

MANUAL v0.30

January 4, 2024

Contents

1	Difference to previous version	1
2	Introduction	3
3	Lidar odometry (step 1)	5
4	Multi view terrestrial laser scan registration (steps 2 and 3)	11
4.1	Step 2	11
4.2	Step 3	15
5	Georeferencing	23
5.1	Georeferencing with point cloud	23
5.2	Georeferencing with WGS84toCartesian	25
6	Questions from end users	29

Chapter 1

Difference to previous version

Added section 5.2: Georeferencing with WGS84toCartesian.

Chapter 2

Introduction

This manual v0.25 is prepared for mobile mapping system https://github.com/JanuszBedkowski/mandeye_controller/blob/main/doc/manual/manual_v0_2/mandeye_dev_manual_v0_2.pdf MANDEYE available as open hardware project. The software is composed of:

- Lidar odometry (for initial trajectory calculation),
- Multi view terrestrial laser scan registration (for final trajectory calculation).

To use the software click the link below:

<https://github.com/MapsHD/HDMapping/releases>
and download the latest version of files: laszip3.dll, lidar_odometry_step_1.exe, multi_session_registration_step_3.exe and multi_view_tls_registration_step_2.exe.
Then put all of the downloaded files in one folder and proceed to next chapter.

Chapter 3

Lidar odometry (step 1)

This software calculates trajectory based on Lidar and IMU data. It based on novel approach that I did not have opportunity to publish (work in progress). Basically it is SAM (Smoothing and Mapping) approach that is using multi view Normal Distributions Transform in pose graph SLAM framework written from scratch in Python (SymPy) and C++ (Eigen).

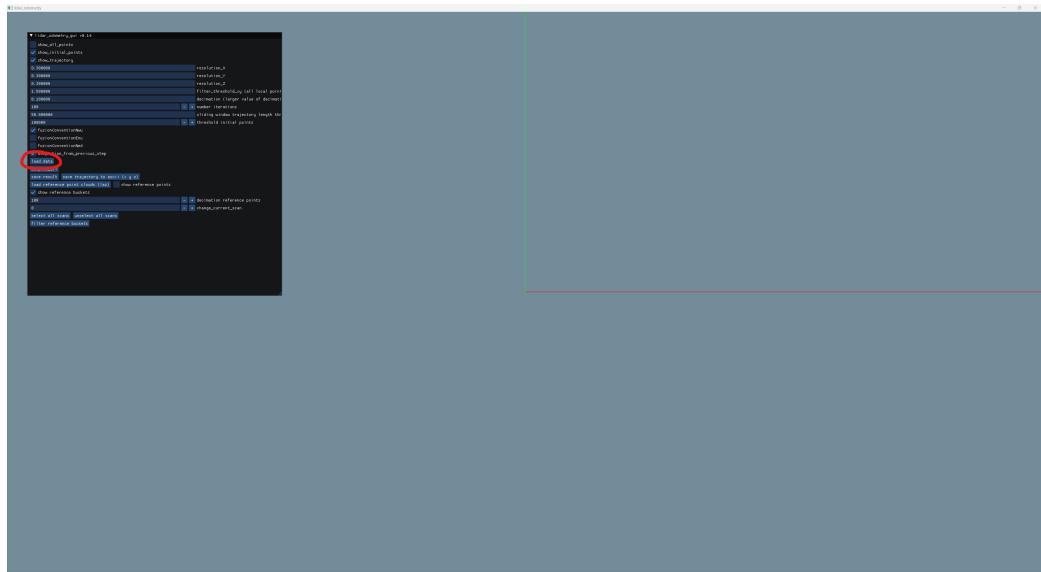


Figure 3.1: Step 1 - loading data.

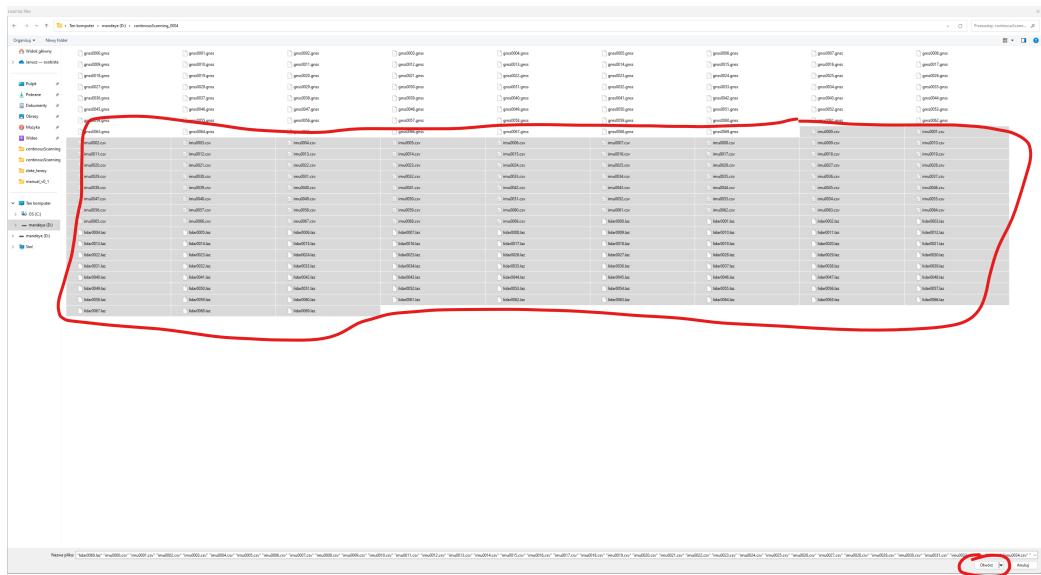


Figure 3.2: Step 2 - select all *.csv and *.laz files from folder that MANDEYE mobile mapping system created on USB drive.

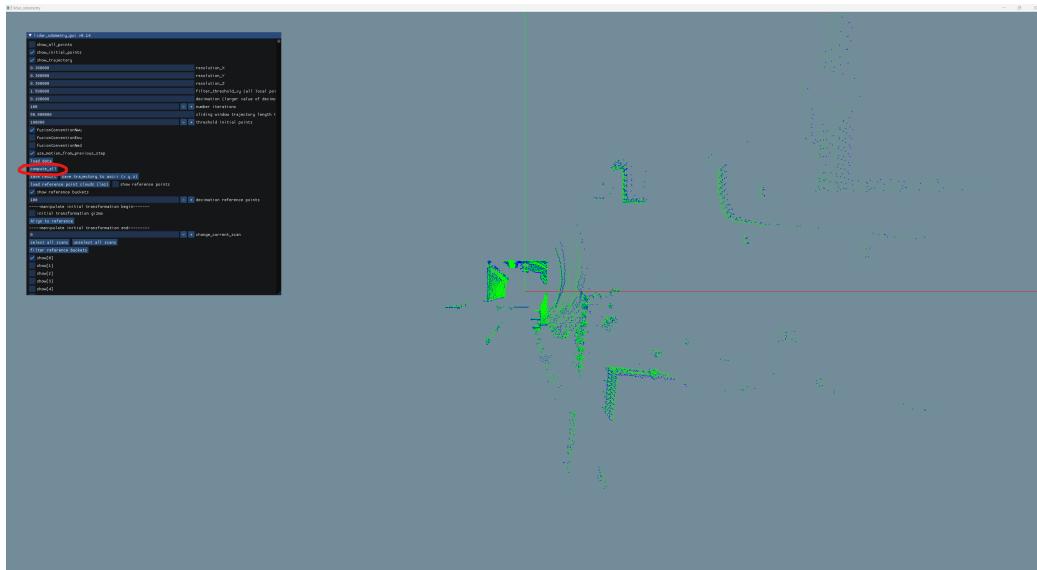


Figure 3.3: Step 3 - press 'compute all'. Check console mean time and folder 'preview'.

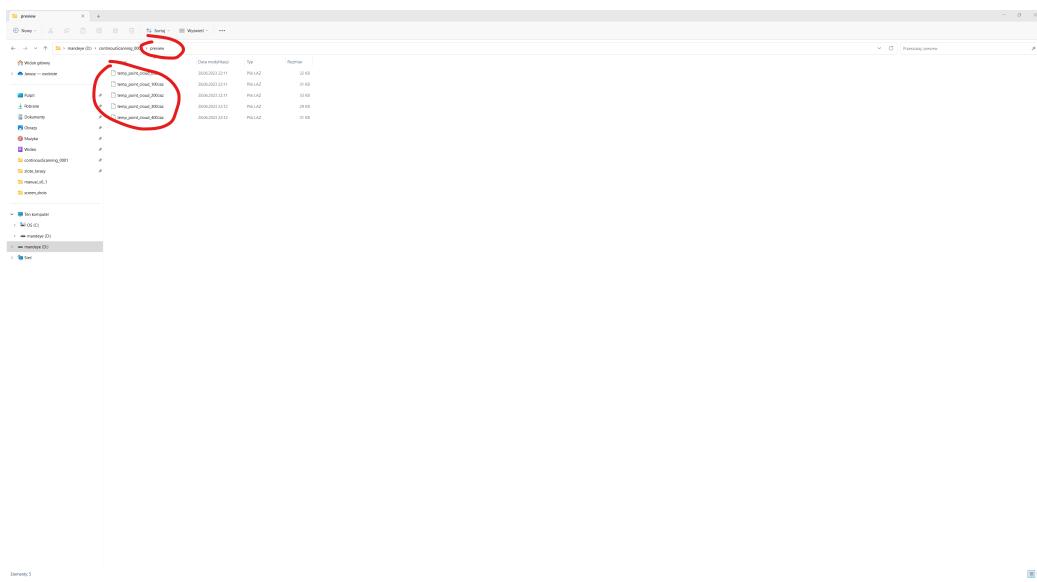


Figure 3.4: Optional step: intermediate results are stored in 'preview' folder.

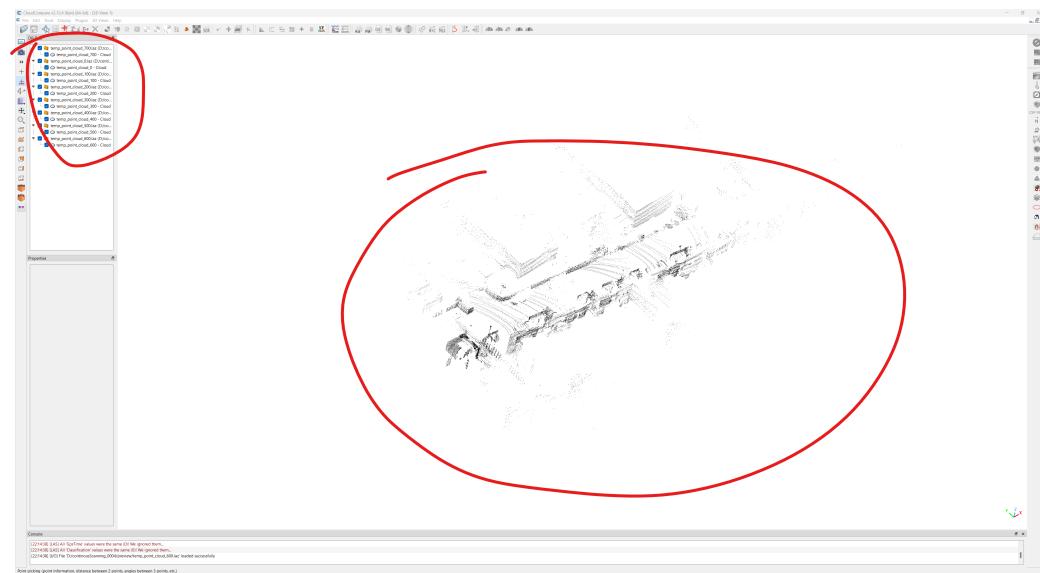


Figure 3.5: Optional step: You can watch the progress in open source Cloud-Compare software by loading all *.laz files from 'preview' folder.

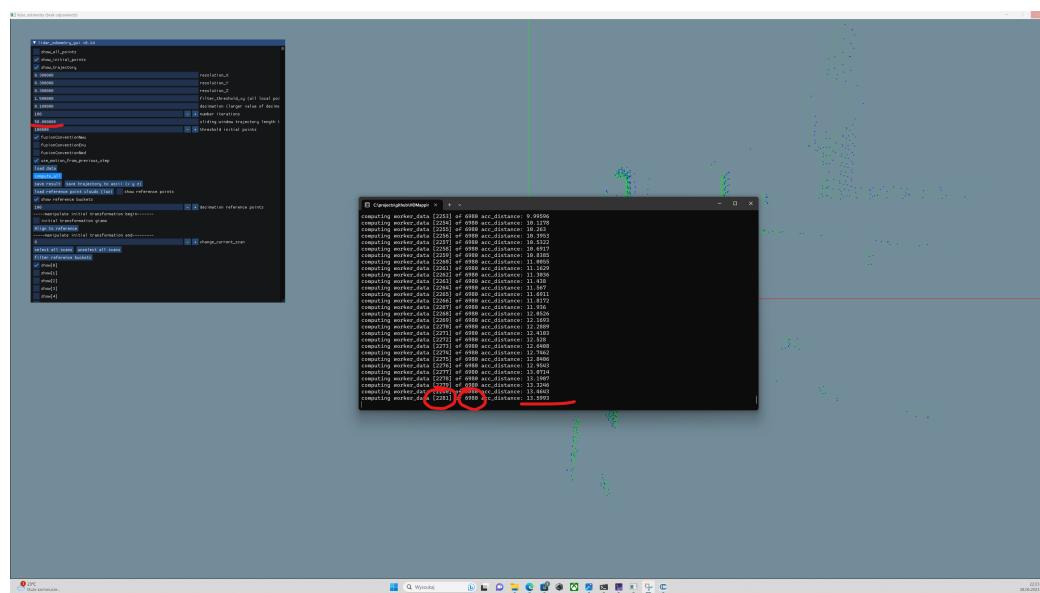


Figure 3.6: Progress in console.

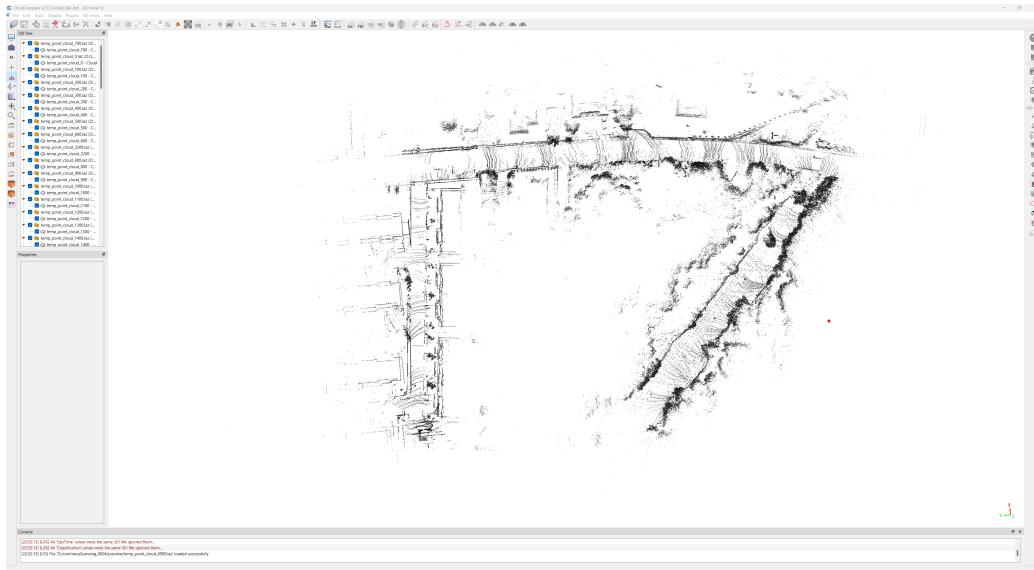


Figure 3.7: Final data in CloudCompare.

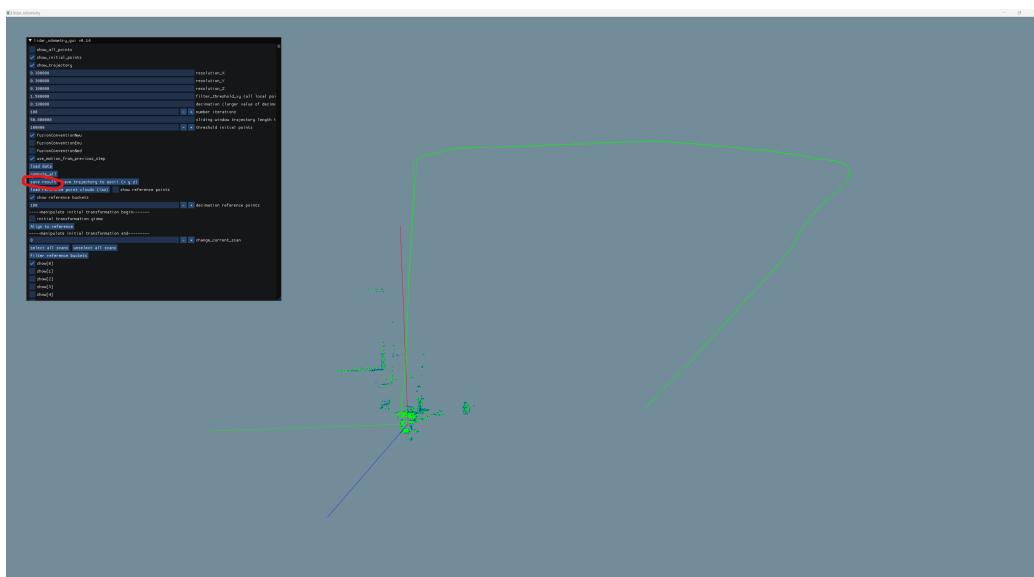


Figure 3.8: Final data ready for export.

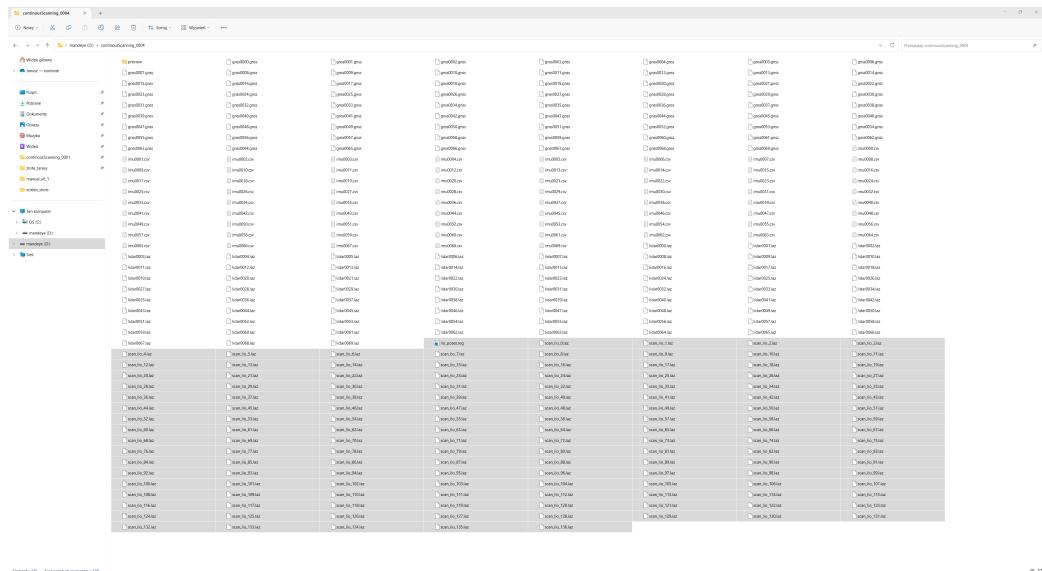


Figure 3.9: Exported final files.

Chapter 4

Multi view terrestrial laser scan registration (steps 2 and 3)

4.1 Step 2

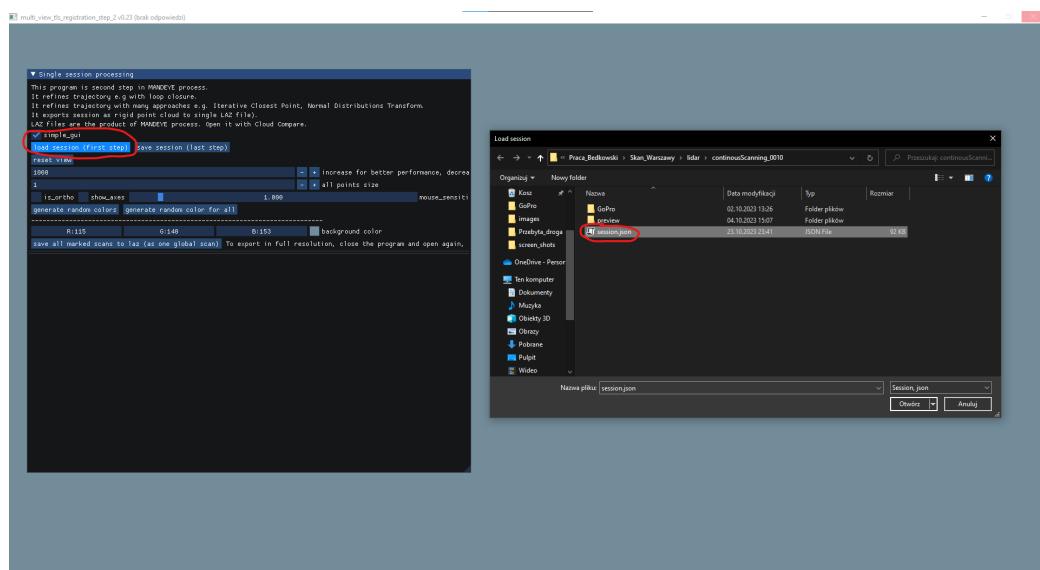


Figure 4.1: Load session.json prepared by 'Lidar odometry'.

12CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

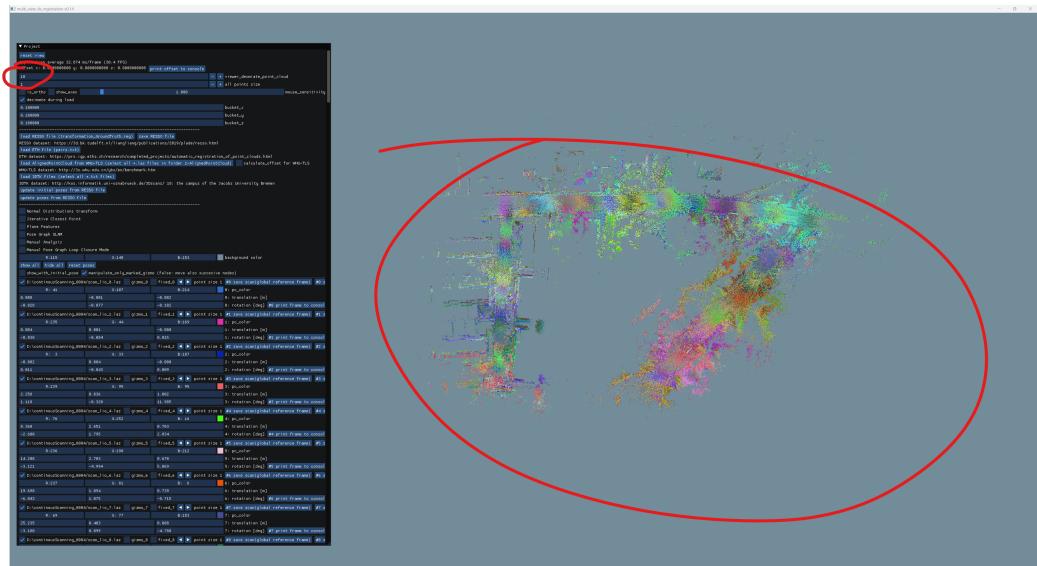


Figure 4.2: Prepare field of view and change decimation to see more points. Generate random colors option is recommended for next steps as every scan will be in a different color.

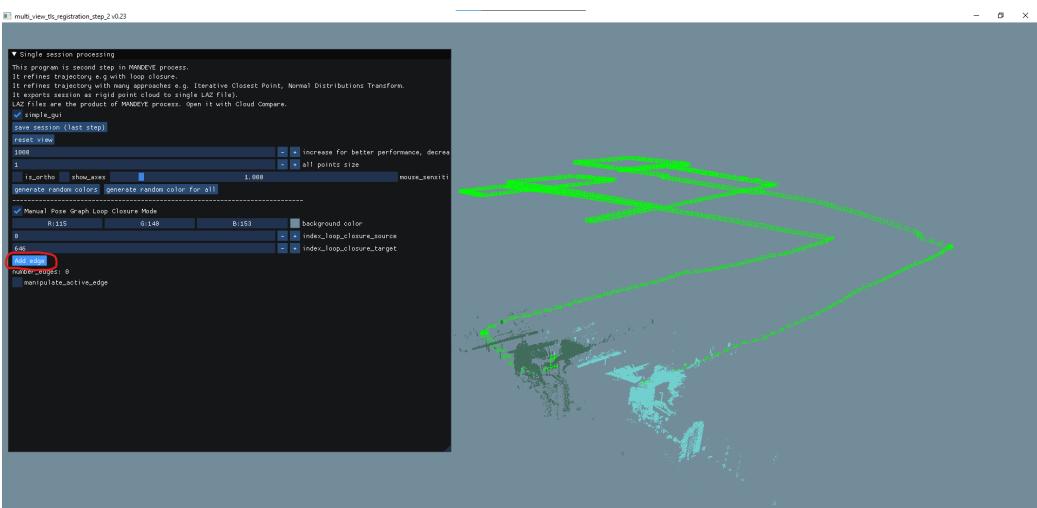


Figure 4.3: Turn on Manual Pose Graph Loop Closure Mod, then choose two different scans that share scanned objects, but difference in their numbers is as big as possible e.g. when you made a loop during scanning and came back to the same place after some time. Then click add edge.

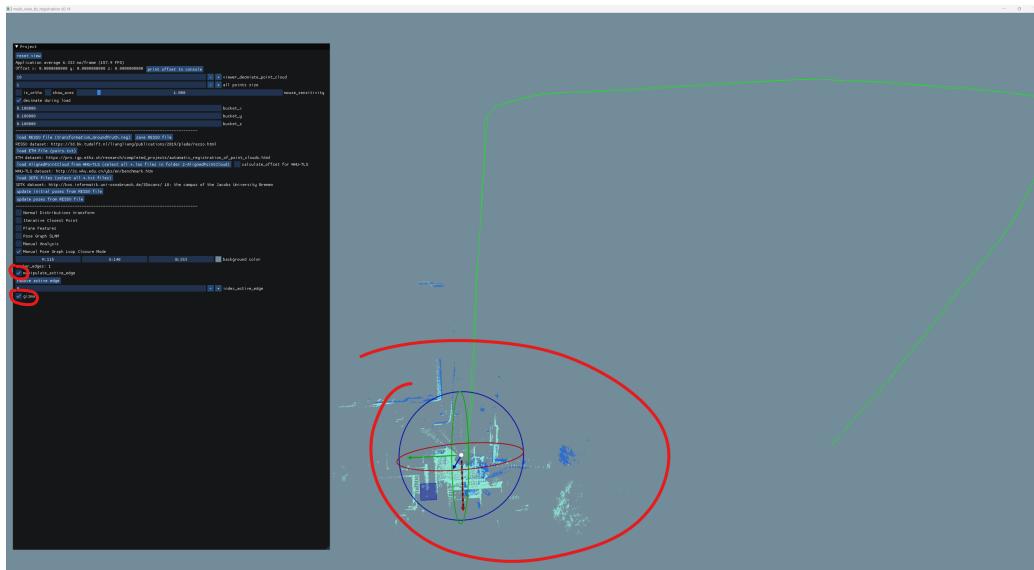


Figure 4.4: Turn on manipulate active edge, turn on gizmo and align scan to scan manually.

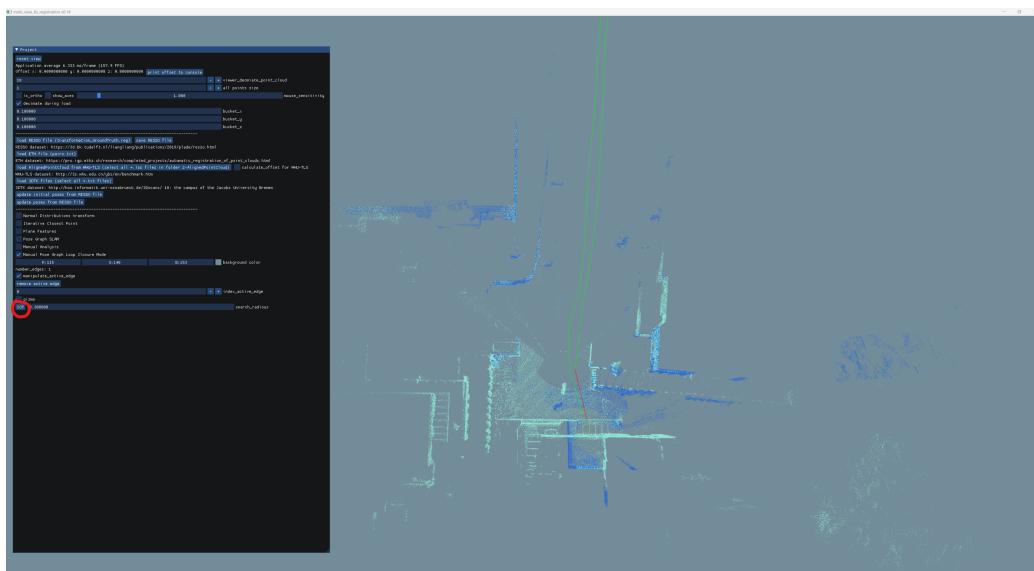


Figure 4.5: Once You are not capable align more accurate, then turn off gizmo and repetitively use ICP until scans align to the level at which nothing can change anymore.

14CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

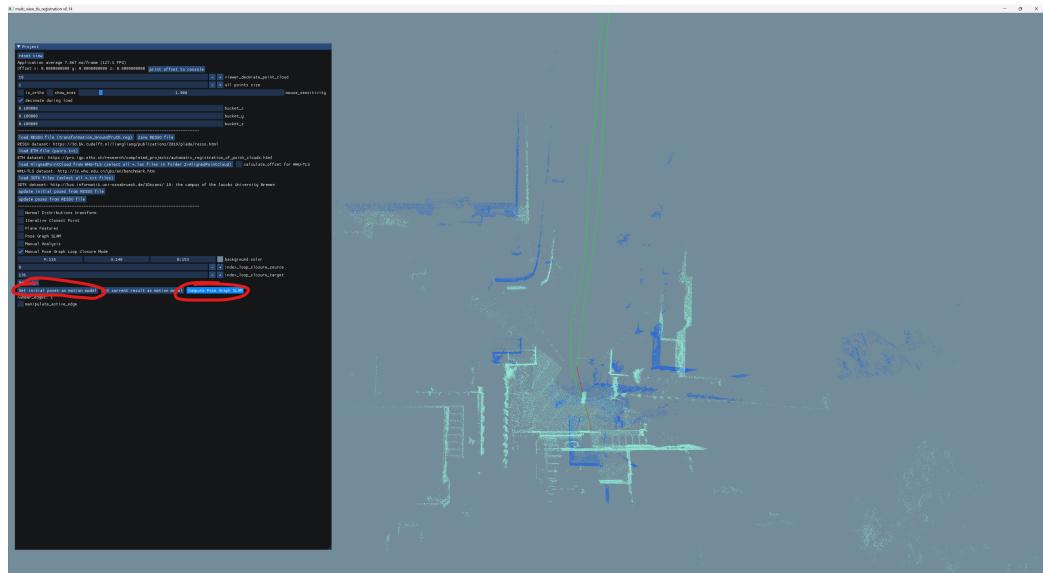


Figure 4.6: Turn off manipulate active edge, click "set initial poses as motion model", then click "compute pose graph SLAM".



Figure 4.7: Turn off Manual Pose Graph Loop Closure Mod and inspect if everything is ok, if not, repeat steps from figures 3.3-3.6 (choose another pair of scans, refine them and compute the pose graph SLAM).

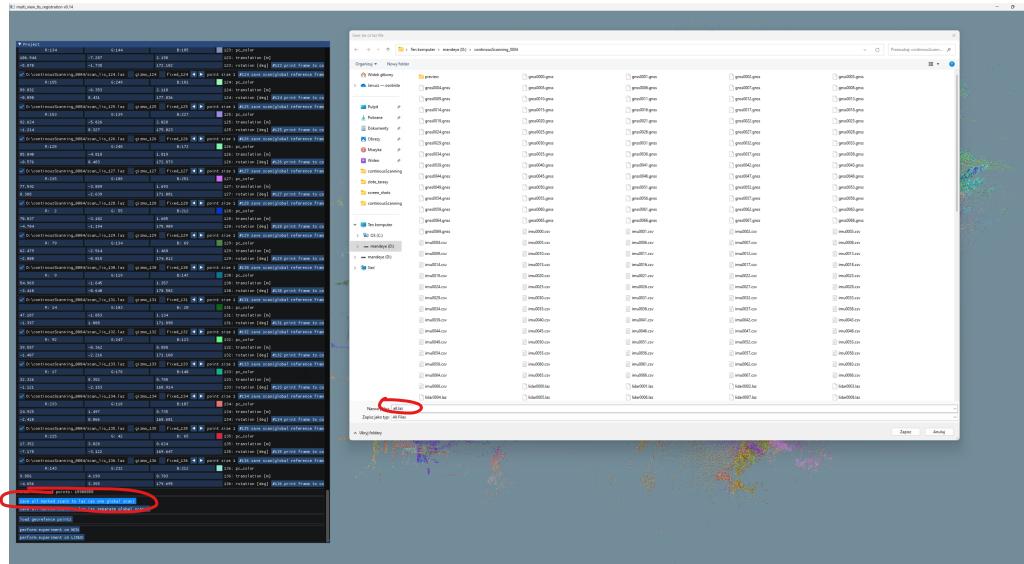


Figure 4.8: Once the job is done click Save session button to save changes to session.json and export data to *.laz. The latter is Your map that can be loaded by e.g. CloudCompare.

4.2 Step 3

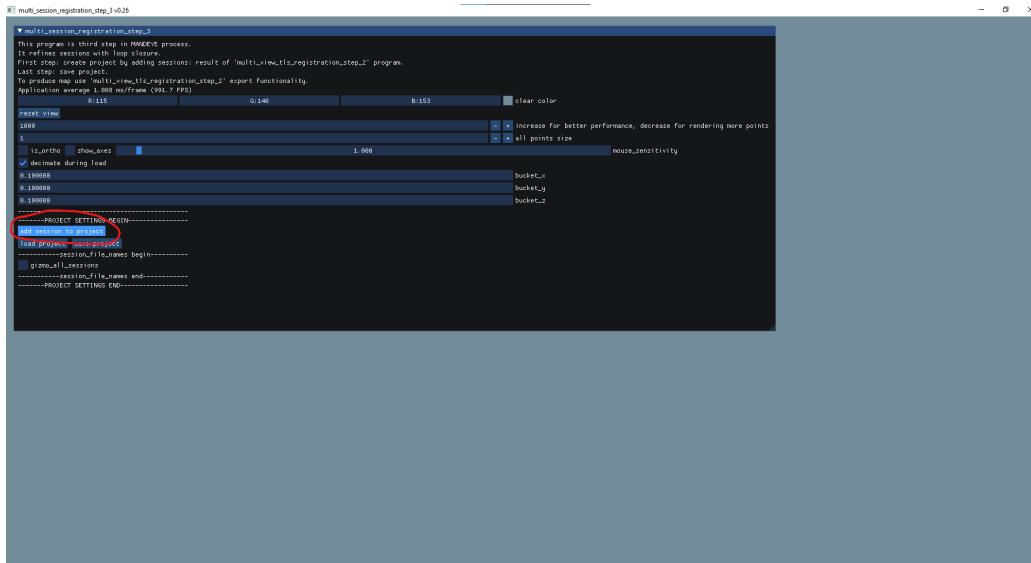


Figure 4.9: Add sessions that you want to align.

16 CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

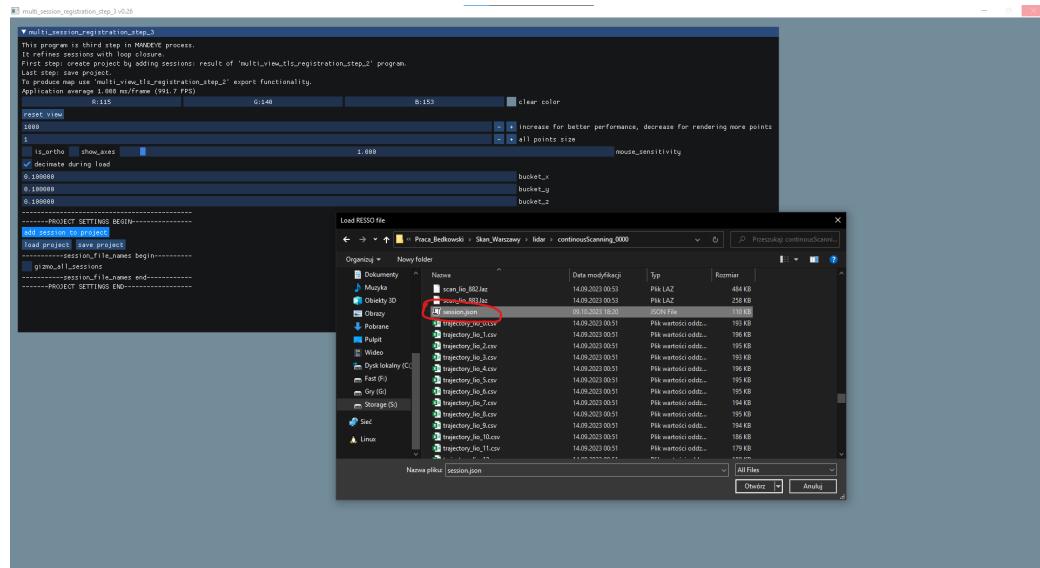


Figure 4.10: Choose session.json files - effects of the lidar odometry step.

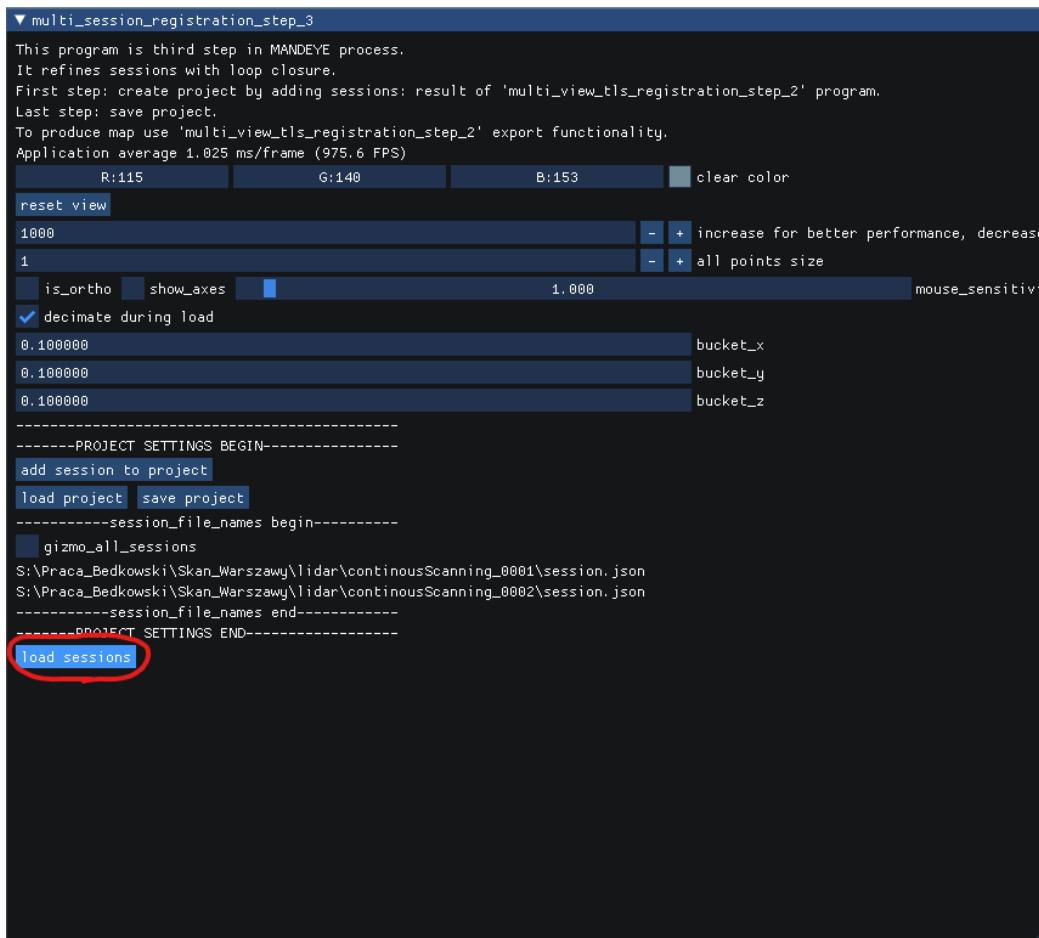


Figure 4.11: Click load sessions button and wait for the chosen sessions to load.

18CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

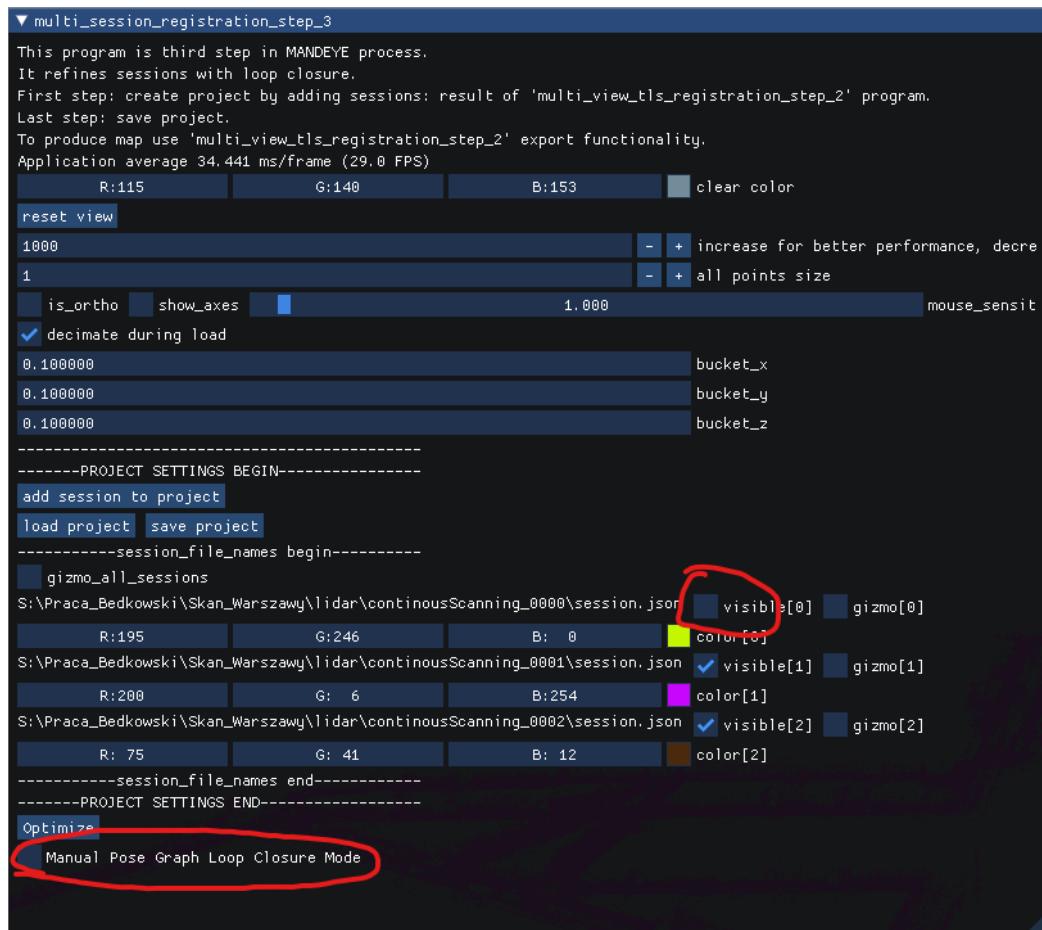


Figure 4.12: When all of the sessions have loaded activate Manual Pose Graph Loop Closure Mode. If more than 2 sessions were loaded, deactivate sessions till two of them remain. After that the button should appear.

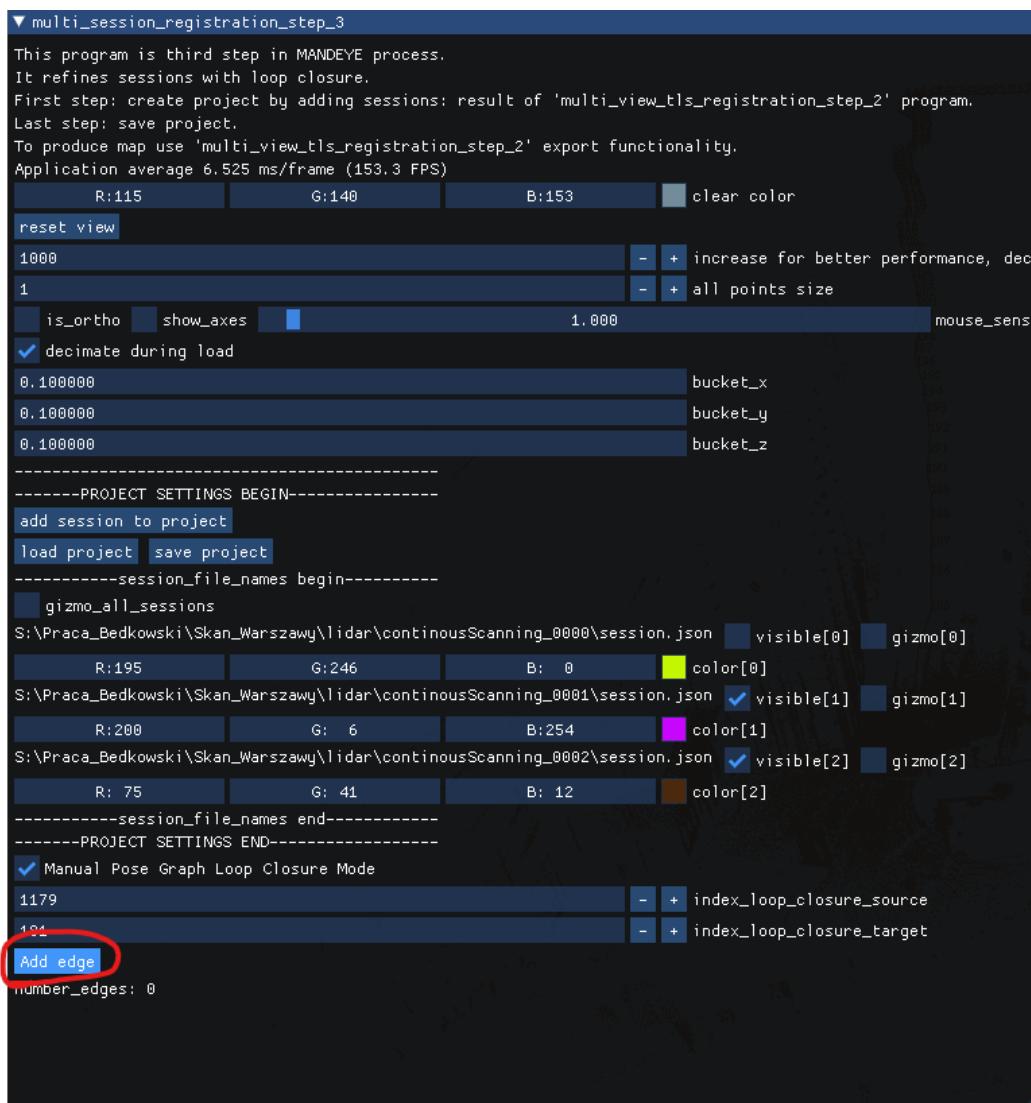


Figure 4.13: Choose 2 individual scans of the same area, one from the first session, other from the second session and click add edge.

20CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

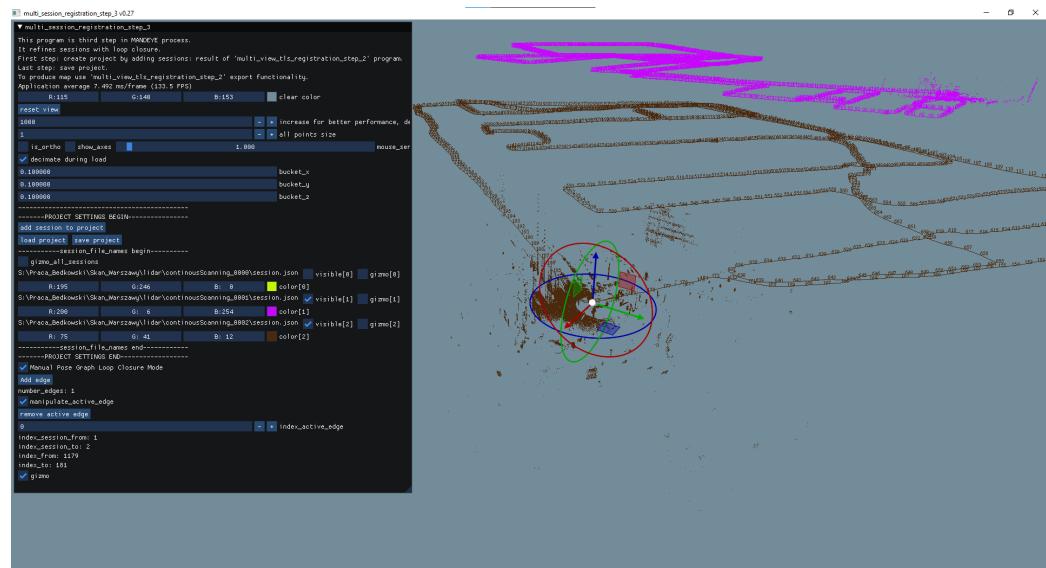


Figure 4.14: Click manipulate active edge, then gizmo and as in the step 2 align scans as precisely as possible and then repeatedly use ICP till nothing changes.

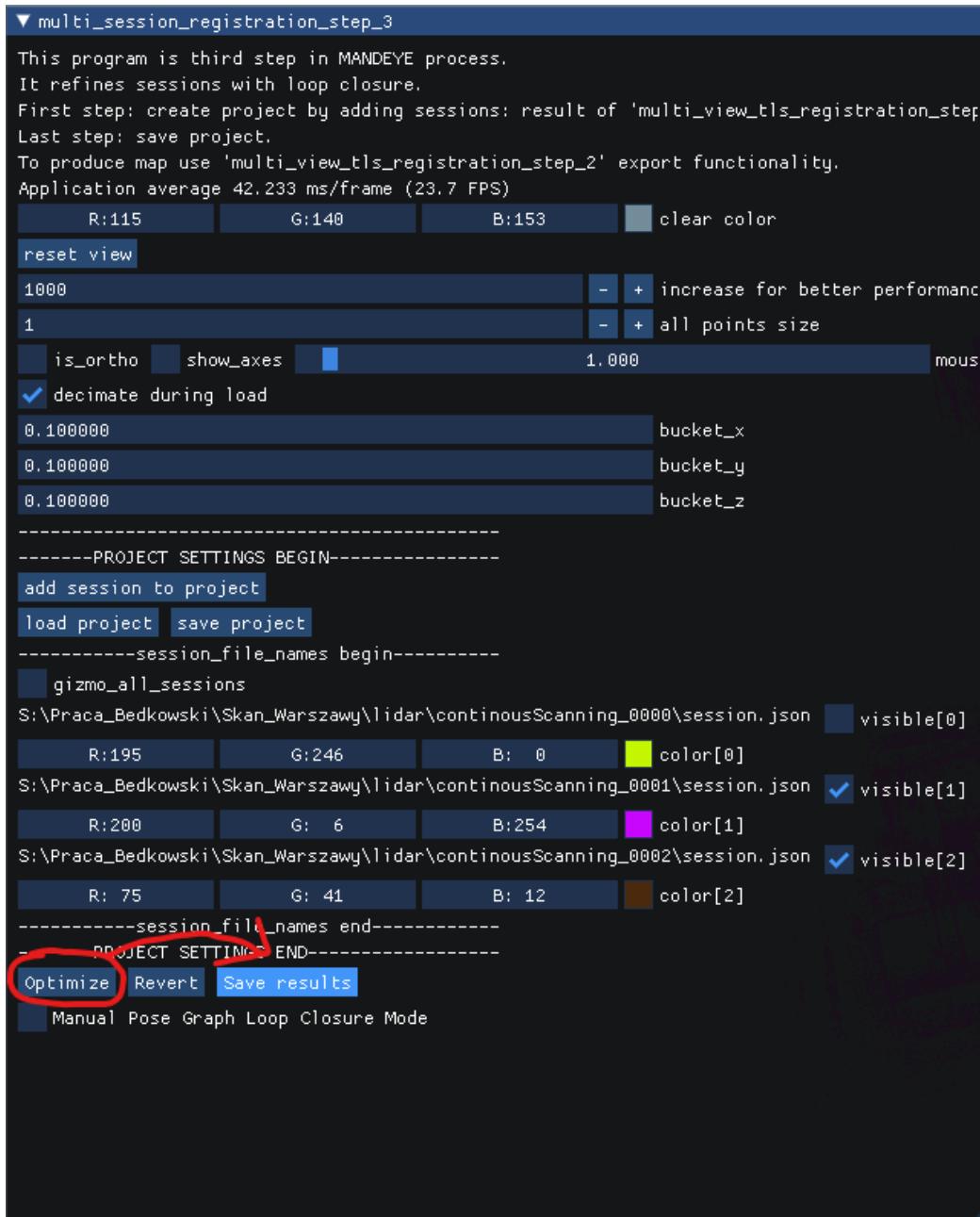


Figure 4.15: After aligning scans turn off Manual Pose Graph Loop Closure Mode, click Optimize and if everything is ok then click Save results. Should anything go wrong and sessions haven't orientated as planned just use Revert button. Repeat steps 3.13-3.15 until two sessions are aligned with a satisfying effect.

22CHAPTER 4. MULTI VIEW TERRESTRIAL LASER SCAN REGISTRATION (STEPS)

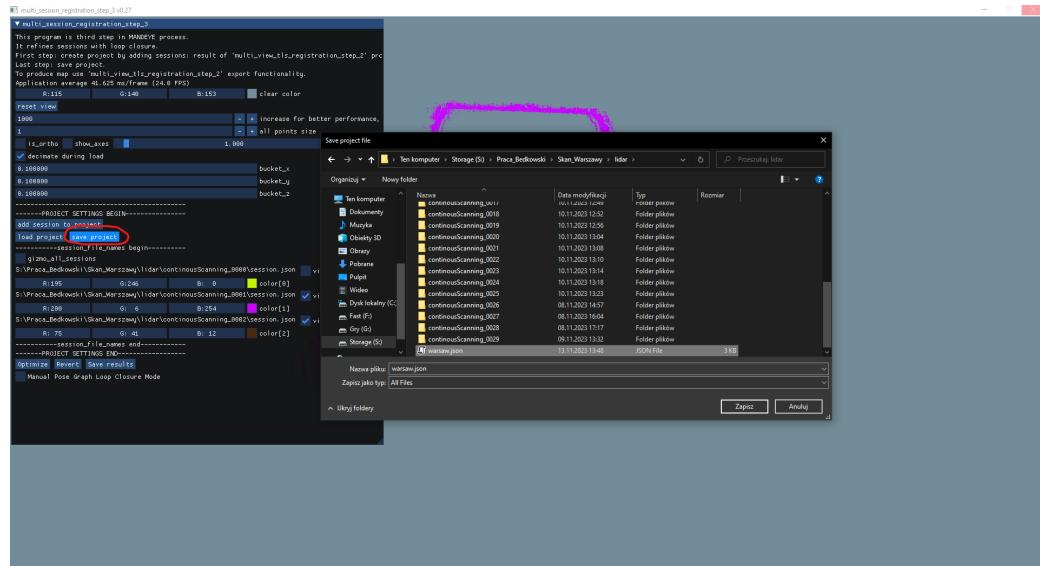


Figure 4.16: At the end or in the middle of work you can save your project to .json file, which can be loaded next time multi session registration step 3 is used.

Chapter 5

Georeferencing

5.1 Georeferencing with point cloud

It is possible to set session as ground truth. Thus, optimization process (Pose GRAPH SLAM) will not change its poses. Other sessions can be aligned against ground truth session by adding edges.

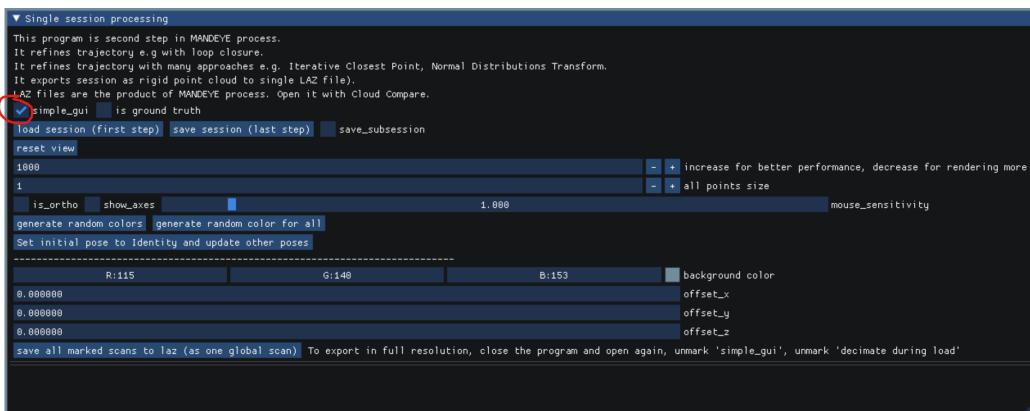


Figure 5.1: Use multi view tls registration step2 program to open TLS files.

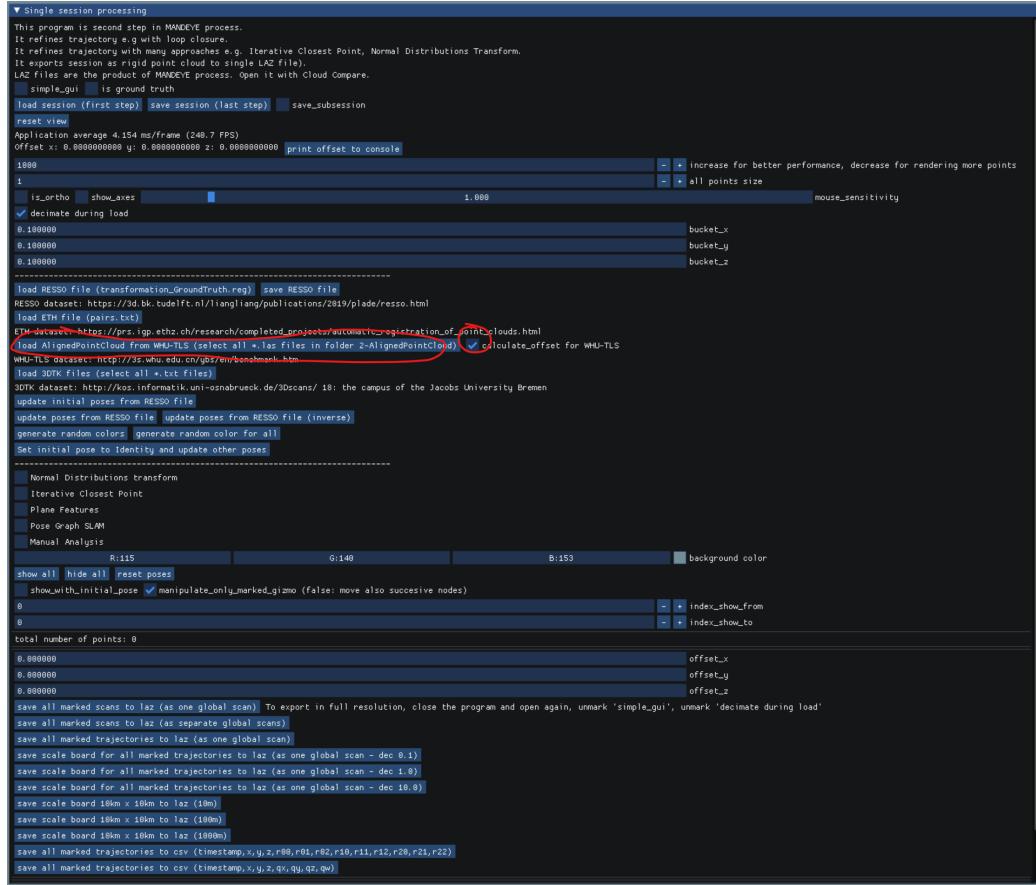


Figure 5.2: Mark calculate offset for WHU-TLS, load AlignedPointCloud from WHU-TLS (select all *.las/laz files in folder)

```

▼ Single session processing
This program is second step in MANDEYE process.
It refines trajectory e.g with loop closure.
It refines trajectory with many approaches e.g. Iterative Closest Point, Normal Distributions Transform.
It exports session to rigid point cloud to single LAZ file).
LAZ files are the product of MANDEYE process. Open it with Cloud Compare.
simple_gui is ground truth
Load session (first step) Save session (last step) save_subsession
Reset view
Application average 24.729 ms/frame (40.4 FPS)
Offset x: 635985.8783896384 y: 488138.7183231985 z: 119.8816449897 print offset to console
10 increase for better performance, decrease for rendering more point
1 all points size
is_ortho show_axes 1.000 mouse_sensitivity
✓ decimate during load
0.180000 bucket_x
0.180000 bucket_y
0.180000 bucket_z

Load RESSO file (transformation_GroundTruth.reg) save RESSO file
RESSO dataset: https://3d.bk.tudelft.nl/liangliang/publications/2019/plade/rezzo.html
Load ETH File (pairs.txt)
ETH dataset: https://pris.ip.ethz.ch/research/completed_projects/automatic_registration_of_point_clouds.html
Load AlignedPointCloud from WHU-TLS (select all + .laz files in folder 2-AlignedPointCloud) calculate_offset for WHU-TLS
WHU-TLS dataset: http://3s.whu.edu.cn/gbs/en/benchmark.htm
Load 3DTK files (select all + .txt files)
3DTK dataset: http://www.informatik.uni-stuttgart.de/3Dscans/ 18: the campus of the Jacobs University Bremen
Update initial poses from RESSO file
update_poses_from_RESSO_file update_poses_from_RESSO_file (inverse)
generate random colors generate_random_color_for_all
Set initial pose to Identity and update other poses

Normal Distributions transform
Iterative Closest Point
Plane Features
Pose Graph SLAM
Manual Analysis
Manual Pose Graph Loop Closure Mode
R:115 G:140 B:153 background color
show all hide all reset poses
show_with_initial_pose manipulate_only_marked_gizmo (false: move also successive nodes)
0 index_show_From
0 index_show_to
D:\ISOK\78787_845208_N-34-138-B-d-2-2-2-3.laz gizmo_0 fixed_0 point size 1 #0 save scan(global reference frame) #0 shift points to center
R: 41 G:107 B:24.4 0: pc_color #0_1CP #0 print frame to console #0 choose_geo
D:\ISOK\78787_845209_N-34-138-B-d-2-2-2-4.laz gizmo_1 fixed_1 point size 1 #1 save scan(global reference frame) #1 shift points to center
R:235 G: 44 B:169 1: pc_color #1_1CP #1 print frame to console #1 choose_geo
total number of points: 13531621
0.000000 offset_x
0.000000 offset_y
0.000000 offset_z
save all marked scans to laz (as one global scan) To export in full resolution, close the program and open again, unmark 'simple_gui', unmark 'decimate during load'
save all marked scans to laz (as separate global scans)
save all marked trajectories to laz (as one global scan)
save scale board for all marked trajectories to laz (as one global scan - dec 0.1)
save scale board for all marked trajectories to laz (as one global scan - dec 1.0)
save scale board 10km x 10km to laz (10m)
save scale board 10km x 10km to laz (100m)
save scale board 10km x 10km to laz (1000m)

```

Figure 5.3: 1: save RESSO file, 2: update initial poses from RESSO file (select file from 1), 3: update poses from RESSO file (select file from 1), 4: set checkbox is ground truth, 5: save session.

5.2 Georeferencing with WGS84toCartesian

It uses <https://github.com/chrberger/WGS84toCartesian/tree/master> WGS84toCartesian. It is a small and efficient library written in modern C++ library to convert WGS84 latitude/longitude positions to/from Cartesian positions using Mercator projection. If You have MANDEYE with GNSS receiver, then it saves data in gnssXXXX.gnss files. This is ASCII file with

timestamp lat lon alt hdop satellites-tracked height age time fix-quality

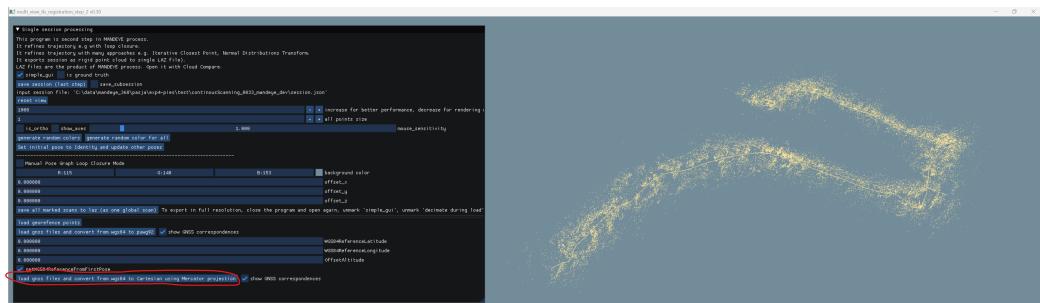


Figure 5.4: Georeferencing step 1: load gnss files and convert from wgs84 to Cartesian using Mercator projection.

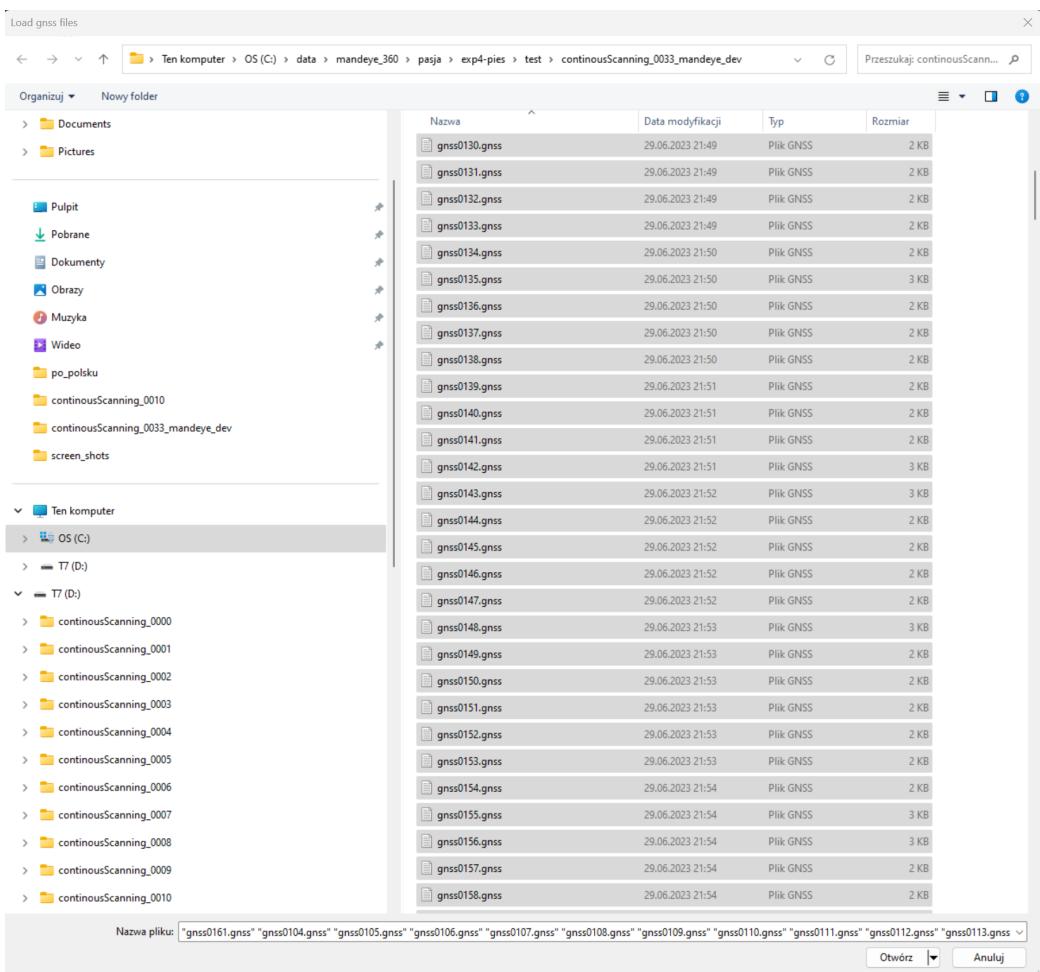


Figure 5.5: Georeferencing step 2: mark all gnss files and load.

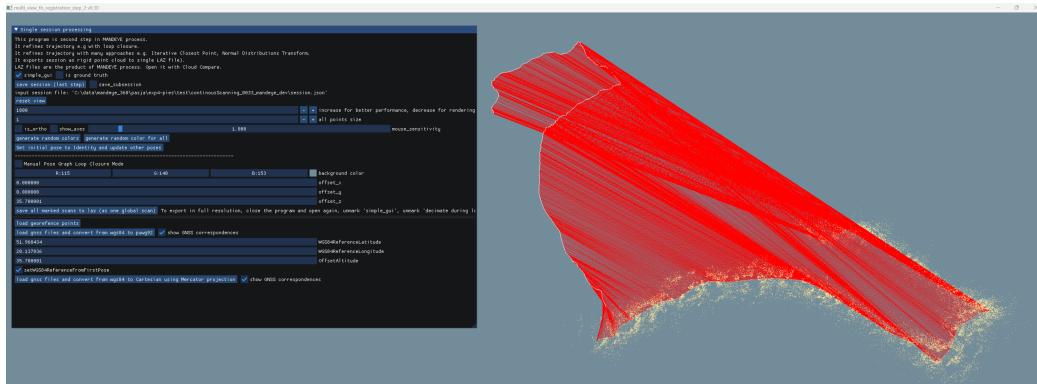


Figure 5.6: Georeferencing step 3: check/uncheck 'show GNSS correspondences' to see gnss-poses correspondences. Remark: You can use gizmo for manual initial trajectory to GNSS alignment.

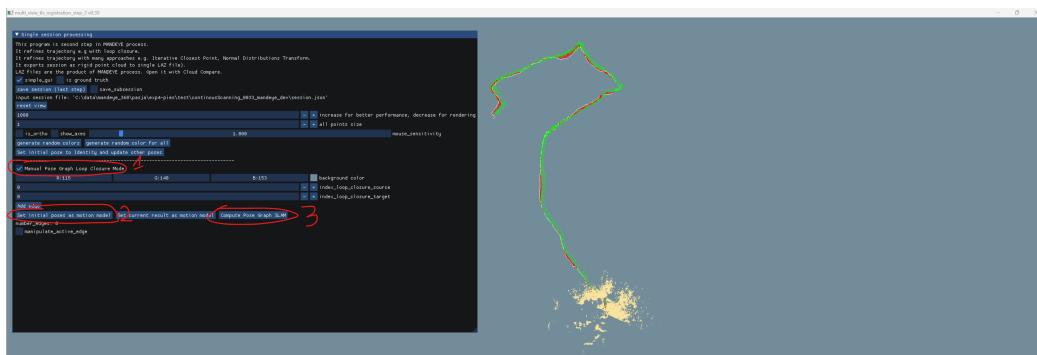


Figure 5.7: Georeferencing step 4: Check Manual Pose Graph Loop Closure Mode, then set initial poses as motion model, then Compute Pose Graph SLAM.

Chapter 6

Questions from end users

Do you have recommendations on how to best record data?

I recommend stop/scan mode for most accurate mapping. Continuous mapping is for increase the time of the survey.

How much distance can be between two consecutive start/stop acquisitions?

I suggest not more than 10 meters.

Do they need to overlap? To which degree?

Stop/scans should be overlapped at least 50%.

Continuous scanning: can the sensor change its tilt/angle during the recording phase? Or does it assume being in a upright position all the time?

I suggest that MANDEYE is somehow a upright position all the time.

How “fast” am I allowed to move (I actually did a rather slow walk).

I was tested it up to 8 km/h

Can the sensor change height while recording?

Yes.

Why first scan is blurry?

You should follow https://github.com/JanuszBedkowski/mandeye_controller/blob/main/doc/manual/manual_v0_2/mandeye_dev_manual_v0_2.pdf - section 2.2 turn on continuous scanning (MANDEYE DEV/PRO)

