (Folder)	DynoScan3D	Collated list of references and software used

1.2 List of Resources

Name	Source/Link	Description
Python	Various IDEs Online	Requires at least a terminal console to input commands
Scan3D	GitHub Resources Folder	Structured Light calibration and reconstruction
CloudCompare	Website	Point cloud clean up and alignment
MeshLab	Website	Meshing Software

2 Operation

2.1 Software Setup

1. Generation of Gray Code patterns

Run the SaveGrayCode.m file along with the other function files in that same folder. After specifying the projector's resolution, the set of gray code images will be generated in a subfolder.

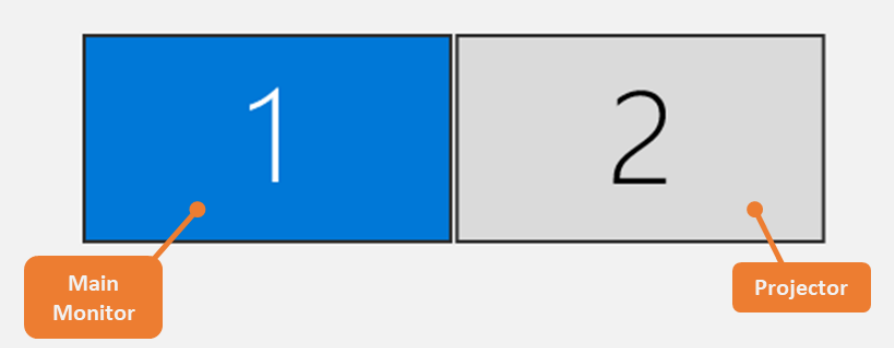
2. Image acquisition

Move the subfolder into the same folder as DSLR_Capture_Integrated.py and DSLR_Capture.py is in. Ensure that the computer has enough memory to store all images taken. An estimate is about 50-80 mb per scan.

2.2 Hardware Setup

1. Wiring

Connect the projector to the computer via a HDMI attachment and camera via a USB. Connect the Arduino Uno to the computer via the serial communication cable. Plug in the AC-DC IEC connector to the mains.



The projector should be connected to the computer as a second screen, with this screen to the right of the main monitor. This is as OpenCV does not allow one to specify which screen to show images on, displaying it on the main monitor by default. This is overcome by offsetting the image to the right such that it appears on the projector instead.

2. Scanner placement

Scanner should be placed such that the object takes up the majority of the projected screen's space. Furthermore, the camera should be positioned such that most of the captured image is taken up by the projected patterns but must be wholly contained in the image. Once the scanner is in place, adjust the knob on the projector to ensure the focal point is on most of the object. If the object has a large variation in depth, it is advisable to do multiple scans of that object at different distances from the scanner such that sufficient data is obtained for alignment.

3. Calibration board

A white and black checker-box pattern should be printed against some rigid backing material. The pattern should be about the size of the projected screen and have a different number of boxes in the vertical and horizontal direction. Also, the width of each square should be known accurately. Previous experiments have used a board of 7x9 internal corners with square width of 18 mm.

2.3 Operation

1. Calibration

Run the DSLR_Capture.py script, making sure that the calibration board is stably propped up on the turntable. After one scan has been done, move the calibration board to another position, still facing the projector, and hold it in position. A scan is completed when an input prompt is observed in the terminal. Follow the input and enter 'c' to continue another scan.

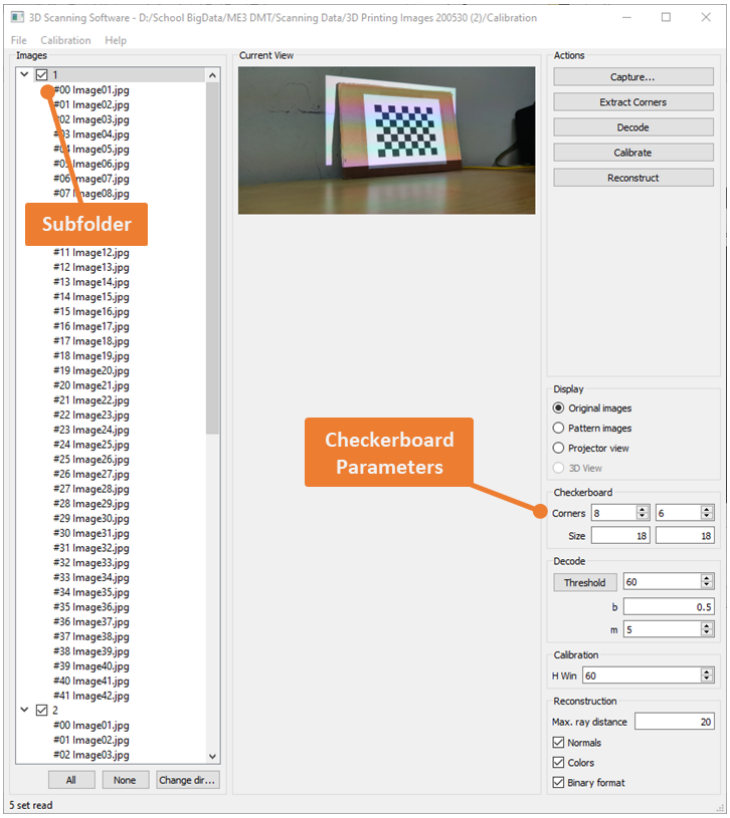
At least 5 scans are required for proper calibration, with the board placed in varying angles each time, but not such that the checker box patterns cannot be detected on the camera.

2. Image Acquisition

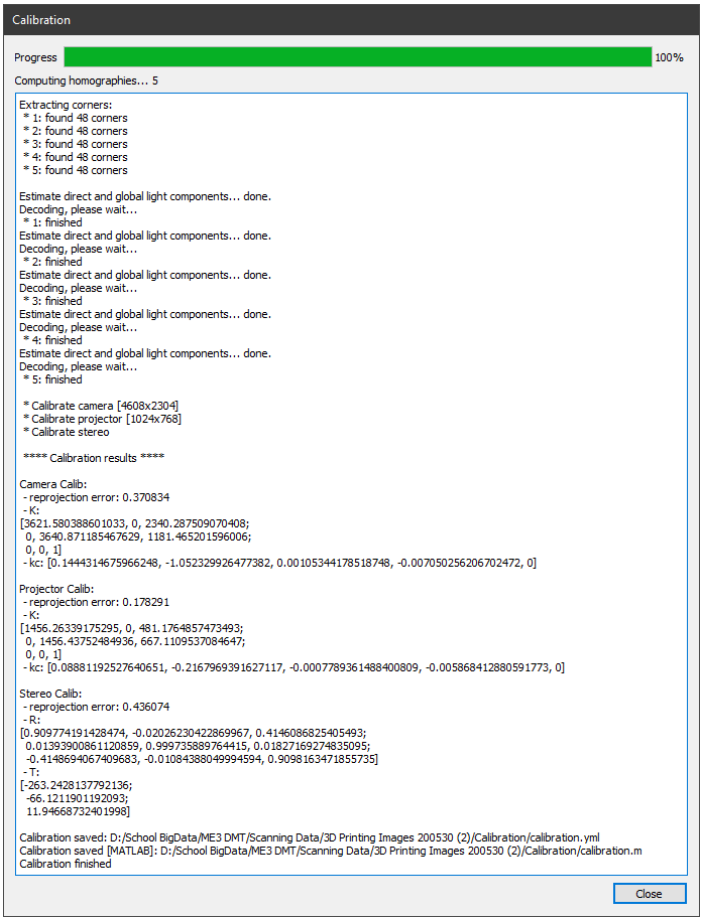
Following this, place the object on the turntable and run the DSLR_Capture_Integrated.py script. Enter the desired angular increment per scan and the initial scan will run. After each scan, follow the prompt to either continue scanning, scan a new object, or terminate the program.

3 Data Processing

3.1 Calibration and Reconstruction



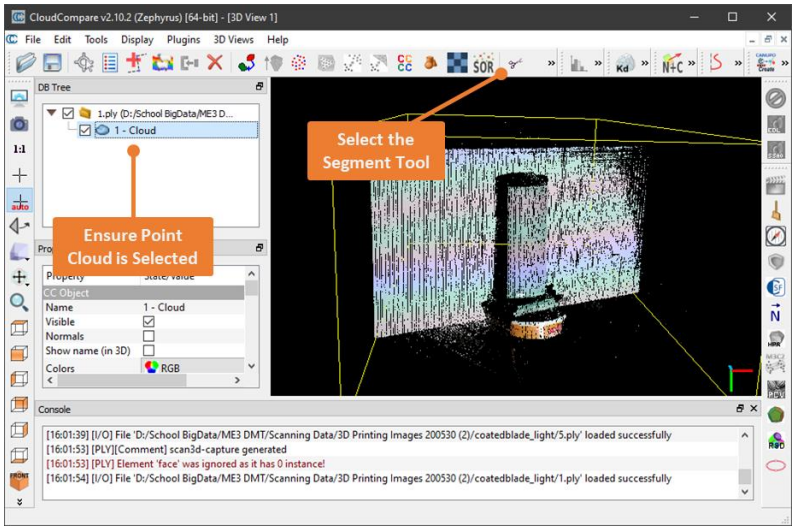
Open scan3d-capture-x64.exe, found in the resources folder. Using File > Change Dir, navigate to the folder containing subfolders of the captured checker box images. After that, enter the number of internal corners and the square size of each box in mm. Ensuring that all the subfolders containing the different calibration positions are selected (checked), select the 'Calibrate' action.



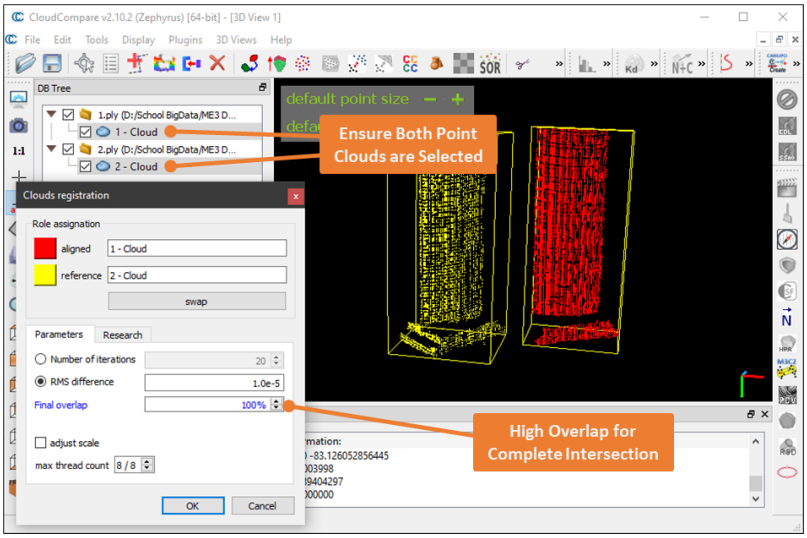
If corners are not found, the internal corner numbers are either wrongly specified, or the image is too dark for the checker boxes to be recognized. These calibration parameters are then saved in a .yml and a .m file in the same parent folder, and the .yml file can be loaded from the Calibration > Load tab to use these values.

After calibration, change the directory to the folder containing the subfolders of the captured object images. To reconstruct a point cloud, select a subfolder (the checkmarks are only for calibration), and select the ‘Reconstruct’ action. If successful, a prompt to save the .ply file should appear.

3.2 Cleaning Up and Alignment

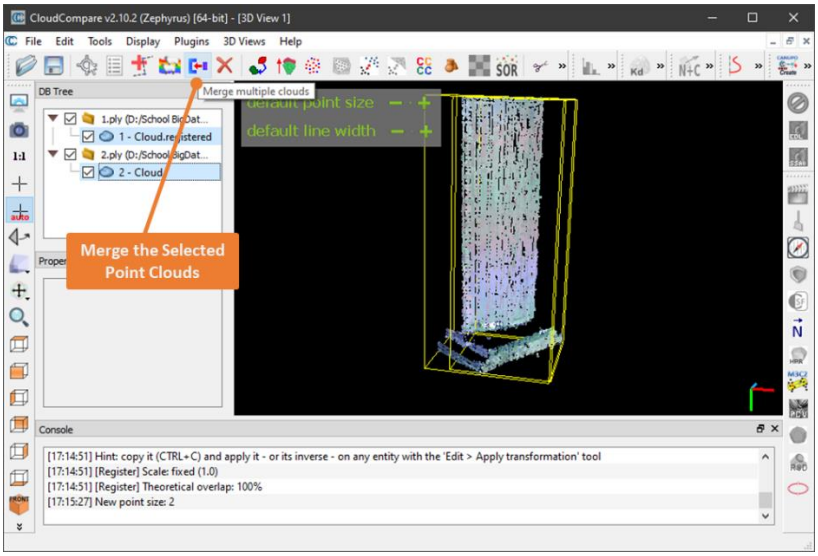


Once the point cloud is loaded, the unwanted parts can be cut out using the segment tool. A detailed operational guide can be found on the software's documentation. Further removal of noisy points can be done via the SOR filter, found in the Tools > Clean tab.



Alignment is then done using the Fine Registration tool, found in the Tools > Registration. While a detailed explanation of the various parameters can be found on the documentation, it is important to note that the adjust scale box should not be ticked, and the final overlap should only be 100% if the two models intersect almost completely.

Sometimes, incorrect alignment occurs if the final overlap is overestimated. It was found that segmenting portions of both point clouds that completely intersect instead is found to be more reliable and accurate. The detailed procedure is found at the bottom of the Fine Registration documentation linked above, with greater success given with sharp corners and flat surfaces.



Once the different scans have been aligned, they can be merged by using the merge button indicated in the above image. A complete point cloud is then exported via selecting that point cloud and going to File > Save.

3.3 Meshing

Two options in Meshlab have been used with some success in creating a smooth watertight mesh. The first is the Screened Poisson Surface Reconstruction, found in the Filters > Remeshing, Simplification, and Reconstruction tab. This can be done directly with an imported point cloud.

Another option is to first down-sample the point cloud using Filters > Point Set > Point Cloud Simplification, from which a variety of different surface reconstruction methods can be used, such as Marching Cubes or Ball Pivoting, all found in the Filters > Remeshing, Simplification, and Reconstruction tab.