

Pre-requisites.

OS Windows 10 or above

Python 3.10. Installation guide:

<https://www.digitalocean.com/community/tutorials/install-python-windows-10>

Installation Process.

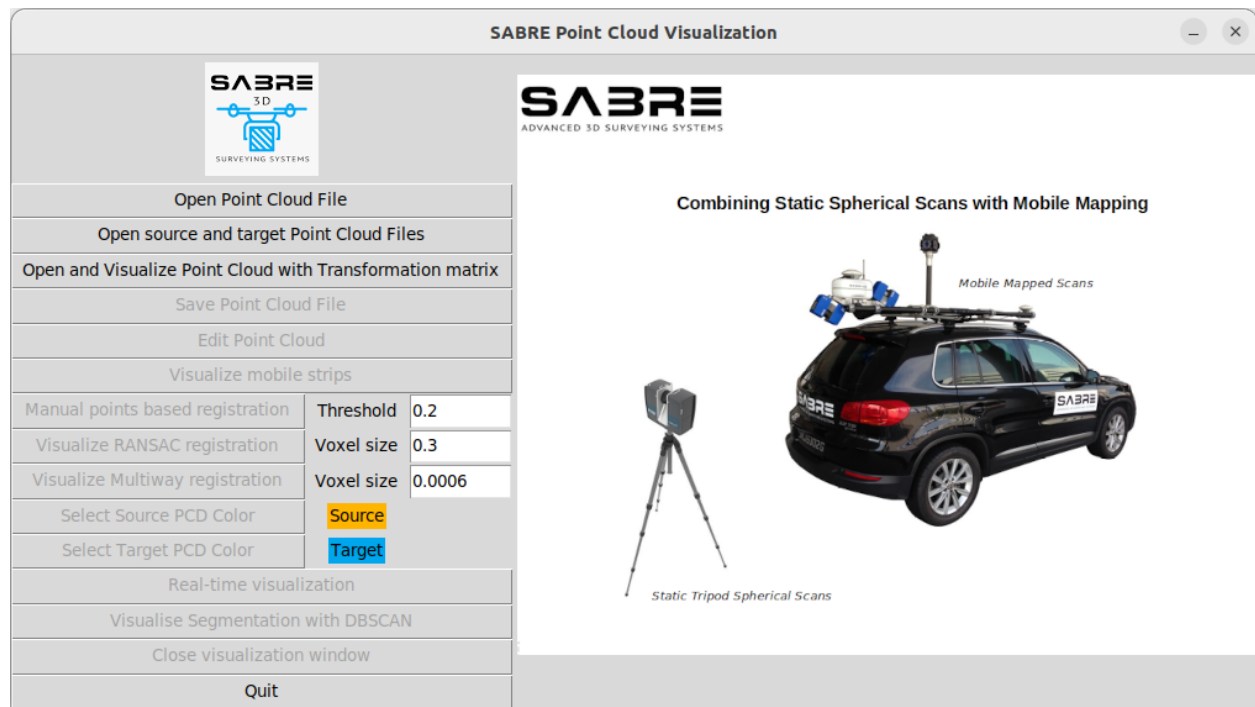
After the prerequisites are fulfilled.

1. Run the cmd with administrator rights
2. Go into the application main folder using the cd command.
Example: `cd some_folder\SABRE_app\`
3. Run `app_installation.bat` from the application main folder, this file installs all necessary libraries for the application and needs to be launched only once.

Now we are ready to launch our application,

launch file: `viz_run.bat`

Main interface.



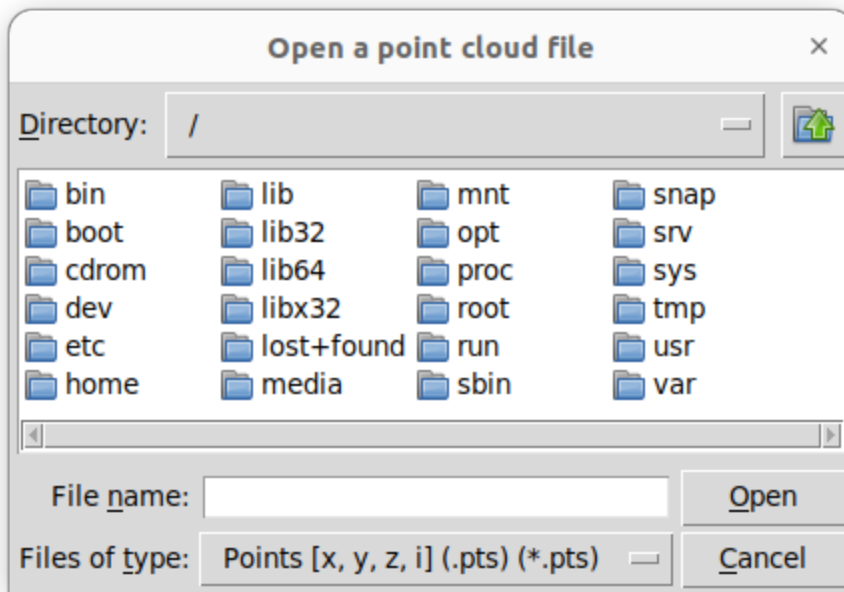
‘Open Point Cloud File’ and ‘Open source and target Point Cloud Files’ buttons.

Start working with the application by loading point cloud data for point cloud editing (the topmost button) or for 2 point clouds registration (the second button from the top). The rest of the buttons are greyed out and not available until the point cloud(s) is(are) loaded.

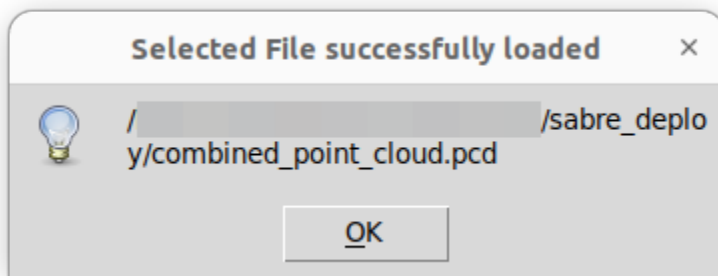
Point Cloud file formats the app can work with:

1. .pts files with xyz coordinates, xyz coordinates and intensity, xyz coordinates and rgb colors (inhouse built loader)
2. .e57 point cloud format (pye57 library)
3. .xyz, .xyzn, .xyzrgb, .ply, .pcd (open3d library)
4. .las, .laz files (laspy library)
5. .ply, .stl, .fdx, .obj, .off, .gltf, .glb Triangle mesh files (open3d library, Not tested!!!)

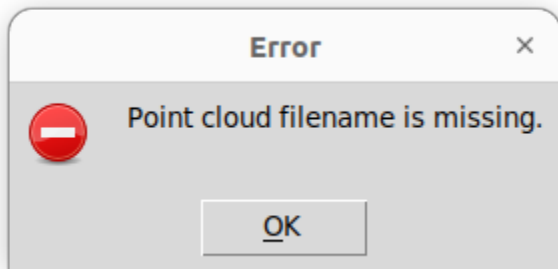
The Point Cloud file can be selected from the ‘Open Point Cloud’ dialogue menu.



When the point cloud for editing is loaded the corresponding message is displayed.



When the load file is cancelled the information error message is displayed.



After the point cloud data is loaded in the terminal user can see the message:

[Info] Successfully read /home/.../sabre_deploy/combined_point_cloud.pcd

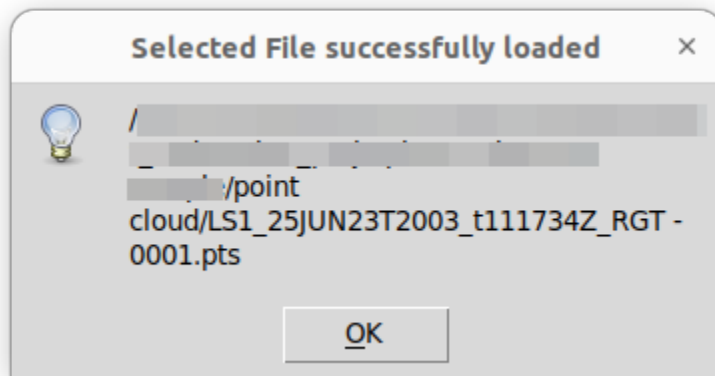
-----PCD:

PointCloud with 36071743 points.

This message informs the user that the point cloud file was read successfully and information on the point cloud amount of points.

The new options 'Save point cloud file' and 'Edit Point Cloud' become available from the main menu after the file is loaded.

The 'Open source and target Point Cloud Files' button leads the user to the 'Open Point Cloud File' dialogue menu. After the first (source) point cloud file loaded information message is displayed.



At the terminal, the information on the loaded file is displayed:

xyz points shape (6203468, 3)

intensity points len (6203468,)

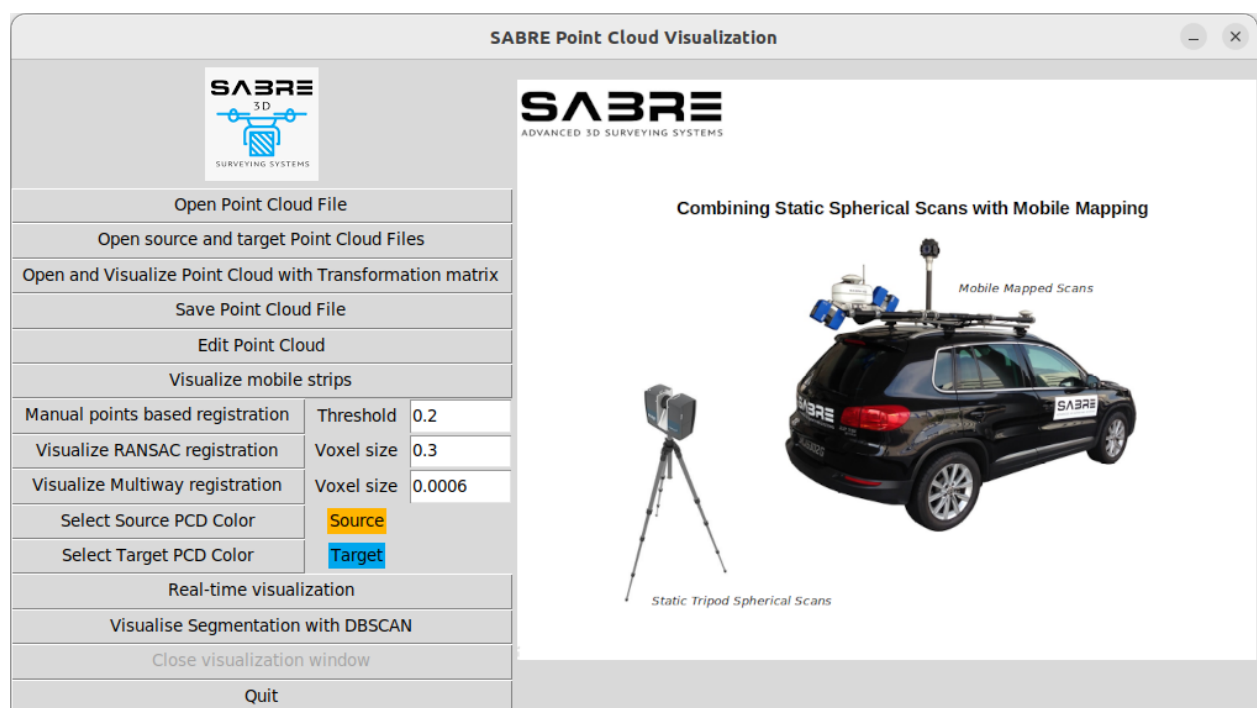
intensity_rgb points shape (6203468, 3)

[Info] Successfully read /.../point cloud/LS1_25JUN23T2003_t111734Z_RGT - 0001.pts

-----PCD:

PointCloud with 6203468 points.

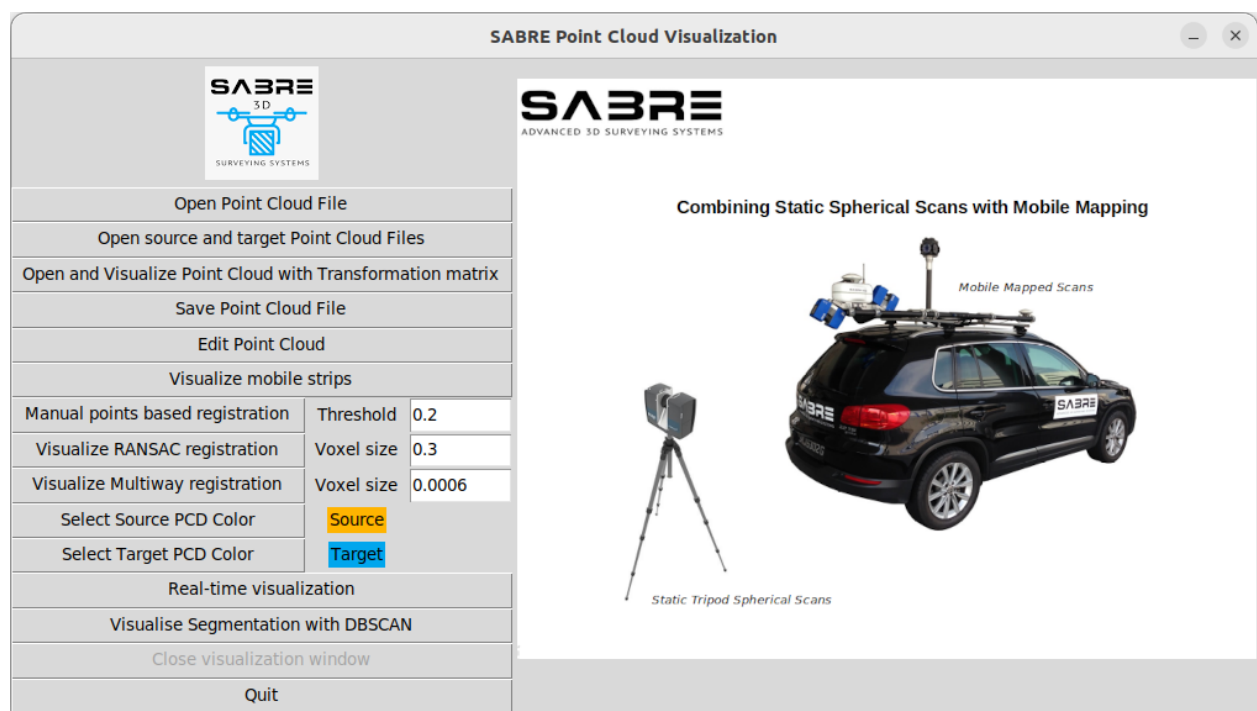
After the user confirmation, the app opens the dialogue to load the second (target) point cloud file. When the second file is loaded the information window and terminal information are displayed similarly as after the first file was loaded.



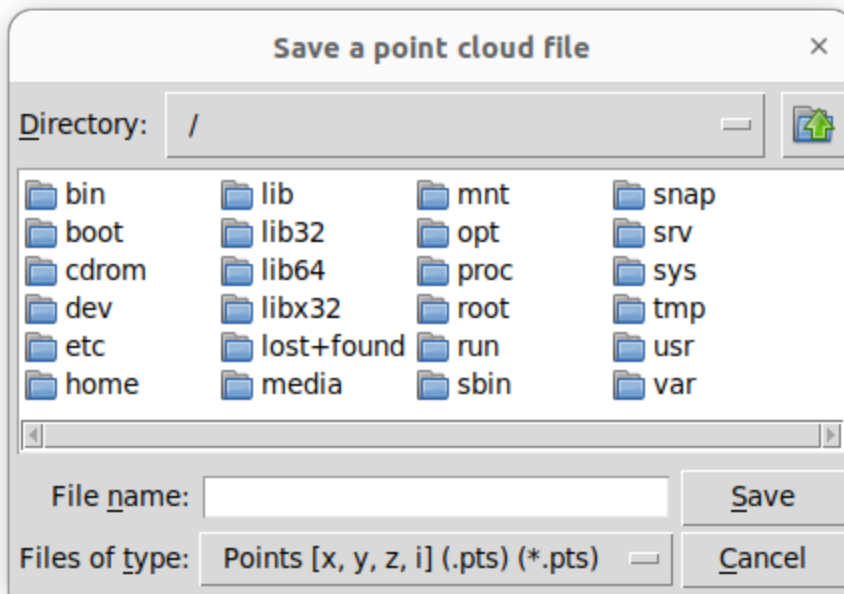
‘Open and Visualize Point Cloud with Transformation matrix’ button.

This functionality gives the possibility to the user to apply the transformation to the point cloud with the transformation matrix of dimensionality (4x4) saved previously with one of the registration processes. After the user presses this button the app asks the user to choose the point cloud file (process described above), and the transformation matrix .txt file. When the last file is loaded the transformation and visualization processes are initiated.

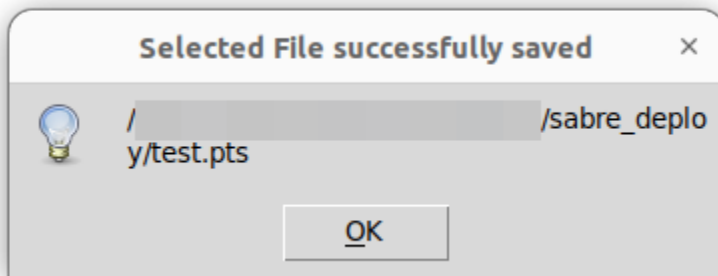
‘Save Point Cloud’ button.



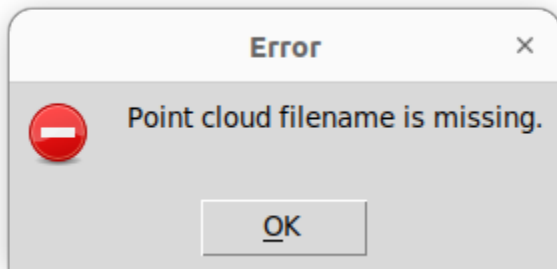
The ‘Save Point Cloud’ button gives the user access to the Save Point Cloud dialogue menu




This dialogue menu gives the user to choose the location and format of the saved point cloud file. The format choice is available from open3d library: .txt, .pts, .xyz, .xyzn, .xyzrgb, .ply, .pcd, triangle mesh files .ply, .stl, .fdx, .obj, .off, .gltf, .glb (mesh files are not tested!!!)
 Save confirmation message.



When the save file is cancelled the information error message is displayed.



The 'Save Point Cloud File' button saves the  first loaded point cloud (source) in case of two files were loaded (source and target).

'Edit Point Cloud' Visualisation screen.



Edit screen keys:

```
[Open3D INFO] -- Mouse view control --
[Open3D INFO] Left button + drag      : Rotate.
[Open3D INFO] Ctrl + left button + drag  : Translate.
[Open3D INFO] Wheel button + drag       : Translate.
[Open3D INFO] Shift + left button + drag : Roll.
[Open3D INFO] Wheel                  : Zoom in/out.
[Open3D INFO] -- Keyboard view control --
[Open3D INFO] [/]                : Increase/decrease field of view.
[Open3D INFO] R                  : Reset view point.
[Open3D INFO] Ctrl/Cmd + C       : Copy current view status into the clipboard.
[Open3D INFO] Ctrl/Cmd + V       : Paste view status from clipboard.
[Open3D INFO] -- General control --
[Open3D INFO] Q, Esc           : Exit window.
[Open3D INFO] H                : Print help message.
[Open3D INFO] P, PrtScn       : Take a screen capture.
[Open3D INFO] D                : Take a depth capture.
[Open3D INFO] O                : Take a capture of current rendering settings.
```

[Open3D INFO] Alt + Enter : Toggle between full screen and windowed mode.

[Open3D INFO] -- Render mode control --

[Open3D INFO] L : Turn on/off lighting.

[Open3D INFO] +/- : Increase/decrease point size.

[Open3D INFO] Ctrl + +/- : Increase/decrease width of geometry::LineSet.

[Open3D INFO] N : Turn on/off point cloud normal rendering.

[Open3D INFO] S : Toggle between mesh flat shading and smooth shading.

[Open3D INFO] W : Turn on/off mesh wireframe.

[Open3D INFO] B : Turn on/off back face rendering.

[Open3D INFO] I : Turn on/off image zoom in interpolation.

[Open3D INFO] T : Toggle among image render:
no stretch / keep ratio / freely stretch.

[Open3D INFO] -- Color control --

[Open3D INFO] 0..4,9 : Set point cloud color option.

[Open3D INFO] 0 - Default behavior, render point color.

[Open3D INFO] 1 - Render point color.

[Open3D INFO] 2 - x coordinate as color.

[Open3D INFO] 3 - y coordinate as color.

[Open3D INFO] 4 - z coordinate as color.

[Open3D INFO] 9 - normal as color.

[Open3D INFO] Ctrl + 0..4,9: Set mesh color option.

[Open3D INFO] 0 - Default behavior, render uniform gray color.

[Open3D INFO] 1 - Render point color.

[Open3D INFO] 2 - x coordinate as color.

[Open3D INFO] 3 - y coordinate as color.

[Open3D INFO] 4 - z coordinate as color.

[Open3D INFO] 9 - normal as color.

[Open3D INFO] Shift + 0..4 : Color map options.

[Open3D INFO] 0 - Gray scale color.

[Open3D INFO] 1 - JET color map.

[Open3D INFO] 2 - SUMMER color map.

[Open3D INFO] 3 - WINTER color map.

[Open3D INFO] 4 - HOT color map.

[Open3D INFO] -- Editing control --

[Open3D INFO] F : Enter freeview mode.

[Open3D INFO] X : Enter orthogonal view along X axis, press again to flip.

[Open3D INFO] Y : Enter orthogonal view along Y axis, press again to flip.

[Open3D INFO] Z : Enter orthogonal view along Z axis, press again to flip.

[Open3D INFO] K : Lock / unlock camera.

[Open3D INFO] Ctrl + D : Downsample point cloud with a voxel grid.

[Open3D INFO] Ctrl + R : Reset geometry to its initial state.

[Open3D INFO]	Shift + +/- : Increase/decrease picked point size..
[Open3D INFO]	Shift + mouse left button : Pick a point and add in queue.
[Open3D INFO]	Shift + mouse right button : Remove last picked point from queue.
[Open3D INFO]	
[Open3D INFO]	-- When camera is locked --
[Open3D INFO]	Mouse left button + drag : Create a selection rectangle.
[Open3D INFO]	Ctrl + mouse buttons + drag : Hold Ctrl key to draw a selection polygon.
[Open3D INFO]	Left mouse button to add point. Right mouse
[Open3D INFO]	button to remove point. Release Ctrl key to
[Open3D INFO]	close the polygon.
[Open3D INFO]	C : Crop the geometry with selection region.
[Open3D INFO]	

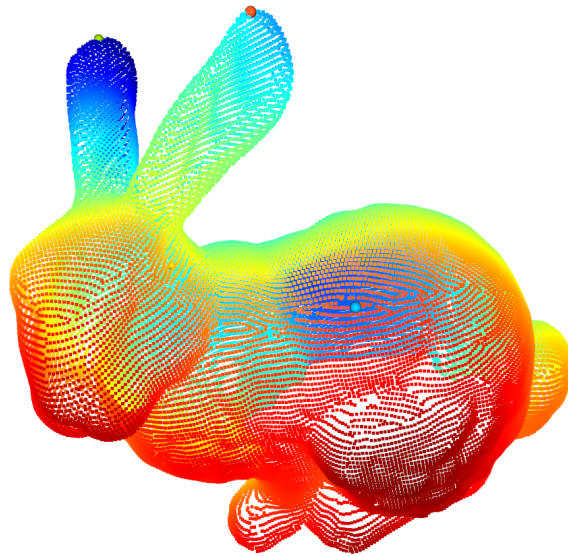
When the editing is finished the user needs to press q or x at the top right corner of the visualization window. After that, the visualisation window can be closed from the main menu by pressing the 'Close visualisation window' button.

'Visualize mobile strips' button.

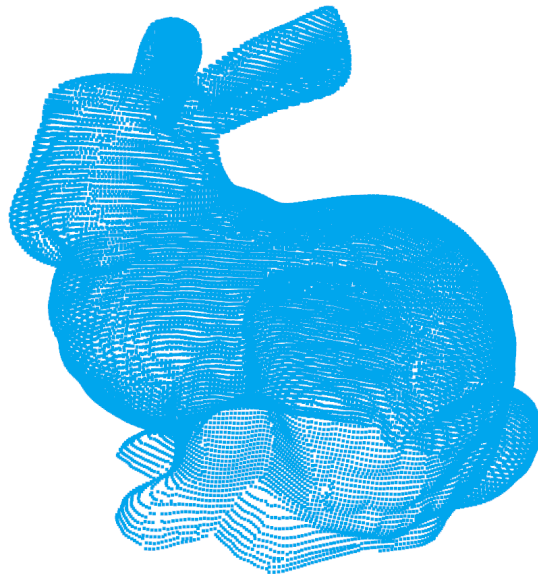
This button starts the attachment of two point clouds by the coordinates of the points. Can be used on two mobile stripe scans or two static scans made at the same location. The behaviour is unpredictable on the combination of the types of scans (Error or two points on the visualisation screen).

'Manual points-based registration' button.

This button launches the visualization of two point clouds on the same canvas and then the visualisation of the point clouds one by one for the user to select the points of correspondence. The threshold parameter is the threshold distance for the correspondence points search.



The correspondence points are the points from both point clouds based on which the alignment (registration) is made in this case. The correspondence points can be selected by the user by pressing [Shift + left mouse click]. The selected point coordinates and numbers are displayed in the terminal window. The user needs to select at least 3 points on each point cloud visualisation in the same sequence for each one. The amount of selected points must be the same. The more points selected the better alignment (registration) result is provided. After selecting the corresponding points and closing the visualizations the registration result visualisation is provided.



'Visualize RANSAC registration' button.

This button initiates the process of the RANSAC algorithm registration process. The voxel size parameter affects the down-sampling of the initial point clouds in the data preparation stage and the voxelisation during the registration process. The smaller the voxel size the closer the amount of points in the downsampled point cloud is to the amount of points in the original point cloud and the smaller voxel size during the RANSAC registration process. Selecting the voxel size too small may significantly prolong the registration process or lead the system to run out of memory. The big values of voxel size may lead to decreased accuracy of the registration. The result of the point clouds registration is saved as the 'ransac_combined_point_cloud.ply' point cloud file and the 4x4 transformation matrix is saved in the 'ransac_transformation_matrix.txt' text file.

'Visualize Multiway registration' button.

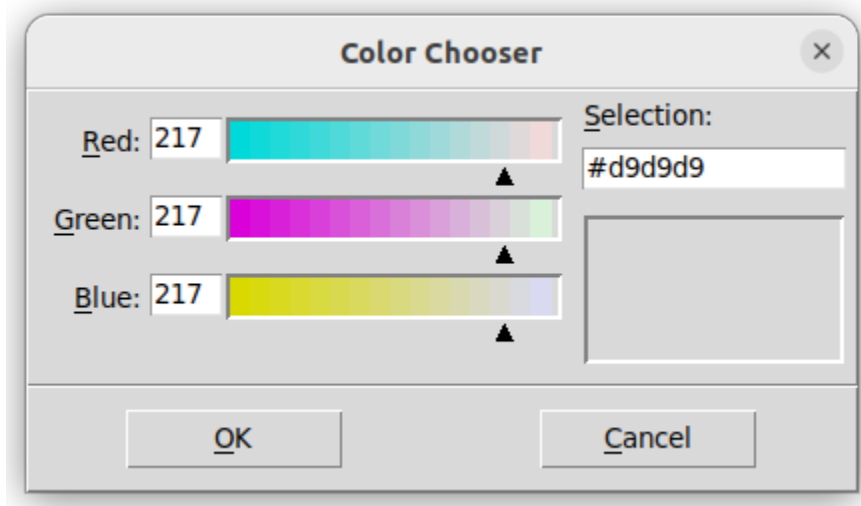
This button initiates the process of the Multiway algorithm registration process. The voxel size parameter affects the down-sampling of the initial point clouds in the data preparation stage and the voxelisation during the registration process. The smaller the voxel size the closer the amount of points in the downsampled point cloud is to the amount of points in the original point cloud and the smaller voxel size during the Multiway registration process. Selecting the voxel size too small may significantly prolong the registration process or lead the system to run out of memory. The big values of voxel size may lead to decreased accuracy of the registration.

The result of the point clouds registration is saved as the 'mw_combined_point_cloud.ply' point cloud file and the 4x4 transformation matrix is saved in the 'mw_transformation_matrix.txt' text file.

'Select Source PCD Color' and 'Select Target PCD Color' buttons.

These buttons are to select the colors of the source and target point clouds on the registration visualisation screen.

The colors can be selected by adjusting the Red, Green, and Blue scroll bars or inputting Hex or RGB values into the corresponding fields in the 'Color Chooser' window:

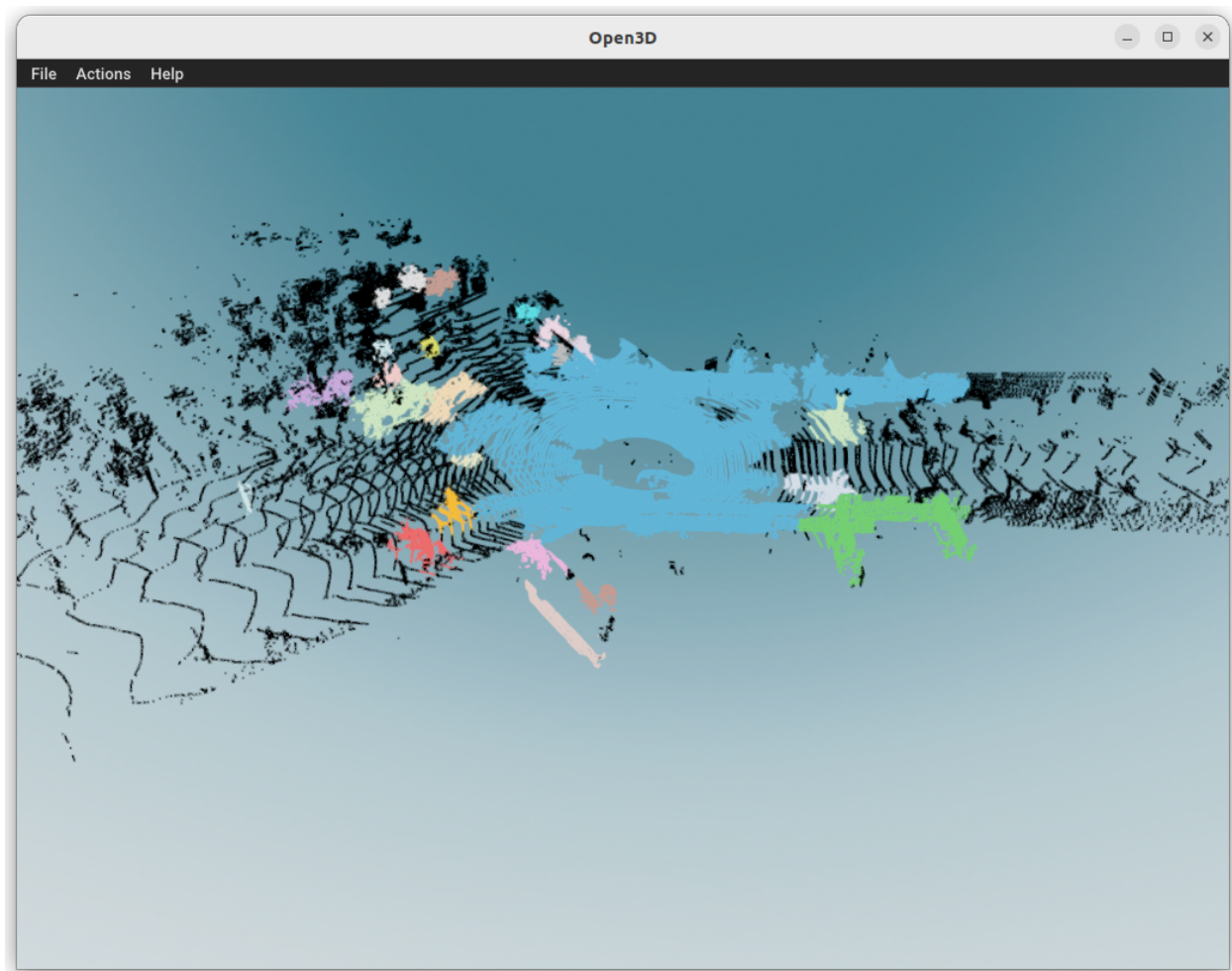


'Real-time visualization' button.

This button visualizes the opened point cloud and updates this visualization every 20 seconds. The update process is looking for the new point cloud data files in the '/real-time_workdir/' folder which is located in the main app folder. When the files are found the new points from these files are added to the visualization.

'Visualise Segmentation with DBSCAN' button.

This functionality visualizes the result of the density-based DBSCAN point cloud segmentation process. The algorithm finds the areas where the neighbour points are positioned close to each other and colors these areas with different colors.



'Close visualization window' button.

The 'Close visualization window' button closes the opened visualization window if any.

'Quit' button.

The 'Quit' button closes the app.