

Genuage User Manual

Version 1.0

Thomas Blanc

April 9, 2020

Abstract

A manual for the Genuage software, a point cloud visualization software utilizing VR technology, mainly intended for use with Single Molecule studies data.

It will detail the various functionalities of the software and how to use them.

Contents

1 Installation and Execution	3
1.1 System Requirements	3
1.2 Supported VR Hardware	3
1.3 Installation and Launch	3
2 Desktop Interface	3
3 Data Loading	4
3.1 File Format	4
3.2 Load Data From File	4
3.3 Load Data From TCP/IP Transfer (Matlab/Python Supported)	5
4 Data Saving	5
4.1 Saved Metadata Format	5
4.2 Save Selected Points (Cropping)	5
5 Visualization	6
5.1 Point Cloud Visualization	6
5.1.1 Point Rendering	6
5.1.2 Internal software variables for point clouds	6
5.1.3 Visualization Parameters	6
5.2 Point Trajectories Visualization	8
5.2.1 Trajectories Rendering	8
5.2.2 Frames Display Options and Animation	9
5.2.3 Internal software variables for trajectories	9

5.2.4	PointOverlay	9
5.3	Point Orientation Visualization	9
5.3.1	Orientations Rendering	9
5.3.2	Internal software variables for orientations	9
6	Cloud Calculations	10
6.1	Density Calculation	10
6.1.1	Density Calculation Method	10
6.1.2	Density Calculation Result	10
7	Operations On Loaded Clouds	11
7.1	Change software variables configuration and reload the file data	11
7.2	Cloud Linking	12
8	VR Interactions	13
8.1	Controller interface layout	13
8.1.1	HTC Vive Controller	13
9	VR Tools	13
9.1	Pointer Tool	13
9.2	VR Tools Menu	14
9.3	Interface for the Object Placement Tools	15
9.4	Object Placement Tools - Quantification Tools	18
9.4.1	Counter Tool	18
9.4.2	Ruler Tool	18
9.4.3	Angle Measure Tool	19
9.4.4	Profiler Tool	20
9.4.5	Object Placement Tools - Selection Tools	21
9.5	Note on Point Selection	21
9.6	Selection Sphere Tool	21
9.6.1	Selection Convex Hull Tool	21
9.7	Visualization Tools	23
9.7.1	Clipping Plane Tool	23
10	Native Plugins : C/C++ powered analysis tools	24
10.1	Infer3D : 3D Inference on trajectories	24

1 Installation and Execution

1.1 System Requirements

SYSTEM REQUIREMENTS (MINIMUM)

GPU : Nvidia Geforce 1050 or equivalent

Operating System : Windows 7, 8 or 10

Other : SteamVR must be installed for VR functionalities to work

SYSTEM REQUIREMENTS (RECOMMENDED)

GPU : Nvidia Geforce 1070 or equivalent

Operating System : Windows 7, 8 or 10

Other : SteamVR must be installed for VR functionalities to work

1.2 Supported VR Hardware

VR Hardware :

- HTC Vive / Valve Index with Vive controllers

- Oculus Rift / Oculus Rift S with Oculus Touch controllers

(Other headsets may work for visualization purposes, but the controller layout was made for HTC Vive controllers.)

1.3 Installation and Launch

The software is packaged into a standalone folder with an executable file, no libraries other than SteamVR are required to install and run it.

2 Desktop Interface

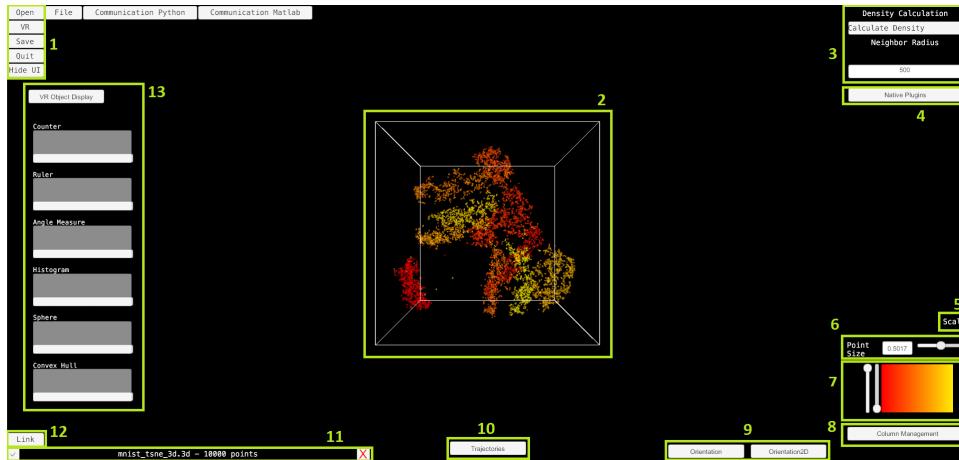


Figure 1: Genuage Desktop Interface

1. Main Menu (Contains buttons to load files, change to VR or Desktop mode, save data, quit the software or hide the on screen UI)
2. Loaded point cloud
3. Density Calculation menu
4. Native Plugins Menu
5. Cloud Point Scaling menu
6. Point Size slider
7. Colormap selection and saturation menu
8. Data Reload menu
9. Orientation Display buttons
10. Trajectory display button
11. Data selection button (click the center part to select, red cross button to close the data, toggle to select the dataset for cloud linking)
12. Cloud Linking button
13. VR Objects Display menu

3 Data Loading

3.1 File Format

Genuine supports ASCII file formats, the conventions are as follow : every row represents one point localization, and columns are separated with tabs, each column represent one parameter of the data: localization, trajectory, orientation...

Genuine can read any text file extension and is unit agnostic.

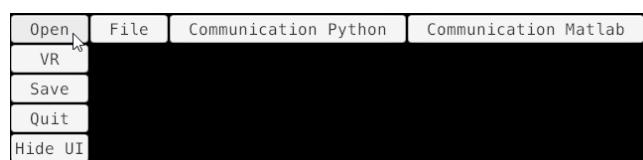


Figure 2: Data Loading UI

3.2 Load Data From File

To load data from a file, click the appropriate button in the load menu, this will open a browser window to select the files to load. [IMAGE]

3.3 Load Data From TCP/IP Transfer (Matlab/Python Supported)

To load data from a byte transfer, click the appropriate button in the load menu, this will open a window informing the user that the software is waiting for the transfer, then, run the script to send the data from your Python or Matlab program of choice to transfer your data. Genuage will load the point after receiving and processing everything.

Scripts for python and matlab are provided.

4 Data Saving

Genuage has functions to save clouds with metadata files so the software can reload the parameters previously set and the operations the user did previously. The software also allows the user to save selected regions of points into a new file.

4.1 Saved Metadata Format

When a full point cloud is saved, Genuage also creates a metadata file in the .JSON format, this file contains the following information • The visual parameters used during the session

- The List of all selected points
- Data about the VR objects placed in the data
- Results of various analyses on the data

Genuage uses these metadata to reload the user's configuration and analyses when the saved file is loaded next.

A matlab script is provided to extract data from the metadata files.

4.2 Save Selected Points (Cropping)

This function saves all the selected points into a new file, no metadata will be saved.

5 Visualization

5.1 Point Cloud Visualization

5.1.1 Point Rendering

The points are rendered as transparent spheres, with an amplifying effect on their luminosity when they overlap.

5.1.2 Internal software variables for point clouds

To represent each points, Genuage uses five variables, the position in the x y and z axes, the color value or intensity, and the time of apparition.

5.1.3 Visualization Parameters

ColorMaps

Points are colorized according to the normed value of their **Color** variable, colors are taken from a color map texture.

Genuage offers the following standard color maps to colorize the points.

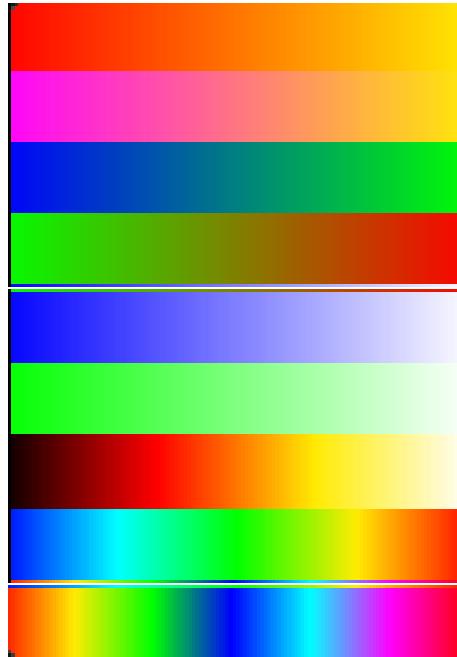


Figure 3: ColorMaps

1. autumn
2. spring

3. winter
4. greenred
5. Blues
6. Greens
7. hot
8. jet
9. hsv

Change Scale

The clouds can be scaled globally with one value or with separate value for each axis.

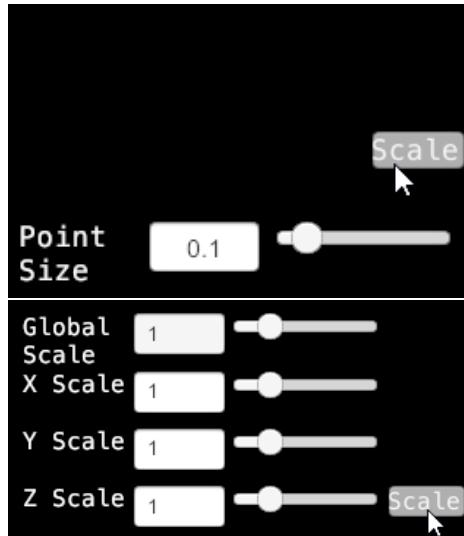


Figure 4: Where to find the scale option in the UI.

Change Point Size

The global size of the points can be set, with a slider or a direct input.



Figure 5: UI used to set the point size

Thresholding

Thresholds can be defined to exclude points above or under specific parameters :

- x coordinates
- y coordinates
- z coordinates
- color
- time

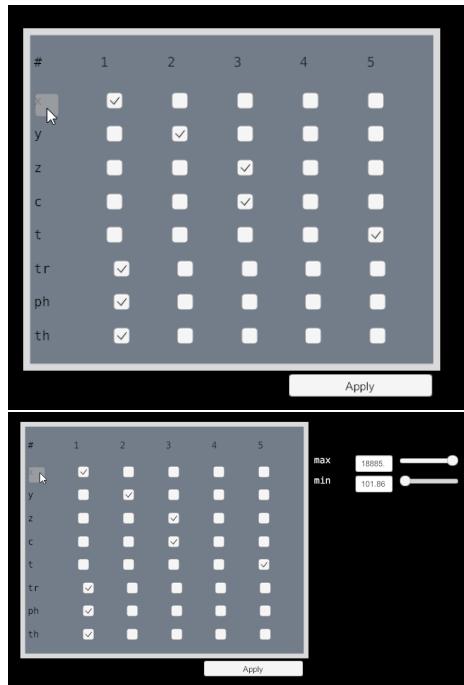


Figure 6:

5.2 Point Trajectories Visualization

5.2.1 Trajectories Rendering

Trajectories are rendered as lines linking the points.

Individual trajectories are represented by the **Trajectory** variable, which is used as the identifier for all the points belonging to one trajectory.

The order of the points in a trajectory are represented by the **Time** variable.

Trajectories are by default colorized by their id on the **Jet** colormap.

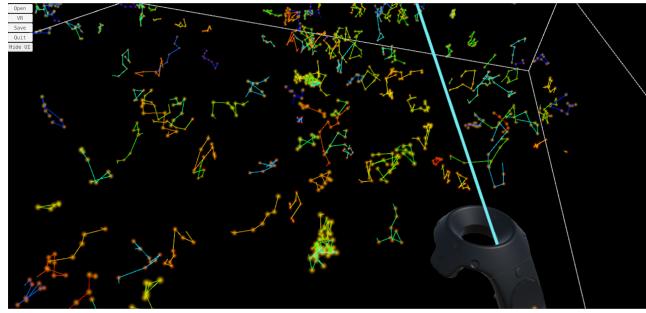


Figure 7: Point Trajectories

5.2.2 Frames Display Options and Animation

the trajectory interface allows the user to visualize the evolution of trajectories in time, either automatically with animations running at different speed, or manually by shifting through the frames of time.

The **dragontail** option is used to determine how many points should remain visible after the current frame.

5.2.3 Internal software variables for trajectories

Trajectories are defined by an id value shared by all the points in the trajectory, the order of apparition of the points is determined by their time value.

5.2.4 PointOverlay

It is possible to overlay the render of the point localizations and even the orientations to the trajectory display.

5.3 Point Orientation Visualization

5.3.1 Orientations Rendering

Orientations are rendered as lines going through the points.

5.3.2 Internal software variables for orientations

Orientations of the points are represented by the two angles of the spherical coordinate system (phi and theta), in radians.

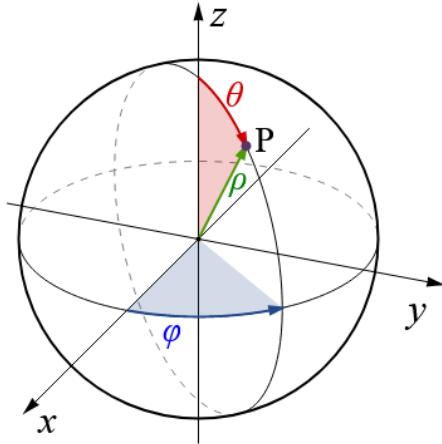


Figure 8: Spherical coordinates system

6 Cloud Calculations

6.1 Density Calculation

Genuage can calculate local density for each points of a dataset, with a neighbor radius given by the user.

6.1.1 Density Calculation Method

Because the density calculation method is a high complexity problem, a simplified approach was used in Genuage to speed up processing times when dealing with large amounts of data.

The space occupied by the dataset coordinates is split into a number of regions and all the points each of the regions are detected, when calculating the number of neighbors for each points, the algorithm only looks at all the other points in the region and the adjacent regions.

The local density calculated for each point corresponds to the number of points that have a distance inferior to the specified **Neighbor Radius**.

6.1.2 Density Calculation Result

When the density has been calculated for all points, a new data column appears in the **Data Reload** menu, labelled "d".

It represents the number of neighbors within the radius chosen, and can be set to represent any of the **software variables**.

An example of use is to set the **color** variable to this column in order to color all the points based on their density.

7 Operations On Loaded Clouds

7.1 Change software variables configuration and reload the file data

This function is used to change which column represents which software variable for an already loaded dataset. The selection is done by ticking the checkboxes in the reloading window, one column can be linked to several variables at once.

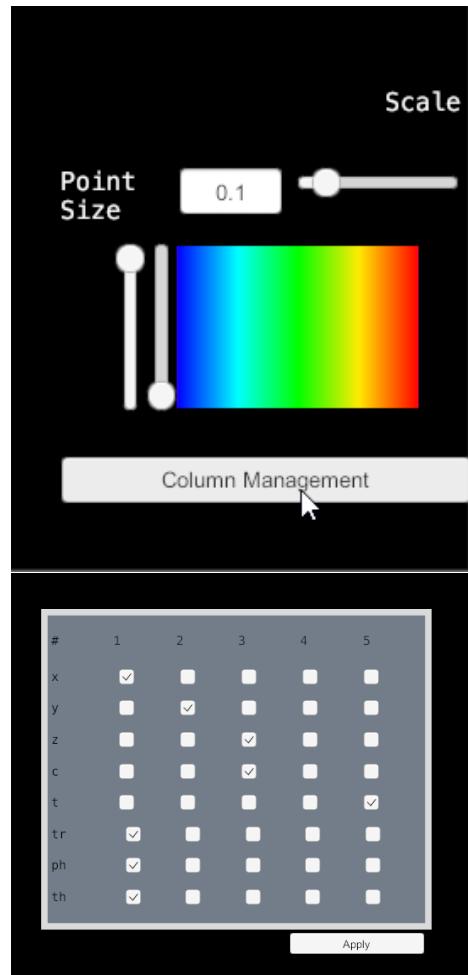


Figure 9:

File Columns		1	2	3	4	5	6	7	8
Software Variables	X Y Z Color Time Trajectory Theta Phi	x 							

7.2 Cloud Linking

The Cloud Link function is used to represent two or more clouds in the same space.

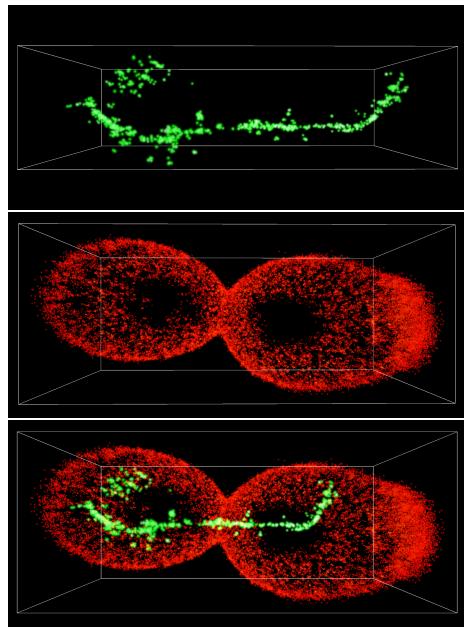


Figure 10: Example of two clouds and the result when they are linked

To link loaded clouds, the toggle at the left side of their selection button must be on.

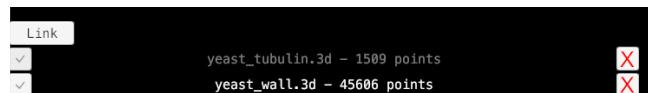


Figure 11:

8 VR Interactions

8.1 Controller interface layout

Below are detailed the button layouts of the controllers supported by the software.

Only one controller is necessary to manipulate the data and use all of the VR tools.

8.1.1 HTC Vive Controller

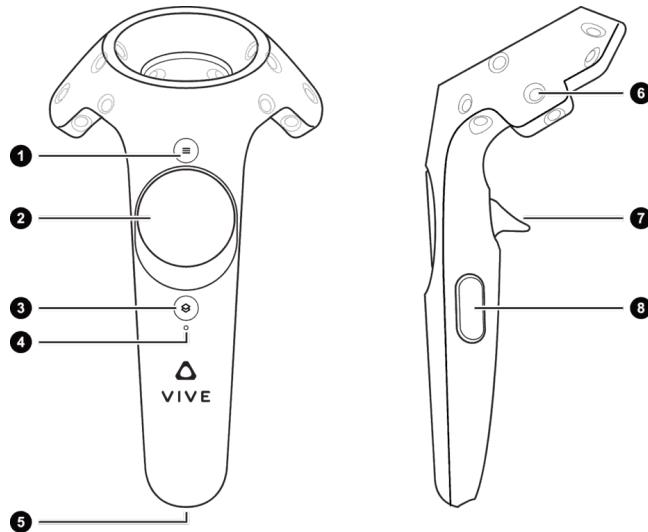


Figure 12: HTC Vive controller button layout

1. VR Tools Menu Button
2. Trackpad (used differently by various tools, see further sections)
3. Controller Power Button
4. Power Light
5. Micro USB Charging Port
6. IR Receptors for tracking
7. Trigger (used to grab objects and interact with the VR interface)
8. Side Grips (not used)

9 VR Tools

9.1 Pointer Tool

The Pointer tool allows the user to grab some objects and interact with vr interface elements by pointing the controller at them and pressing the

Trigger.

The laser pointer linked to the controller model is used to represent the tool, it is blue normally and turns green when it points to an object that can be interacted with.

The Pointer tool is always active.

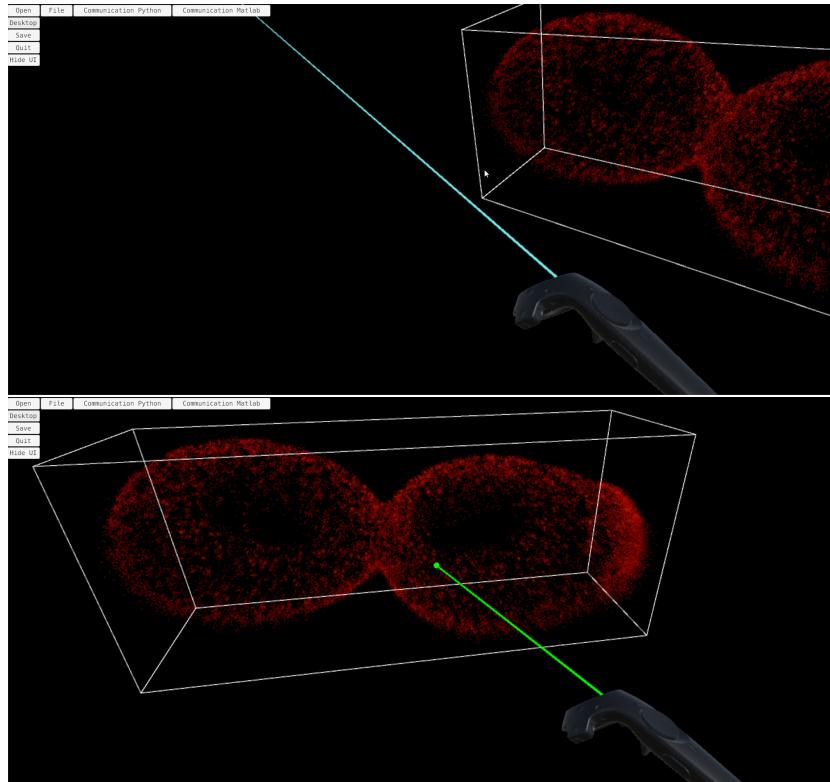


Figure 13: Pointer tool

9.2 VR Tools Menu

The Tool Menu is displayed and hidden by pressing the **Menu** button on the controller.

The menu can be moved by grabbing the white square above the buttons with the **Pointer** tool, and the buttons are used to activate each of the VR tools.

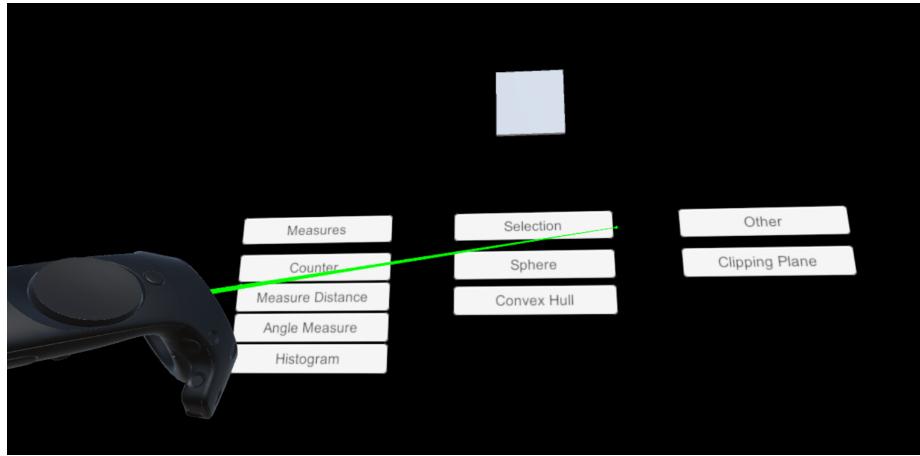


Figure 14: VR Tools Menu

9.3 Interface for the Object Placement Tools

The Object Placement Tools are tools allowing the user to place objects inside the data for various purposes. All these tools use the same interface for object placement.

The objects placed all belong to an Object Set, and multiple sets can be created. To do this the **Trackpad** of the controller is used.

The **Trackpad** is a tactile part of the HTC Vive controller, here different zones of the trackpad will correspond to different buttons. An image is associated with each zone, and when the user puts their finger on the corresponding side, the image is colored in green and a descriptive text pops up next to it to explain what the button does.

The layout of the two interfaces, one to place the objects, the other to select the sets, are detailed below :

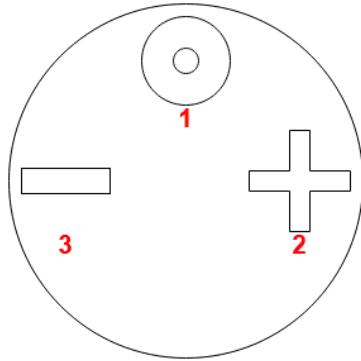


Figure 15: interface for object placement

1. Go to Set Selection (activates the set selection interface)
2. Add a new object to the set
3. Delete an object from the current set (said object must be touched by the sphere pointer)

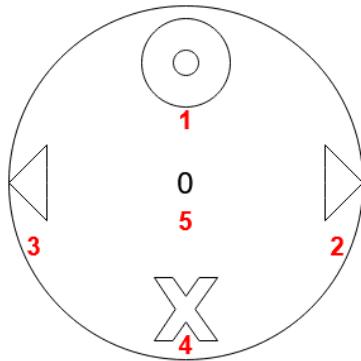


Figure 16: interface for object set selection

1. Go to Object Placement for the current set(activates the object placement interface)
2. Go to next set
3. Go to previous set
4. Delete current set
5. Current set

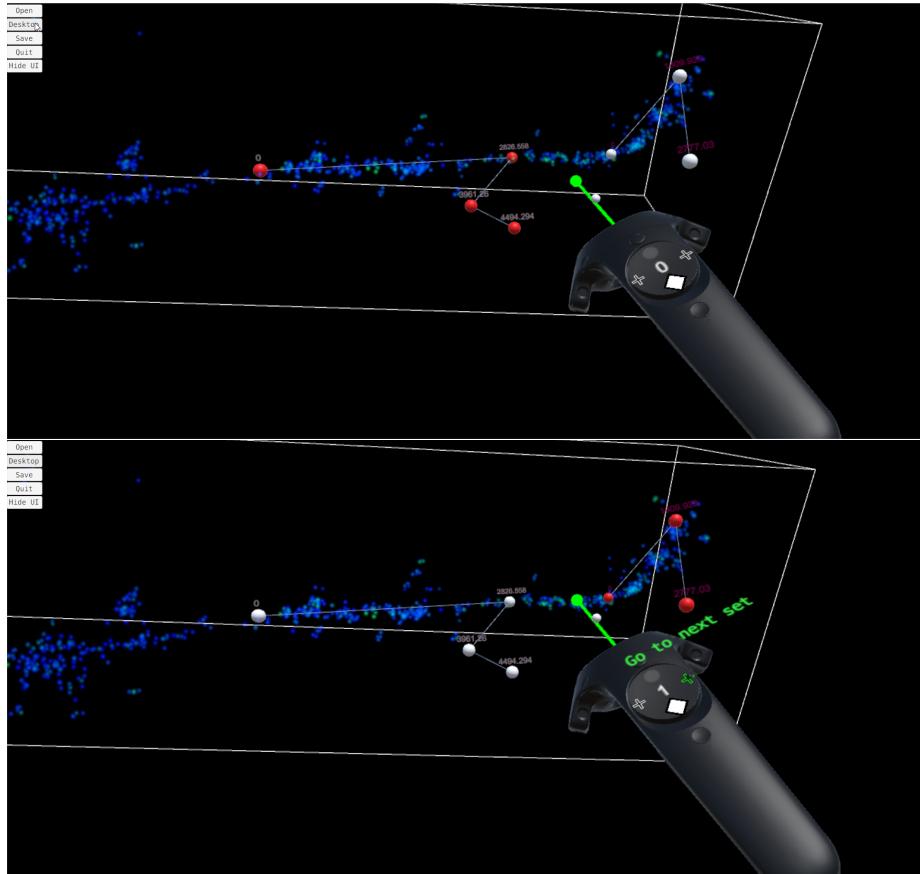


Figure 17: Example of the Set system

Here the points making up the two distance measures belong to Set 0 and 1 respectively and they are highlighted in red when the corresponding set is selected with the Set Selection Interface.

9.4 Object Placement Tools - Quantification Tools

9.4.1 Counter Tool

The **Counter** tool is a simple tool used to place numbered spheres inside the data. Each sphere set has a specific color for the text.

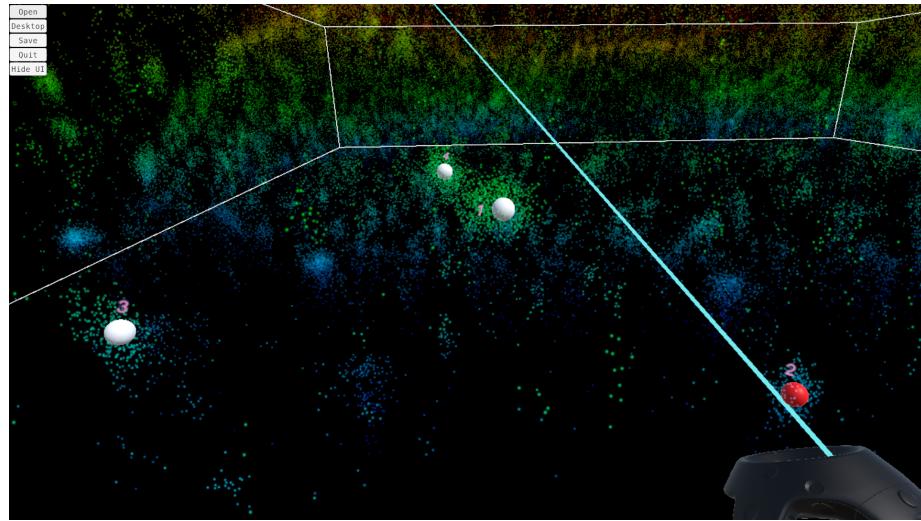


Figure 18: Counter Tool

9.4.2 Ruler Tool

The **Ruler** tool is used to measure distances inside the data. The spheres placed are linked and the distance from each sphere to the first one is displayed above it. Each sphere set represent one measure.

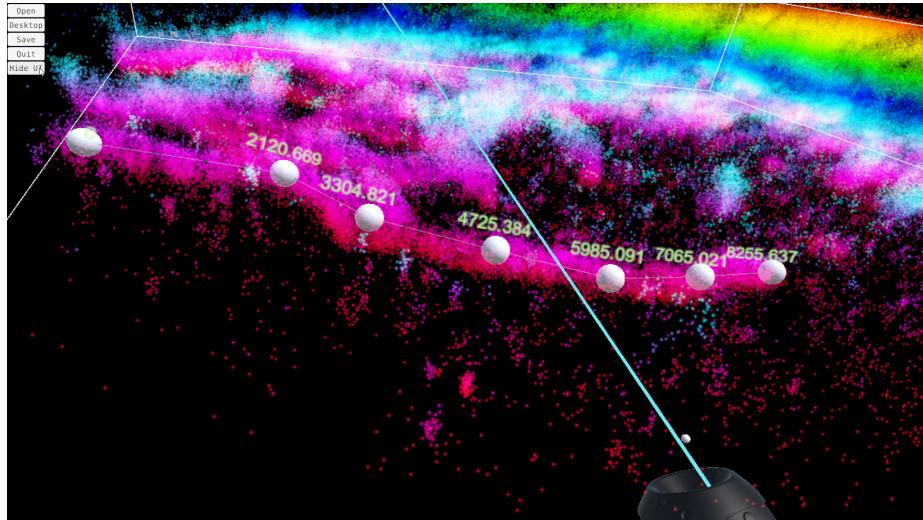


Figure 19: Ruler Tool

9.4.3 Angle Measure Tool

The **Angle Measure** tool is used to measure angles inside the data. Every two spheres placed form a segment, and once four sphere have been placed, the angle in degrees between the two segments is displayed. Each set represents one angle.

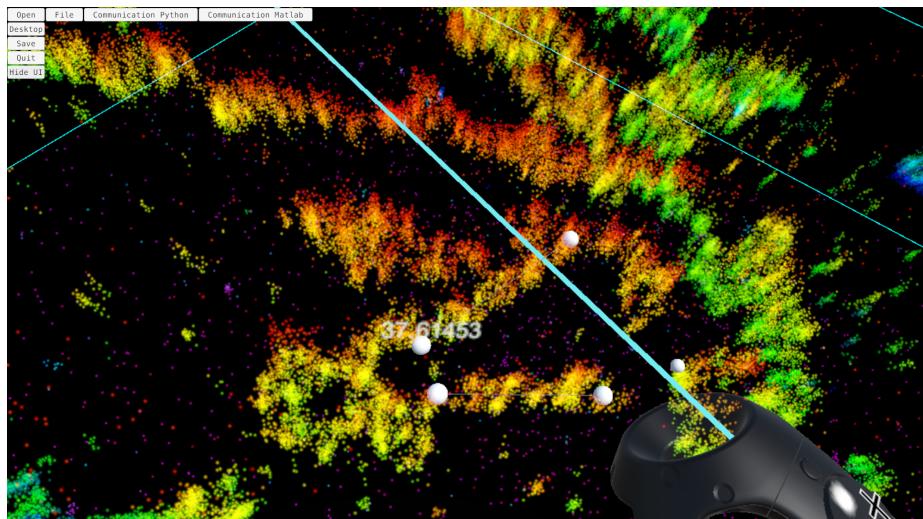


Figure 20: Angle Measure

9.4.4 Profiler Tool

The **Profiler** tool is used to get a profile of the point distribution inside an area of the dataset.

The tool places a cylinder inside the data to determine the area to measure into, then divides the cylinder into multiple slices or blins.

All the points in each blin will be counted and the results displayed as a histogram in a VR window.

To create the cylinder, push the **Add Object** button a first time, this will create both ends of the cylinder, the first one staying at the place the controller was pointing to when the button was pushed, and the other will stick to the controller, allowing the user to move it.

Then place the controller to where the other end of the cylinder should be. Press the **Add Object** button a second time to validate the position. After both ends of the cylinder are placed, move the controller closer or away from it to set the height of the cylinder, press the **Add Object** button a second time to validate the height.

When the height is set, the cylinder is complete, and the controller will switch to the blin selection interface. When the number of blins in the cylinder is changed, a result VR window will appear and display a histogram of the number of points in each section of the cylinder.

When the operation is complete, use the **Validation button** of the interface to finish the process and return to the **Object Placement interface**.

The interface for the blins is detailed below :

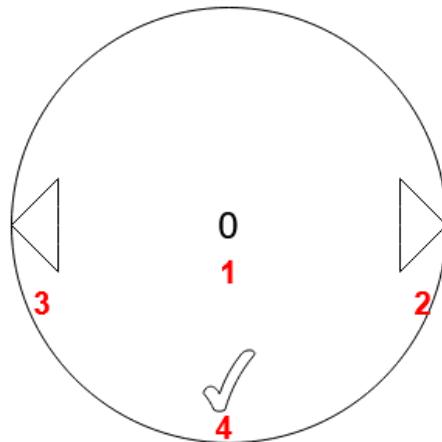


Figure 21:

1. Number of blins
2. Add one more blin
3. Subtract one blin

4. Validate the number of blins and exit to Object Placement Interface

9.4.5 Object Placement Tools - Selection Tools

9.5 Note on Point Selection

To determine the points that are selected, Genuage sums up all the sources of selection (I.E all the selection regions created by the tools detailed below).

Furthermore, selected points that are marked as hidden (points that are outside the thresholding limits for example) will not be taken into account for any selection operation.

9.6 Selection Sphere Tool

The **Selection Sphere** tool allows the user to draw a sphere inside the data. To use it, place the controller where the center of the sphere should be, press and hold the **Add Object** button. While holding the button, move the controller away to set the preferred radius for the sphere, once it's set, release the button to end the placement.

Once the sphere is set, all the points inside will be marked as selected, their color turning green.

Each set represents one sphere.

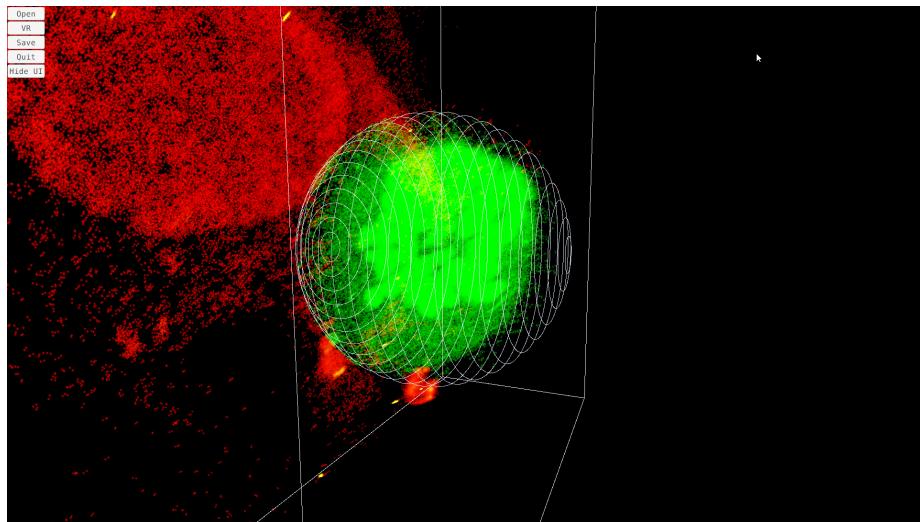


Figure 22: Sphere

9.6.1 Selection Convex Hull Tool

The **Selection Convex Hull** tool is used to draw a custom shape to select specific areas. It places spheres inside the data, once four spheres have been placed, they are linked and form a volume. All the points

inside are marked as selected and colored green. Further points added will expand and shape the volume and select more points.
Each set represents one convex hull and one selection volume.

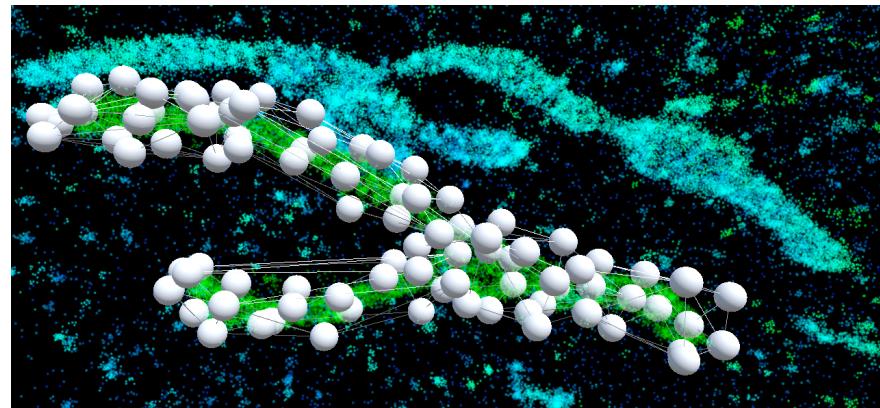


Figure 23: Convex Hull

9.7 Visualization Tools

9.7.1 Clipping Plane Tool

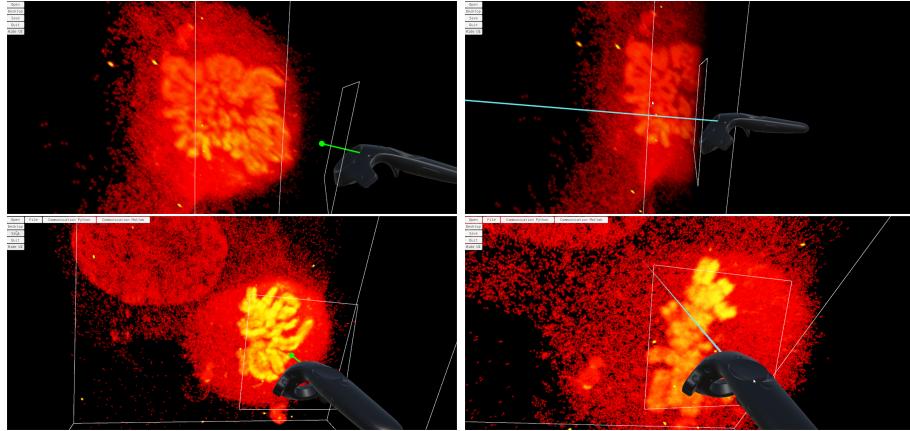


Figure 24: Clipping Plane

The **Clipping Plane** tool puts a plane, represented by a square in front of the controller that will remove all points behind it from display. It allows the user to dig into the data using the controller. This tool is compatible with the Object Placement Tools.

10 Native Plugins : C/C++ powered analysis tools

Native Plugins are libraries of C or C++ code that can be accessed by Genuage to perform various analyses.

Below is a list of the pre-installed native plugins and their functions.

10.1 Infer3D : 3D Inference on trajectories

Infer3D is a Bayesian inference script used to calculate the diffusion coefficient and possibly the 3D force of a sample of point trajectories. Infer3D runs on a sample of point trajectories that are taken from all the selected areas in the currently active point cloud.

The INfer3D code optimizes the result a given function that can be selected in the UI :

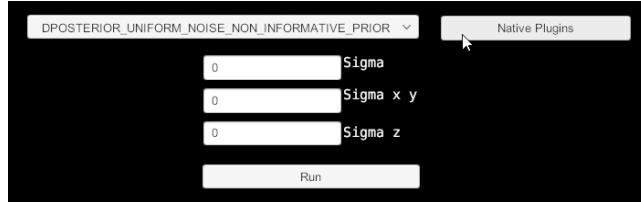


Figure 25: UI for the Infer3D Native Plugin

The different functions that can be selected represent different hypotheses on the environment, different priors and posteriors.

Three variables can also be set, **Sigma**, **Sigmaxy** and **Sigmaz**, they represent the noise in microns, either the global noise for **Sigma**, the noise on the x and y axes for **Sigmaxy** and the noise on the z axes for **Sigmaz**

List of all the function options

Pure diffusion functions :

1. Likelihood No noise
2. Likelihood Uniform noise*
3. Likelihood Asymmetric noise**/***
4. Likelihood noise only in z***
5. posterior no noise non informative prior
6. posterior no noise conjugate prior
7. posterior uniform noise non informative prior*
8. posterior asymmetric noise non informative prior**/***
9. posterior noise only in z non informative prior***

Diffusion and velocity functions :

1. simple Posterior function*

2. Likelyhood No noise
3. Likelyhood Uniform noise *
4. Likelyhood Asymmetric noise **/***
5. Likelyhood noise only in z ***
6. posterior no noise non informative prior
7. posterior no noise conjugate prior
8. posterior uniform noise non informative prior *
9. posterior asymmetric noise non informative prior **/***
10. posterior noise only in z non informative prior ***

* : uses the Sigma variable

** : uses the Sigma x y variable

*** : uses the Sigma z variable