-Manual-

Status: June 14, 2014

Carsten Fittler
Ottmar Kullmer
Sonja Leitsbach
Alexandra Papagno
Heiko Zimmermann

Investigator: **Senckenberg Forschungsinstitut Frankfurt am Main**
Abteilung Palaeoanthropolgie und Messelforschung
Sektion Tertiaere Saeugetiere
PD Dr. Ottmar Kullmer
Senckenberganlage 25
60325 Frankfurt/Main
`http://www.senckenberg.de`

DFG Forschergruppe 771
Steinmann-Institut fuer Geologie,
Mineralogie und Palaeontologie
Nussallee 8
53115 Bonn
`http://www.for771.uni-bonn.de`

Contractor: **ZiLoX IT**
Heiko Zimmermann & Stephan Lorsbach GbR
Hauptstraße 60
55595 Wallhausen
`http://www.zilox-it.de`

Entirely written in LaTeX

Contents

Preface

The *Occlusal Fingerprint Analyzer* (OFA) was created as part of the research project conducted by research group 771 of the University of Bonn. Its focus is the function and evolution of mammalian dentition. The reconstruction of movement patterns of both fossilized and recent mammalian teeth is based on the striae of tooth surfaces, as these striae are available in recent as well as in fossil material.

As objective of the project
"Function and increased efficiency in the dentition of mammals - phylogenetic and ontogenetic influences on the chewing apparatus."
the influence of tooth morphology on jaw movement throughout mammal phylogeny and ontogeny is studied.

Besides the usual methods, the research group applies three-dimensional recording and imaging techniques to describe the complicated interaction of the contact surfaces of upper and lower dentition. This virtual simulation of chewing patterns will yield new insights into the development of attrition facets and improve knowledge of mammal dentition function.

The programming of the *Occlusal Fingerprint Analyzer* was outsourced to a company in the private sector. The intention was to write an application running on multiple operating systems such as Windows or Linux. In order to make the software available to a large group of scientists even after termination of the research project, an open source license was chosen.

The software has been developed from 2008 until 2014 and is always open fur suggestions and improvements.

Have fun,
the developers
Wallhausen, March 2014

Part I.

Introduction

1. Purpose and Application

For most anthropologists, the distinguishing trademark of humankind is the capability to think about what has been and what will be. A concept of past and future seems to be unique to humans, although there are hints that our next of kin - gorillas, bonobos and chimpanzees - are emulating us.

The mysteries surrounding what has occurred in the past and especially *why* that past has occurred, have driven scholars throughout history. With nearly seven million years of human evolution, only in the small portion of that time in which writing existed, a mere 4000 years, is the luxury of using historical documents as an information source available.

Hard evidence, the remains of creatures that once lived, is the only way to learn about fossil species. Teeth create this opportunity, because they preserve well in most soils and thus form an essential part of every paleontological analysis. Not only the number of teeth, the dental formula and the tooth shape shed light on diet; the specific pattern of abrasion allows to draw an exact picture of food composition. Besides the number of teeth, tooth shape, and dental formula to shed light on diet, specific abrasion patterns are also vital evidence of food composition.

With the technological advancements of present-day science, it is possible to perform high-resolution scans of teeth and jaws and generate 3D-models from the data. Such models allow for the calculation of varying abrasion grades without the necessity of human contact and prevent damaging of the original. Furthermore, even the smallest indentation on the tiniest tooth can be easily examined through the enlargement of the 3D-model.

Before the development of the *Occlusal Fingerprint Analyzer* (OFA), it was impossible to analyze teeth and check for possible movement patterns at the same time as no previous software offered all the necessary tools. The OFA is not only applicable in paleontological research; even dentists and dental technicians can use the OFA for reconstructing tooth surfaces. The software's open source license permits users to improve the program, specializing it for various specific needs and purposes.

2. System Requirements

In order to run the *Occlusal Fingerprint Analyzer*[1], we recommend the following requirements:

Minimum Requirements:

- CPU: Dual Core Intel/AMD 32/64Bit, 4200+/2200MHz
- 2GB RAM
- Graphics: nVidia GeForce 6600GT or AMD/ATI Radeon x800
- 1GB free disk space
- Monitor Resolution of 1280x1024
- Operating System: 32Bit Windows XP SP2 or 32Bit GNU/Linux 2.6 with X

Recommended:

- CPU: Quad Core Intel: Core i5/i7, AMD Phenom II X4
- 4GB+ RAM
- Monitor Resolution of 1920 x 1080 or higher
- Graphics: Card with 1GB+ graphics memory (necessary for loading huge models)
- Operating System: 64Bit GNU/Linux 3.x with X, e.g. Ubuntu 12.04.4 LTS

[1] In the following as OFA.

3. Installation

This chapter will give a short introduction on how to install OFA onto the supported operating systems (OS). There are 32 and 64-Bit variants of the software. It is not possible to install a 64-Bit OFA on a 32-Bit operating system.[1]

Due to the individual memory management of the OS, there are differences in software performance and the loading of large models. These differences do not effect the outcome of calculations.

The above mentioned concerns mainly 64-Bit operating systems with a large amount of RAM (>4GB). In the test phase, a 64-Bit GNU/Linux system[2] has proven to be best when working with models exceeding the supported 2 million vertices.

It is very likely that the OFA software will work on Windows versions other than Windows XP with SP2 but other versions are not supported and have not been tested. Thus we can not guarantee proper performance with other versions.

3.1. Windows

Download the newest version of the OFA software and execute the setup. Please follow the instructions on the welcome screen and click next. Then agree to the *GNU General Public License*[3].

Choose which components should get installed. The core module and the scene viewer will always get installed, because the program can not function without these. It is recommended to install the preset components.

Finally, choose an installation path for the OFA.

3.2. GNU/Linux

This version comes in a pre-compiled package for Ubuntu 9.10 [4]. Download it and it will unpack and install itself after double clicking on the item.

[1] The 32 and 64-Bit represent the address range of memory the OS is capable of addressing which basically is the main difference between the two (the program benefits of more memory).

[2] e.g. download it for free at `http://www.ubuntu.com`, `http://www.gentoo.org`

[3] See `http://www.fsf.org/licensing/licenses/gpl.html`

[4] Note that this package can get installed on every fork of Ubuntu (e.g. Kubuntu, Xubuntu, etc.).

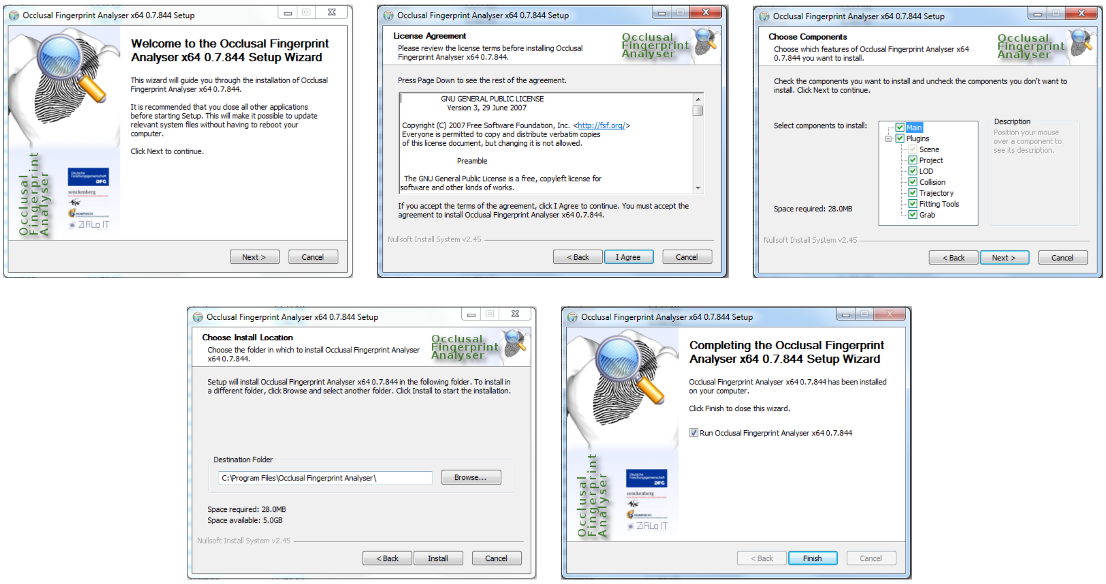


Figure 3.1.: Windows Installation

If another distribution is used, it is necessary to download the source and compile the program. Read the instructions within the source folder for more information.

OFA can get installed on a Ubuntu by executing the following command on a console:
`sudo dpkg -i name_of_the_ofa_package.deb`

The graphic package installer will automatically be executed when trying to open the downloaded package (e.g. double click on it).

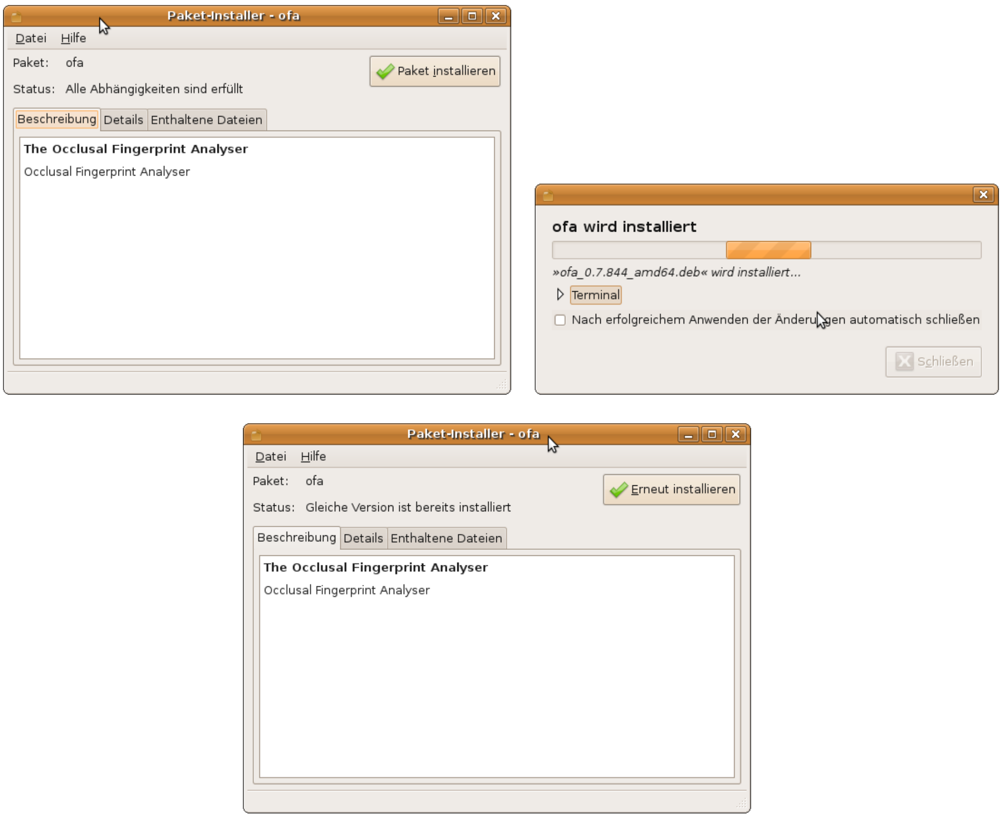


Figure 3.2.: Ubuntu Linux Installation

Part II.

Theoretical Background

Two key features are essential for understanding *Part III* (III, page 17) of this manual and using the program to its full potential. The first feature is the occurrence of collisions and how they are handled. The second is the ability to send a model on a trajectory that will most likely interfere with the position of another model.

4. Collision

A collision occurs when solid objects hit. This can also be applied to the contact of teeth in the chewing process. The detection of these collisions had to be implemented by the developers. A number of algorithms were tested for best performance while achieving the same result. The *Octreesphere Kdtree Threaded Pool List* showed the best results, calculating complex collisions most rapidly.

4.1. Octreesphere KdTree Threaded Pool List

This algorithm organizes the data of a model in a so-called octree. This means the model gets divided into eight identical zones. If one of them contains no vertices it will get removed in an optimization process.

Depending on the depth level, each zone is split into eight subzones repetitively (see figure 4.1. It has the depth level 1). The depth level is not configurable in OFA; but is preset based on the programmer's experiences. Also, the *octreespheres* are not visible in the costumer version of the program, because they proved to be a heavy drain on resources even for powerful computers.

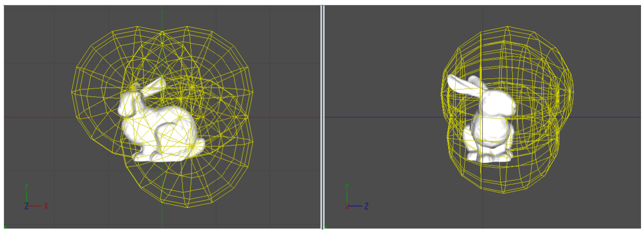


Figure 4.1.: View on the Stanford Bunny from two different directions with spheres of a Octreesphere visible only in the debug version of OFA

OFA preselects the zones in which a collision has happened, removing most unnecessary information from the collision calculation (see figure 4.2). Also all vertices of each model is stored in a seperate kdtree.

The octree and kdtree algorithm divides the data due to their position in space.
The algorithm checks each vertex, if it has come within a predefined distance (see 9.1.1, page 38) of another vertex. If that is the case, the vertex is deemed to be of a collision and is stored in a list. Afterwards collision groups can be established.

Therefore the octree is used to find roughly all colliding vertices, wheras the kdtree is used to find the specific colliding verticies in the given range. When dealing with a smaller amount of data, the kdtree only version is superior to the octree because the octree creation and support time is not necessary. Both algorithms distributes the data to a number of threads[1] which

[1]Wikipedia: `http://en.wikipedia.org/wiki/Thread_(computer_science)`. The number is

improves computing speed rapidly.

4.2. Collision Group

When a collision occurs, all faces involved are noted. Because the data is raw, it needs to be transformed into usable information. The neighboring faces are merged into collision groups. Then it is possible to calculate the size of these groups, perform various other operations, and - if activated - their predecessors from a previous collision can also be calculated (see chapter 17, page 61 and 10.4.1, page 45 for further information).

4.3. Collision Step

A collision step[2] is a collision which has its calculated results stored in the projects memory. The results contain:

- a step comment
- a list of groups
- information on which model a group belongs to
- a groups name
- the number of faces a group possesses
- the predecessor of a group
- a comment per group

These results can get viewed within the *collision viewer* (see chapter 17, page 61) and exported to file. They will also be integrated in a saved project.

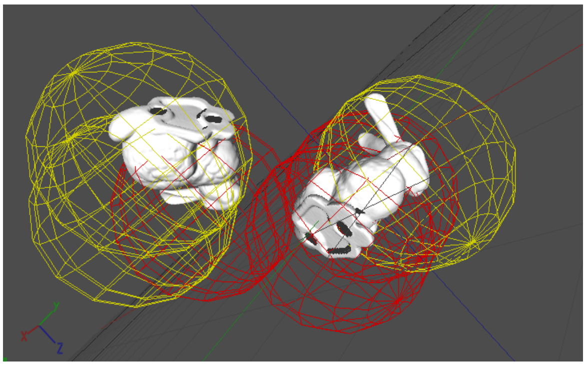


Figure 4.2.: Two Stanford Bunnies brought close to a collision rendering the spheres red, which are in closest proximity with a chance of their vertices colliding

preconfigured based on the number of cpu-cores the computer at hand posses, but can be selected, see 10.1.5, page 43.

[2]Collision step and time step are used as synonyms.

5. Trajectory

Because the chewing cycle does not only consist of one movement with a resulting collision, but rather of a series of correlated movements, this had to be included into the program. To reproduce this criterion, it is possible, with OFA, to define a movement pattern or upload previously mapped trajectory information, gained with a temporomandibular joint recorder.

There are two trajectory algorithms to choose from: the simple and advanced mode. In simple mode, points can be added to the scene and the algorithm will try to follow with a model or a group of models those chain of points. The advanced mode on the other hand is simulating a temporomandibular joint through a triangle. Based on this triangle there can be a path of triangles instead of points.
The main difference between those two choices is that in simple mode a model or group of models will be moved from one point to another point through translations. In advanced mode the transformation between triangles will be archieved trough translations and rotations. So the result can be totally different, assumed the triangles used have been rotated.

5.1. Simple mode

The trajectory setup for simple mode is contained within the *collision path window* (see 16.1, page 57). It lists all path points the trajectory consists of in the respective order that the model will move along them.
The step distance indicates how detailed the movement from one point to the next is; the smaller the distance, the more details are available. Mind that with a small step distance, more time is needed to calculate collisions, as there will be a collision detection for each step along the path.

If the trajectory is manually set, it is very likely that there will not only be collisions, but also deflections and other situations where there is no way to reach a point. In order to counter such issues, algorithms have been placed within the program to prevent aborting.

- **Approximation** Get as close as possible to another model before trying to go around it.
- **Deflection** Try to move around a obstacle with as little deflection as possible.
- **Skip Path Point** Skip a point if it can't be reached.
- **Break Free** Try to free the model when simple deflection is not enough.

  For further information also see 9.2, page 38.

5.2. Advanced mode

The trajectory setup for advanced mode is very similar to simple mode (see 5.1, page 14) and is contained within the *advanced collision path window* (see 16.2, page 58). It lists all created triangles instead of points and will also show them in the respective order the model will try to move along. For further information also see 9.2, page 38.

Part III.

Graphical User Interface

6. Overview

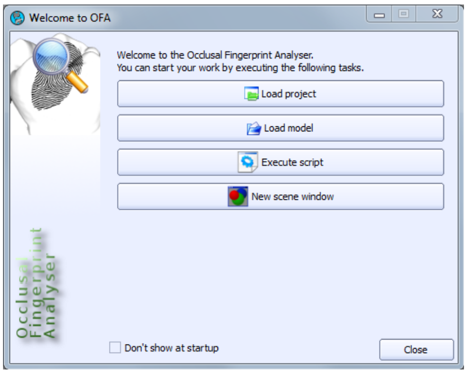


Figure 6.1.: Welcome-screen at startup

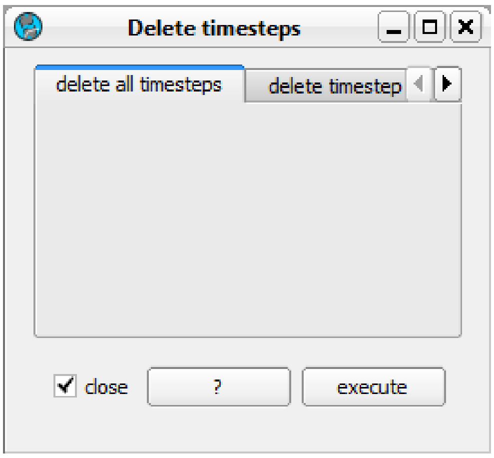
After opening the program, the welcome window will appear. It can be activated or deactivated under Tools > Global (see 10.1.4, page 43).

Due to the architecture of the software, the GUI is variable. At each startup, it is dynamically built and can differ from the one shown in Figure 6.3. Furthermore, every window is movable via drag and drop.

Parameter demanding windows

Each time specific parameters are necessary before command execution, a new window will open demanding further input (see 6.2(a)). There are three options on the bottom of the window:

- **Close**: Decides whether the window is being closed after triggering the command.
- **?**: Opens the help window (see figure 6.2(b)).
- **Execute**: Runs the command.



(a) Example of a parameter demanding window.

(b) The help window for the deletetimesteps command.

Figure 6.2.: Parameter demanding window and its help window

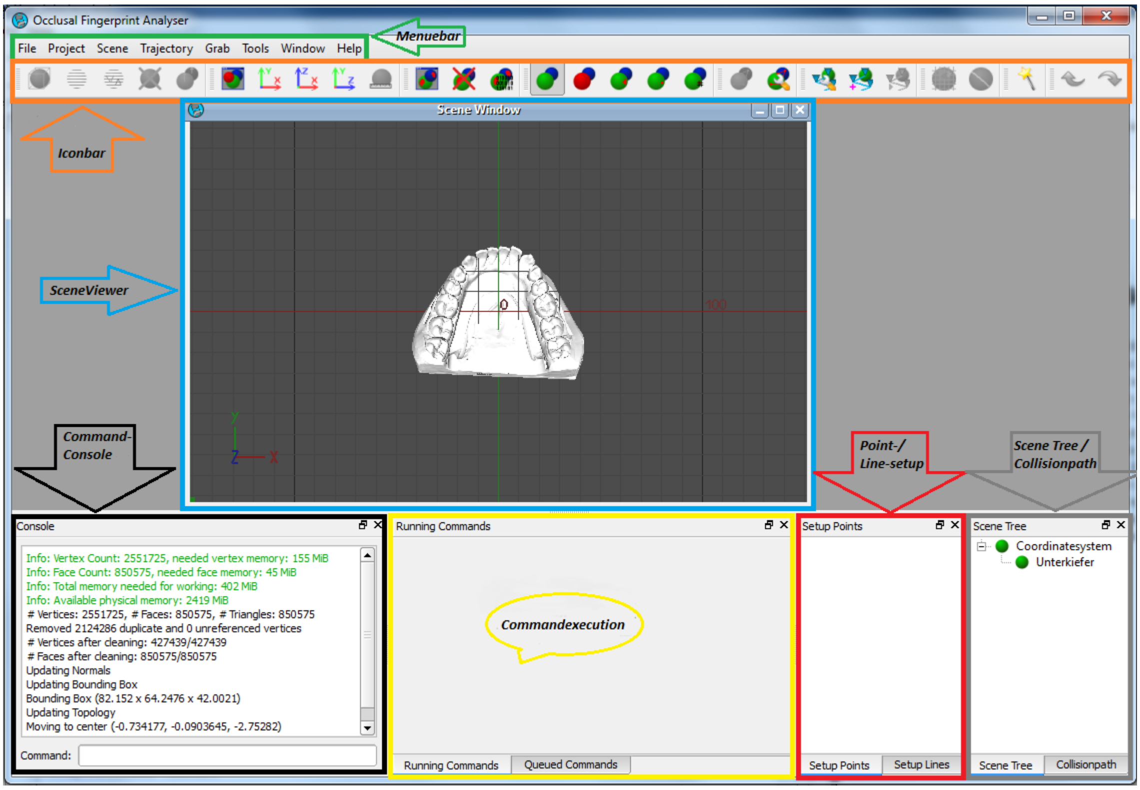


Figure 6.3.: GUI of OFA after loading a model and activating the line-, point- and collisionpath-view

Coloration as follows:

- *GREEN:* The Menuebar (chapter 7, page 20)
- *ORANGE:* The Iconbar (chapter 8, page 32)
- *LIGHT BLUE:* The SceneViewer (chapter 11, page 49)
- *BLACK:* The CommandConsole (chapter 12, page 50)
- *YELLOW:* The CommandExecution (chapter 13, page 51)
- *RED:* The Point-/LightSetup (chapter 14, page 52)
- *GREY:* The SceneTree and Collisionpath (chapter 15, page 53 and chapter 16, page 57)

7. The Menuebar

The following subsection will explain the entries of the menuebar[1].

7.1. File

7.1.1. Load Model *Ctrl + L*

One model or multiple models can be loaded into OFA. In the case of multiple, they will load parallel to one another.

Currently OFA is able to load models coded as:

- Alias Wavefront Objects (*.obj)
- Stanford Triangle Format (*.ply)
- Standard Triangulation Language (*.stl)

Furthermore models can be loaded with the “Load in-place” function at the bottom of the load dialog. With this options the models will use the positions given in the file, in case they have been pre-aligned in other programs. Otherwise the models will be alligned at the z-axis, beginning at the center.

7.1.2. Save Model *Ctrl + S*

Any selected models in the scene tree (see 15, page 53) can be saved as:

- Alias Wavefront Objects (*.obj)
- Stanford Triangle Format (*.ply)

Be aware that multiple selections of models will be saved as one model. Also any color adjustments will be saved. In the save dialog is the option to save the chosen models, the collision of the current collision step (see 4.3, page 13) or both. If collision alone is chosen, only the collided faces will be saved.

7.1.3. Exit *Ctrl + Q*

Closes the program. A security question asks the user if they wish to save the current project before exiting the OFA completely.

[1] Short cuts are printed cursive. If there is more that one shortcut, it means all of them have to be put in to execute the command. The first “+” operator visualises the combining of characters to press. Edged parentheses describes 1:n options to press.

7.2. Project

This item comes with the module project.

7.2.1. New Project *Ctrl + Shift + N*

If a project is running while this command is given, OFA asks whether that project should be saved before initializing a new project. The opening of a new project will delete all unsaved materials from the previous project.

7.2.2. Load Project *Ctrl + Shift + L*

Opens a file dialog in order to load a project.
Warning: Only one project may be loaded at a time!

7.2.3. Save Project *Ctrl + Shift + S*

Saves the current project. If saved for the first time, a project name and the directory in which it should be stored need to be entered.

7.2.4. Save Project as

Opens a file dialog allowing the user to choose where and under which name the project should get saved. The saving process itself will take some time depending on the compression rate of the file (and of course the power of the computer). Find out more about the compression rate at 10.2.2, page 44.

By saving a project, everything within a scene as well as the project setting are stored in the saved file. This includes the models loaded at the time, ensuring that there are no inconsistencies because of different model versions.

7.2.5. View Settings

Opens the Project settings (see 10.2.2, page 44).

7.2.6. Last loaded projects *Ctrl + Shift + [1-8]*

OFA stores the path of the last eight loaded projects and makes them available for fast reload without using any dialog. The last project will always be available with the number 1.

7.3. Scene

In this section of the menu you will find the tools that help shape the scene. Note that these tools do not effect collision nor trajectory calculations.

7.3.1. Add light source *L*

Within a scene, there is a constant light source that can neither be moved nor deleted. But the addition and individual configuration of up to seven light sources is possible. They are marked as green cones in the scene, like in 7.1.

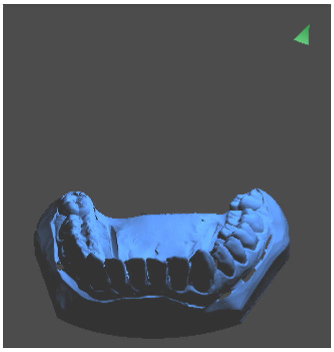


Figure 7.1.: Scene element enlightened with a blue shining light source

To enter the individual configurations, double click on the desired light source in the scene tree (see Figure 7.2).

Light sources have two configurable attributes: the color of light as well as the spread angle which determines the size of the spot at a constant distance to the model.[2]

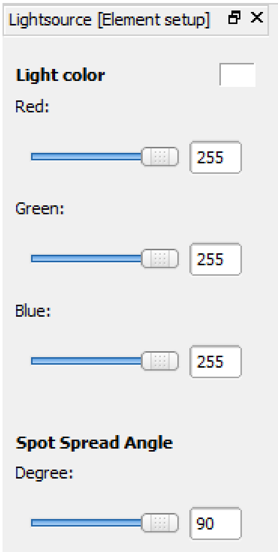
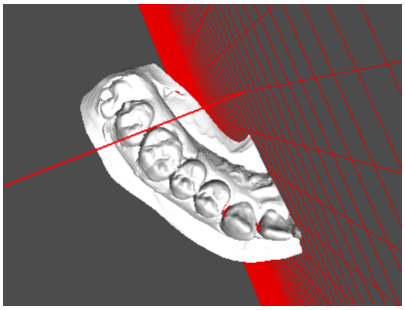


Figure 7.2.: Light Source configuration window

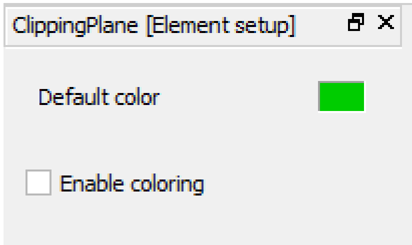
[2]Note that pure OpenGL lights are used, so there will be no shadowing or hiding. This means that light effects are calculated for every triangle that is within the spread angle, regardless of whether there is another triangle between the current triangle and the light source.

7.3.2. Add clipping plane *C*

A clipping plane hides everything that lies behind it (see Figure 7.3(a)). What is hidden by a clipping plane however, is still included in the collision detection and the trajectory calculation.



(a) Clipping plane example



(b) Clipping plane configuration window

Figure 7.3.: Clipping Plane

7.3.3. Add point *P*

Add point allows to add a new point to a selected model. The point will be created at the global x,y,z position chosen in the add point window.

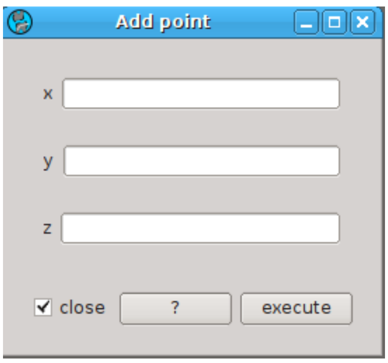


Figure 7.4.: Add point window

7.3.4. Add reference plane *R*

The reference plane is used to reference normal vectors in space against it (see 18, page 64). The first parameter in the "Add reference plane" window is the model or group, the reference plane should be attached to. If attached to a group it will be a reference to all models in the group. Otherwise it will only be a reference to one model. The next three parameters define the point of origin for the plane. The last three parameters define the normal vector of the plane.

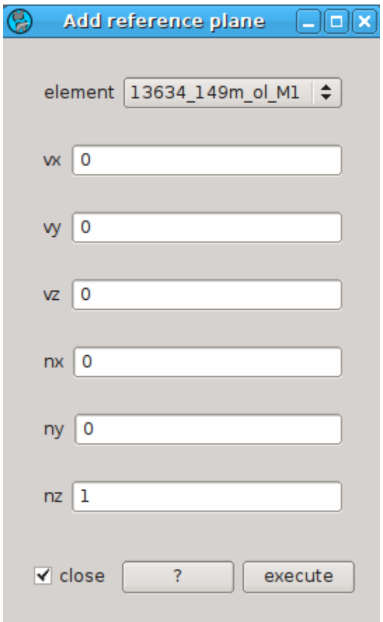


Figure 7.5.: Add reference plane window window

7.3.5. Add selection sphere *S*

With this option a sphere object will be added to the scenetree (see 15, page 53). It has two special options in the context menu of the scenetree: select and unselect faces. Dependend on the option, all faces in it's volume will be selected or unselected.

7.3.6. Add selection plane *CTRL + N, CTRL + P*

A selection plane can be added to select faces from any existing models. Also it can be used to unselect faces. The selection plane can be added to a group of models to limit the selection or unselection only to group members.

7.3.7. Add indentation plane *CTRL + I, CTRL + P*

The indentation plane can be added to the scene to limit the grid used for indentation (see 20, page 78).

7.3.8. New Scene Window *V*

This feature adds an additional scene viewer window to the program. Each additional window allows the user to view the scene from a different perspective. The number of scene windows that can be used is only limited by the space in the central area of the program. However, each window requires additional computer resources (cpu, memory, and graphic power).

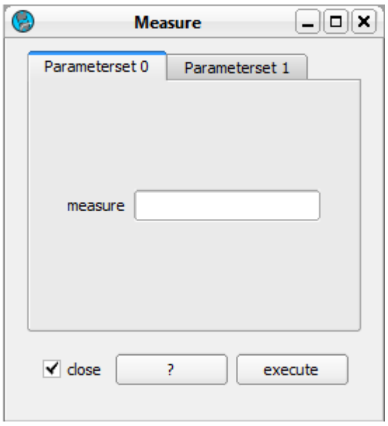
7.3.9. Show selected *F*

This option will zoom on a selected item in the scenetree (see 15, page 53).

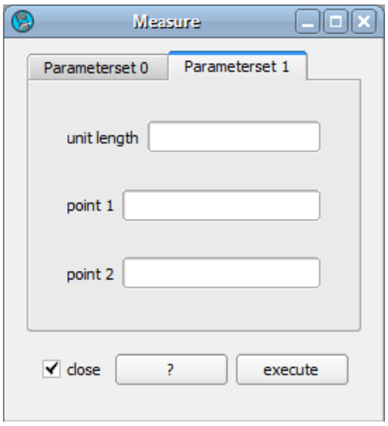
7.3.10. Measure *M*

Before the distances and sizes of areas can get measured within a scene, it is imperative to have a defined base unit within the measuring system. OFA supports both the metric, which is preset, and imperial measure system. This can be changed under *view options*.
In the measure menu, there are two ways to define the base unit:

- Parameterset 0: If the vertex coordinates of the loaded models are scaled correctly or if the scaling factor is known, the value can be applied to the base unit.
- Parameterset 1: Furthermore, the base unit can be defined as the distance between two points that can be placed manually through the set point option (see 14.1, page 52).



(a) Parameterset 0



(b) Parameterset 1

Figure 7.6.: Measure Parametersets

7.4. Diagram

Through the diagram features in OFA, the created collision data can be statistically evaluated.

7.4.1. New

The new option will open the diagram creation dialog, see chapter 19, page 66 for more Information.

7.4.2. Document

This feature creates a new document in edit mode. For more informations on documents, see chapter 19.3, page 72.

7.4.3. Documents

The documents options will open a window with all available documents (see 19.3.3, page 77).

7.4.4. Report

With the report option, a quick report of the current accumulated data will be created (see 19.3.2, page 76).

7.5. Trajectory

In OFA, a trajectory is a path defined by *collision path points*. One model follows that path in a predefined step length. Due to obstacles, it leaves the direct line between two collision points and attempts to locate an alternative route to reach its destination. Each step is memorized by the software. To find out more about the trajectory go to 16, page 57.

7.5.1. Import calculated points

The path points can be saved in a *.txt file (see Figure 7.7). In order for OFA to be able to import the data, there must be only one vertex per row and the coordinates must be separated by a space[3].
When loading a calculated path, it is possible to start the collision detection for each step right away by activating the check box within the window.

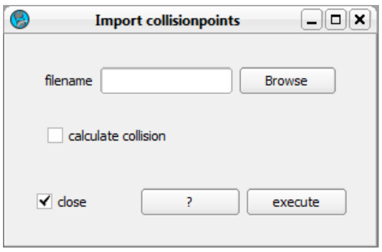


Figure 7.7.: The calculated points import window

7.5.2. Export calculated points

Export Calculated Points exports the calculated path points of a processed collision path to a file. Each line of that file contains a vertex with its x/y/z-coordinates separated by a space. Additionally, some comments about the current scene are included in the header.

[3]Empty lines and lines starting with # are ignored.

7.6. Grab

The grab module makes it easy to obtain illustrations for documentation and publications. With Grab you can take snapshots and make videos of a scene.

7.6.1. Image

With the image tool, a picture of a single scene window can be taken (see figure 7.8). If there is more than one scene window, OFA takes the snapshot from the most recently active window. It is not possible to make a multi-window picture.

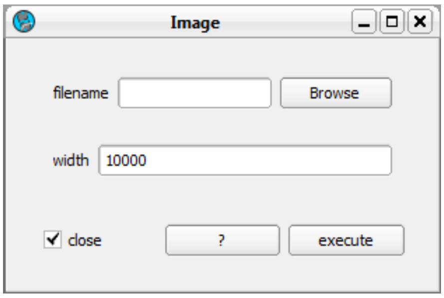


Figure 7.8.: The image grab window

Pictures can be one of the following types:

- Portable Network Graphics(*.png)
- JPEG (*.jpg)
- Tagged Image File Format (*.tif)

Besides the file format, it is also possible to choose the width of an image. The **height** of the image is generated from the aspect ratio of the source scene window.
The restricting factor in the images size is the computer's memory. OFA itself has no limitations. OpenGL, on which the software is based, allows only a very small texture resolution[4]. Nevertheless, simply through partitioning the scene and rendering one piece at a time, it is possible to make pictures with a larger resolution.

Mind that one pixel requires 4 bytes of memory, and a picture with a resolution of 10.000x8.000 will require ~312 MB RAM. With 20.000x16.000 ~1.2 GB RAM is required.

[4]Wikipedia: `http://en.wikipedia.org/wiki/Pixel`

7.6.2. Video

As shown in figure 7.9, it is also possible to grab a video of the most recently selected scene window. If '0' is chosen, the size of the last active scene window will reflect the width and height of the video. Perspective changes will be recorded, too.
If *sync* is enabled, the model used for trajectory will be moved to each calculated collision point. So the *sync* option is comparable to the play command (see 32, page 35).

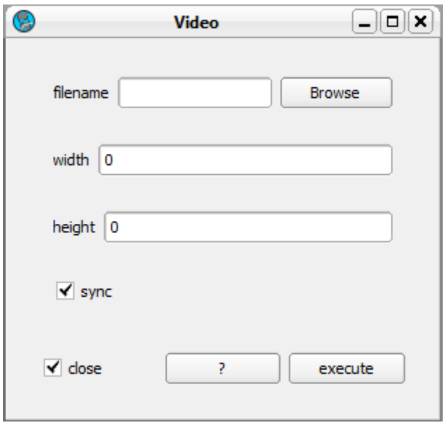


Figure 7.9.: The video grab window

OFA supports several video formats. The following list represents the available codecs on an Microsoft Windows platform.

- MPEG4 v2 (*.avi)
- Flash Video (*.flv)[5]
- Xiph Theora (*.ogv)
- Windows Media Video 8 (*.wmv)

With *GNU/Linux*, there are even more codecs available.

7.7. Tools

7.7.1. List Commands

GUI, data and logic are separated in OFA. This means that any button hit, model movement or other input will activate the corresponding piece of code within the inner part of the program. That is why an item in the menu bar, an entry in the icon bar, or a command in the console are able to start the same operation. These code pieces are called *commands*. Each command is unique in its naming and can get started through the console.

When using the *list commands* button, all commands available in OFA are shown in the console (see 12.1, page 50).

7.7.2. Execute Script

It is possible to write scripts containing a list of commands which will execute them in order. The list can either be handwritten or created with the macro recorder OFA offers (further information in 7.7.4, page 29).

7.7.3. Save Console

Saves the output from the console to a user-defined text-file (*.txt).

7.7.4. Macro Recorder

As mentioned before, this software separates actions from the GUI parts. This offers the opportunity to keep a list of the actions that are performed. If the same version of the program is used, if the models initially involved are accessible, and if the computer has the same capabilities as the one which recorded the macro list, it can be replayed.

This allows for the reproduction of all steps and makes exchanging them with fellow researchers possible. Of course, one can simply save a project but these files have a certain size, especially when dealing with huge models. The recorded macro on the other hand just has a marginal size. This allows for fast distribution of a project.

The recorder can also be used to set OFA to match personal preferences like opening views or insert light sources. To best use this feature, OFA can be started with a parameter, specifying the script to run at startup, for example by running it via console:
`%ofa_dir%\ofa_x64 --script %scriptdir%\script.txt`
The macro recorder allows 5 actions:

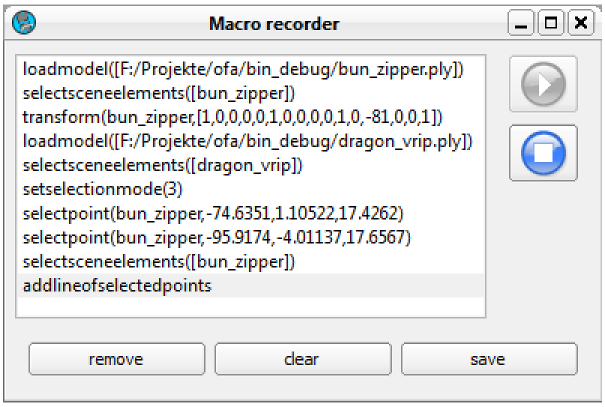


Figure 7.10.: A macro recorder window with a list of recorded commands

- *Start Recording* As the name indicates, this starts recording all commands relevant for the project.
  Figure 7.10 shows an example of a recording. It contains a scene where a model is loaded, transformed to another position within the scene and a second model is loaded. Next is the change of the selection mode to point selection. Two points on the first model are chosen and a line is made from them.
- *Stop Recording* Stops a running recording.
- *Remove* Delete a previously marked command from the list.
  Warning: This feature is possible at any time. However, deleting commands in the middle of the list can have negative effects on the following commands. It is recommended not to remove commands in a session. Removed commands are unrecoverable.
- *Clear* Removes all commands in the list. Again this is unrecoverable!
- *Save* Save the list to a *.txt file.

Tip: If you wish to continue a recorded macro from another session, start the macro recorder before executing the script. All those commands will then be added to the list.

7.7.5. Report Wizard

The report wizard can be used to create a template for any report (see 19.3.2, page 76).

7.7.6. Options

The options menu allows to set global progam options (see 10, page 42).

7.8. Window

Lists all dockable windows of the GUI. They can either be shown or hidden.

There is also the simple control widget, for transforming models or groups of models. In the translate part of the widget, a selected model can be translated in direction of the colored vector:

- *RED:* X-axis
- *GREEN:* Y-axis
- *BLUE:* Z-axis

The "+" or "-" button will increase or decrease the position along the given direction with the entered value. The lock button will synchronize all translation values to the same value.
In the roation section, a rotation axis has to be selected, colored the same as translation. The entered value will then be used to rotate about the selected axis. All used transformations are relative to the actual position. For absolute transformations, the absolute transformation section can be used. It is similar to relative transformation sections. The transformation will be executed if the "Translate" or "Rotate" has been used. The last section gives information about the current center position and used matrix.

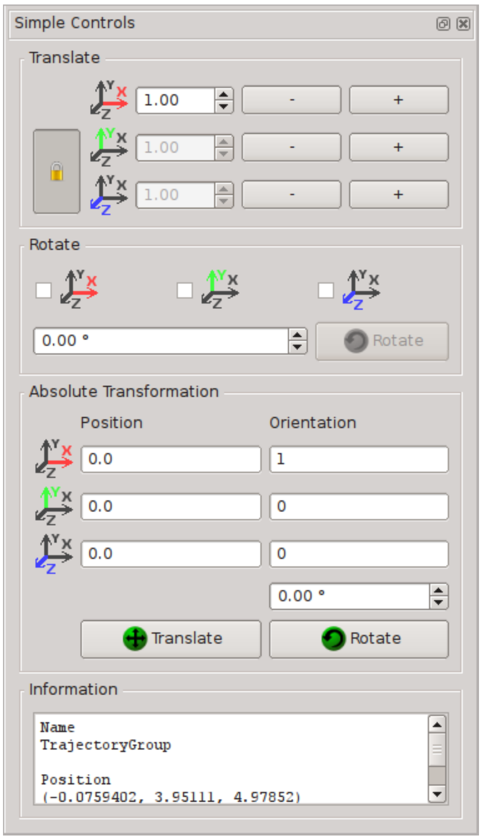


Figure 7.11.: Simple control widget

7.9. Help

Contains all kind of information about the program.

7.9.1. Prepare email

Prepares an email with all informations about the used OFA Version, the operating system in use, the available memory and much more Informations relavant for the developers of OFA in case of problems with OFA.

7.9.2. About

7.9.2.1. About

Contains the version number, the build number, the used version of Qt, and the companies and people involved in the development of the software (see figure 7.12).

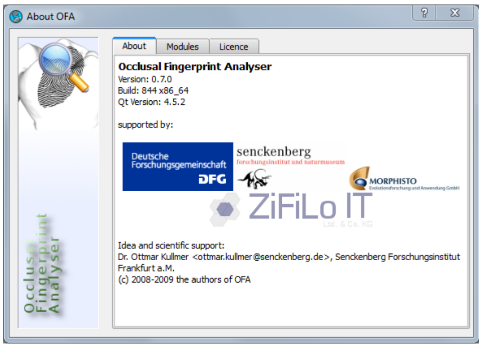


Figure 7.12.: About Occlusal Fingerprint Analyzer

7.9.2.2. Modules

As shown in figure 7.13, page 31, this menu lists each module and its sub-modules in a tree. For each module, there is a short summary of what it does, its authors and additional information (e.g. the modules licence). So one can retrace who wrote which module, which is quite important when multiple sources develop modules for OFA.

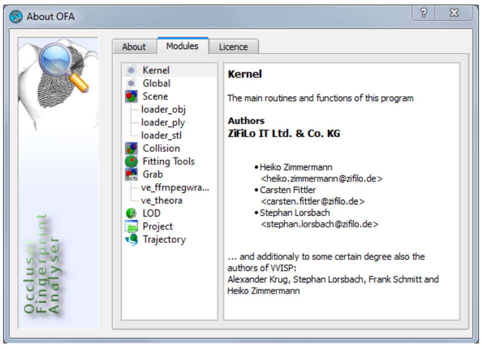


Figure 7.13.: The modules that are currently loaded in OFA

7.9.2.3. License

This tab contains the GPLv3 license OFA is under (see figure 7.14).

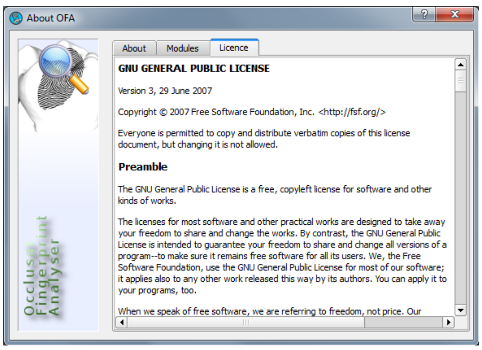


Figure 7.14.: OFAs used license

8. The Toolbar/IconBar

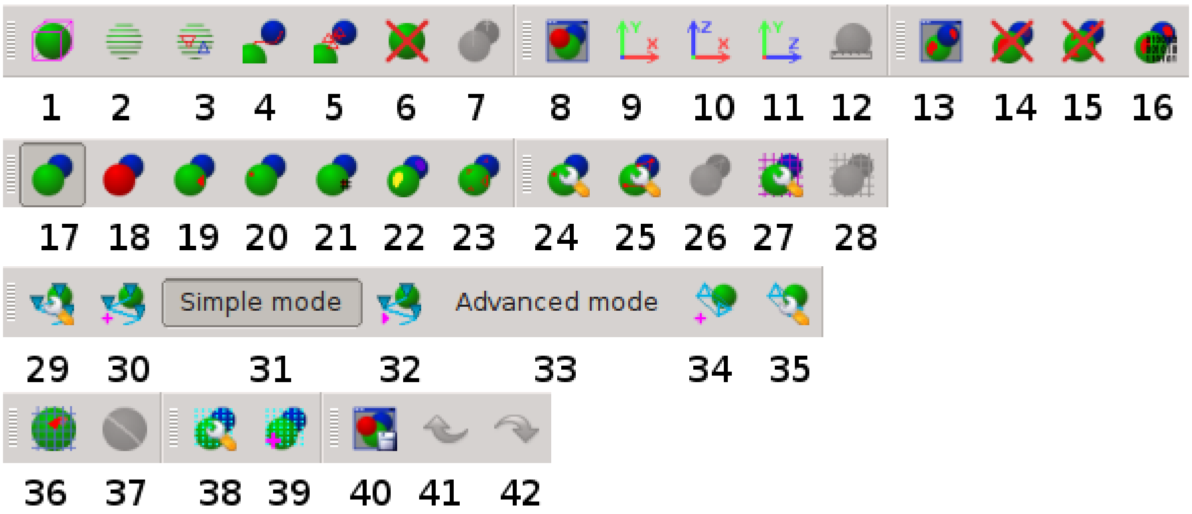


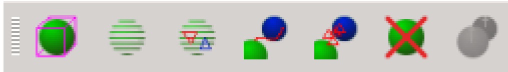
Figure 8.1.: Numbered iconbar divided in four lines

In this chapter, all icons currently existing are explained. For better understanding, the iconbar is separated into groups:

- (01-07) Scene element operations (8.1, page 33)
- (08-12) Scene viewer operations (8.2, page 33)
- (13-16) Collision viewer (8.3, page 33)
- (17-23) Picking operations (8.4, page 34)
- (24-28) Line operations (8.5, page 34)
- (29-35) Trajectory operations (8.6, page 35)
- (36-37) Fitting operations (8.7, page 35)
- (38-39) Indentation operations (7.7.4, page 29)
- (40-42) Journal operations (8.9, page 36)

The individual iconbars can be moved around in the OFA window or outside of the window like any other windows in OFA. Furthermore they can be hidden by right clicking the iconbar and selecting or deselecting the specific iconbar in the lower part.

8.1. Scene element operations



1 2 3 4 5 6 7

1. **Show/Hide Bounding Box** *(B)* Shows/Hides a models bounding box. *Available: Selection of a model or group of models.*

2. **Hide Scene Element** *(H)* Hides the previously selected scene element and renders it invisible within the scene, yet it will still be part of collision detection. *Available: Selection of any element in the scenetree.*

3. **Hide Collision** *(Ctrl+C, Ctrl+H)* Hides the collision groups of the previous selected models. *Available: Selection of a model.*

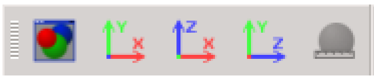
4. **Hide trajectory path** *(Ctrl+P, Ctrl+H)* Hides the simple collision path. *Available: always.*

5. **Hide advanced trajectory path** Hides the advanced collision path. *Available: always.*

6. **Delete active element** *(Del)* Deletes the previously selected scene element **permanently**. *Available: Selection of a scene element.*

7. **Aligns a model to another model** The program will align these models through a best fit algorithm. *Available: Two selected models or groups, three points on each selected model or group.*

8.2. Scene viewer operations



8 9 10 11 12

8. **New Scene Window** *(V)* Opens a new window showing the scene (from a possibly different perspective). This scene window is independent from the first. *Available: always.*

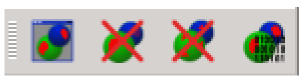
9. **View x-y-Plane** Changes the camera angle of the last active scene window to the x-y-perspective. *Available: At least one loaded model.*

10. **View x-z-Plane** Changes the camera angle of the last active scene window to the x-z-perspective. *Available: At least one loaded model.*

11. **View y-z-Plane** Changes the camera angle of the last active scene window to the y-z-perspective. *Available: At least one loaded model.*

12. **Measure** Opens a window to declare the base unit of this scene by defining the length between two selected points (see 7.6(b), page 25). *Available: Two points created.*

8.3. Collision viewer operations



13 14 15 16

13. **Collision Viewer** *(Ctrl+C, Ctrl+V)* Opens the *collision viewer window* (see 17, page 61). *Available: always.*
14. **Delete Timesteps** In this menu, it is possible to delete all collision recorded so far or to delete a range of steps (see figure 8.2). *Available: always.*
15. **Delete selected faces from timesteps** Is similar to delete timesteps, but deletes selected faces from all or the selected range of timesteps. *Available: always.*
16. **Activate Collision Calculation** *(Ctrl+C, Ctrl+A)* If this is activated, results of collisions occurring after transforming a model manually will be stored as a collision step. *Available: always.*

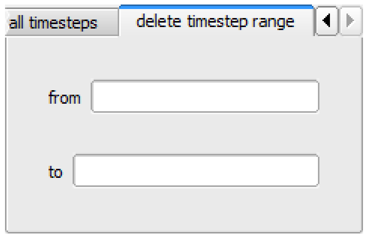


Figure 8.2.: Tab to delete a range of collision steps

8.4. Picking operations

17. **Select None** *(Ctrl+1)* When this mode is active, simple clicks will not have any effect within the scene. *Available: always.*
18. **Select Model** *(Ctrl+2)* Marks a scene element active in the scene tree through clicking on it within the scene. *Available: always.*
19. **Select Face** *(Ctrl+3)* Enables the selection or deselection of multiple faces from models within the scene. *Available: always.*
20. **Select Point** *(Ctrl+4)* Enables the creation points on a model (the size of the selected point can be adjusted 10.3.3, page 44). *Available: always.*
21. **Select nearest Vertex** *(Ctrl+5)* Enables the creation of a point on the position of the nearest vertex from the point clicked upon. *Available: always.*
22. **Select collision** *(Ctrl+6)* Selects and higlights the clicked collision group in the collision viewer (see 17, page 61). *Available: always.*
23. **Clear Face Selection** *(Ctrl+0)* Deselects all selected faces. *Available: always.*

8.5. Line operations

24. **Setup Points** *(Ctrl+G, Ctrl+P)* Opens the points window (see 14.1, page 52) embedded in the lower right section of the program. *Available: always.*
25. **Setup Lines** *(Ctrl+G, Ctrl+L)* Opens the line window (see 14.2, page 52) embedded in the lower right section of the program. *Available: always.*

26. **Add Line** *(Ctrl+N, Ctrl+A)* Connects **all** points created so far through a joint line. **This operation is irrevocable and all points are lost in the process!** *Available: Two points available.*

27. **Angles** *(Ctrl+N, Ctrl+A)* Opens the plane orientation window (see 18, page 64) embedded in the lower right section of the program. *Available: always.*

28. **Attach reference plane** *(Ctrl+R)* Attaches a reference plane to the current selected model. *Available: Three points on the selected model and no points elsewhere.*

8.6. Trajectory operations

29. **Setup collisionpath** *(Ctrl+P, Ctrl+S)* Opens the collision path setup (see 16, page 57) embedded in the lower right section of the program. *Available: always.*

30. **Add point to path** *(Ctrl+P, Ctrl+A)* Adds a new path point within the scene at the position (origin) of the current active scene element. *Available: always.*

31. **Simple mode** Switches to simple trajectory mode (see 16.1, page 57). *Available: always.*

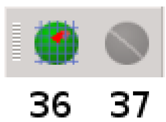
32. **Play trajectory path** *(Ctrl+P, Ctrl+P)* Replays the calculated trajectory with all its collisions, deflections, etc. within the scene. *Available: a calculated path in either simple or advanced mode and the proper selected mode.*

33. **Advanced mode** Switches to advanced trajectory mode (see 16.2, page 58). *Available: always.*

34. **Add triangle** *(Ctrl+P, Ctrl+T)* Adds a triangle within the scene at the position (origin) of the current active scene element or the last added triangle. *Available: Three points available overall, on one model or a group of models, an existing path triangle (which will be copied) available, or a timestep of the advanced path selected that has no proper motion.*

35. **Setup advanced collisionpath** *(Ctrl+P, Ctrl+C)* Opens the advanced collision path setup embedded in the lower right section of the program. *Available: always.*

8.7. Fitting operations



36. **Fit to Plane** Creates a plane from the normals of selected faces. The faces will get sketched onto the plane. *Available: always.*

37. **Fit to Line** Creates two lines (if possible) on the surface of a model (see figure 8.3, page 36). *Available: Two points overall on one model.*
    A plane using the two points as rotation axis is shown. A window opens up allowing for the rotation of the plane. Once execution is selected, an algorithm tries to create the lines where the plane intersects the model, going both clockwise and counterclockwise from point one to point two. The two lines will be added to the scene tree, giving the user the option to delete one and keep the other, if desired.

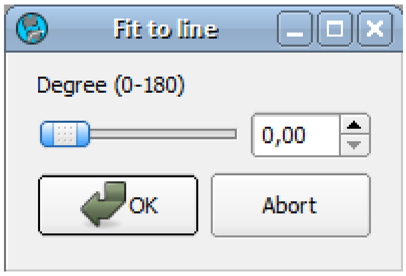


Figure 8.3.: The window to rotate the plane indicating where the lines will get drawn.

If in one direction, there is no possible connection between the two points, the algorithm stops and displays a line up until the point where calculation was aborted.

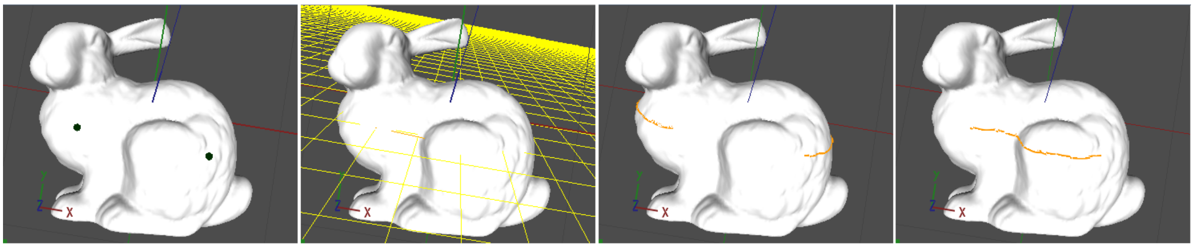
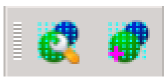


Figure 8.4.: The steps for line fitting: Starting with selecting two points in the left picture, next choosing where the lines should be drawn and, in the two pictures to the right, the two resulting lines.

8.8. Indentation operations

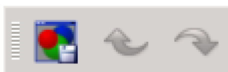


38 39

38. **Gridjobs** Opens the gridjobs window (see 20.3, page 81) embedded in the lower right section of the program. *Available: always.*

39. **New grid job** Opens the indentation setup (see 20.1, page 78). *Available: always.*

8.9. Journal operations



40 41 42

40. **Show savepoints** Opens the savepoint menu. *Available: always.*

    The savepoints window at the right side of OFA, allows to save the different current states of OFA, to reset to at a another time. OFA will automatically create save points prior to important calculations which will impact the current state of OFA, like the calculation of a trajectory path (see 16, page 57) or indentation (see 20, page 78).

    For each savepoint, different states can be stored and restored. To store a new point, the "New" Button can be used. All checked states in the red colored area will be stored. The automatically created save points will have all states available to restore. For a restore of a savepoint, a double click on an entry name in the yellow colored area will restore all states represented in the activated icons before the entry name. Also a selected entry can be restored with the "Restore" button. The "Delete" button will delete all states of a selected entry independent from the icon selection. The blue colored area shows all available icons.

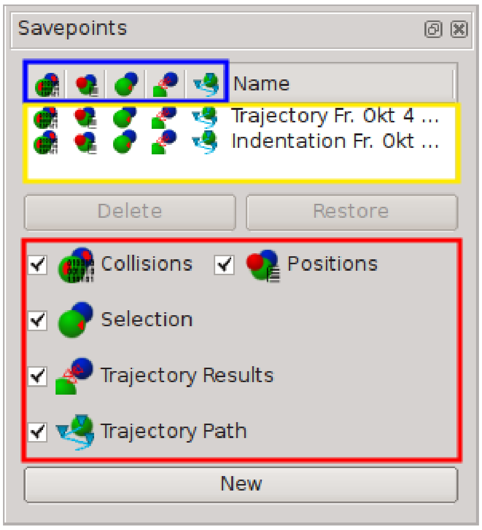
    

    Figure 8.5.: Document view mode

    There are the following states available:

Type	Explanation
Collisions	All available timesteps and collision groups per timestep.
Positions	The position of each scene element.
Selection	Any available face selection.
Trajectory Results	Any calculated path results for simple or advanced mode (see 16.3, page 59).
Trajectory Path	Any self created path points for simple or advanced mode (see 16.3, page 59).

41. **Undo** *(Ctrl+Z)* Reverts a transformation (rotation or translation) executed on a element within the scene. *Available: if any of the described operations have been done.*

42. **Redo** *(Ctrl+Shift+Z)* Restores a previously undone transformation. *Available: if an operation has been undone.*

9. Project Settings

The Project settings will be stored within the OFA project file. All relevant data for the calculation of a collision path are in the project settings to make a project interchangeable.

9.1. Collision Tab

9.1.1. Handler setup

On the handler setup group, there are two options. The first states which collision-detection algorithm is being used. A number of algorithms were tested; the *Octreesphere KdTree Threaded Pool List Global* proved to be best. It is preset within the software and is highly recommended for use.

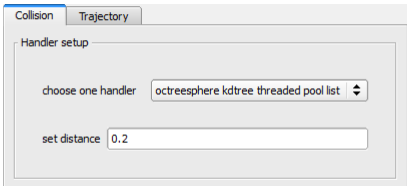


Figure 9.1.: The collision tab of the project settings

Set Distance is the range in which opposing faces are considered to be colliding. This value depends strongly on the models, as they may be scaled differently based on their origin. To find a good value, it is best to open the model options (double click on the model in the scene tree, see 15.4, page 55) and check the average edge length. The set distance should be a bit higher or at least equal to the average edge length to prevent the models from accidentally moving into each other: When the set distance is significantly smaller than the edge length, the algorithm is unable to detect the vertices of a nearby model.

9.2. Trajectory-Tab

The following explains the settings necessary for the trajectory calculation.

9.2.1. Approximation

When a model moves along a path, the distance between the individual steps is predefined. Of course, while detecting tooth contact in a chewing cycle, one model will at some point block the set path of the other model. This means the distance between the two models will lie between the step distance and zero.

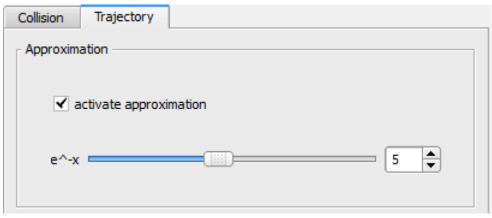


Figure 9.2.: The approximation part of the Trajectory Tab

If the approximation is activated, the program does not stop there, but will decrease the step distance to get as close as possible to the other model. If the step gets too big, it will automatically decrease again.

Step distance diminishment frequency can be defined in the approximation menu; it corresponds to the number represented by the slider.

9.2.2. Deflection

These settings influence how a model's movement continues once it collides with another model. With each step, a calculation on how the direction of the model will be deflected from all the collided faces and their normal vectors is launched.

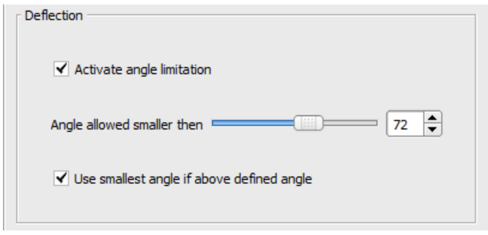


Figure 9.3.: The deflection part of the Trajectory Tab

The settings account for the angle formed by the direction and the normal vectors of the faces. Only if the first check box is activated, will the value given from the slider be used. The slider defines the maximum allowed angle.

If the maximum for this angle is exceeded, OFA will choose the smallest collision group angle even though it is above the angle limitation. To allow this function, activate the second check box.

9.2.3. Skip collision path points

If deflection alone is not enough to pass by an obstacle, activation of the skipping of path points will allow the model to move further in its collision path. The model then can move to the next path point and calculation continues. If skipping is not activated, the collision path calculation will stop.

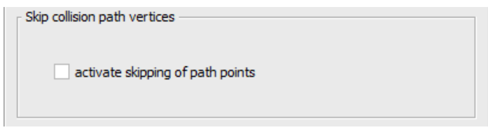


Figure 9.4.: The skip collision path vertices part of the Trajectory Tab

9.2.4. Break Free

Another possibility to improve the continuing calculation of collision paths is the *break free* function. If activated, the model will attempt to free itself by modifying the calculated angle. As soon as an angle is found before the maximum angle is exceeded, the model will move further along its path.

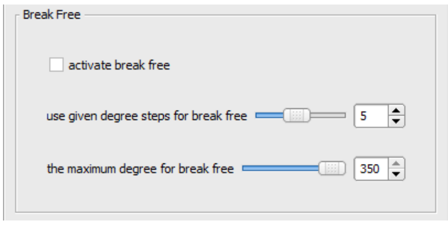


Figure 9.5.: The break free part of the Trajectory Tab

The first slider defines the degrees added to the angle at each step. The second one defines when to stop trying to break free.

9.2.5. Show result triangles

In case of a small distance used for advanced trajectory there can be shown many result triangles from the calculated path in the scene. With this option the number of shown result triangles can be manipulated. The given number will be used as a divider. For example, the default value of 2 will show every second triangle, meaning 50% of the result triangles.

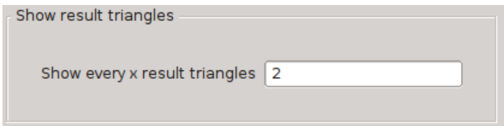


Figure 9.6.: The show result triangles part of the Trajectory Tab

9.2.6. Proper motion

In case models of a group should be available to use proper motion (see 16.4, page 60) the first check box has to be checked. Be advised that all other proper motion options will only be used if this option is checked. Also proper motion does only work with groups of models (see 15.0.3, page 56).

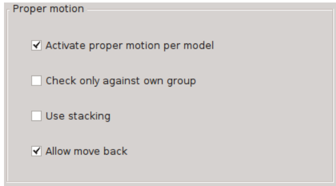


Figure 9.7.: The proper motion part of the Trajectory Tab

In theory, proper motion should move a model away from any models of an opposing group it could collide with. As fewer models involved in collision calculation increases performance, the second check box can be checked to benefit from this fact, as only collision checks against the own group will be made. To be absolutly sure that there is no overlapping or if any overlapping has occured, this option should be unchecked.

The third option allows to stack any movement done by proper motion instead of delta calculation between steps, e.g if maximal radius is 0.4 and stepping is 0.2 the model will not move any further after two steps.

As it is only natural, that after moving models away from any opposing models in case of a collision, they should be moved back to their original position if they are free. The last checkbox can be used to achieve this effect.

10. Options

Options are global settings for OFA. The selected values are used for the current and all later projects.

10.1. Global

10.1.1. Working directories

As shown in figure 10.1, it is possible to select the directories in which scripts, models and projects are usually located. New file dialogs will open with these directories at the start. They are also updated by using the associated commands.

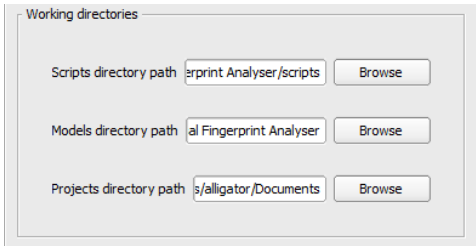


Figure 10.1.: Working directories paths

10.1.2. Undo limit

Translations and rotations performed on anything within a scene through mouse movements, commands or GUI elements are reversible. The user can decide how far back he or she can undo the actions up to a limit of 100 steps.

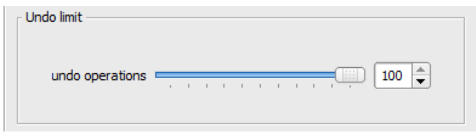


Figure 10.2.: Undo limit options

Warning! The deletion of scene objects is irreversible! Because remembering all those previous actions costs precious memory, this function is adjustable to individual computer power (see Figure 10.2).

10.1.3. Minor security questions

As mentioned before, many actions in OFA, especially deleting objects, are not covered by the undo function.

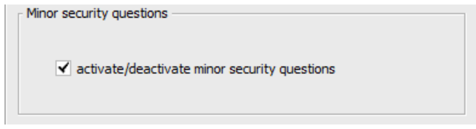


Figure 10.3.: Minor security questions option

To prevent irreversible actions, check the Minor Security Question check box so that security questions will be asked as a warning before irreversible acts are performed (see Figure 10.3).

10.1.4. Welcome GUI

The Welcome GUI will not be shown at program launch when the "Hide Welcome-GUI" check box is activated.

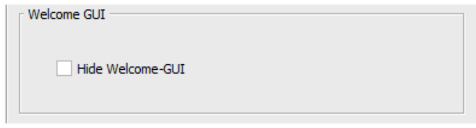


Figure 10.4.: Hide Welcome GUI

10.1.5. Global maximal thread count

In OFA, many features can run parallel, like the calculation of the collision path (see 16, page 57). The amount of parallelism depends on the hardware available

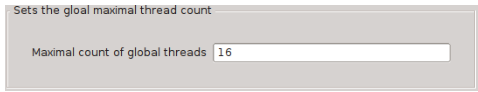


Figure 10.5.: Global maximal thread count

and on the number of global threads allowed. The default (0), will use the processor count multiplied by 2. A higher value might decrease the performance of OFA, as the system could be overloaded. To few threads, might not use the full capacity of the existing hardware. Also few threads is a good if there are other programs open which need processing time.

10.2. Project

10.2.1. Measure system

Defines which measure system is applied in the project to calculate distances and the size of areas. As mentioned before, the metric and imperial system are possible.

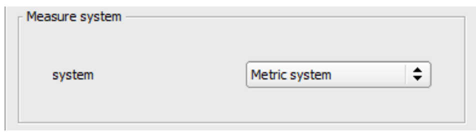


Figure 10.6.: The measure system option

10.2.2. Compression rate

When saving a project, everything the scene contains as well as the project settings are stored in a single file. With large models, this can easily use up a lot of disk space. That is why a packing system was included.

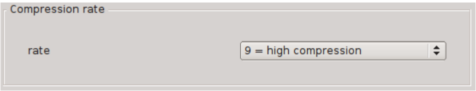


Figure 10.7.: Global tab with compression rate option

The compression rate goes from *0->no compression* to *9->high compression*. Achieving a high compression can take some time depending on cpu power. The compression rate impacts saving a project, but not loading a project.

10.3. Scene

10.3.1. Mouse speed

This menu allows for the configuration of mouse input when transforming objects in a scene window (e.g. rotating a scene). The mouse input can be adjusted to individual preferences (see Figure 10.8).

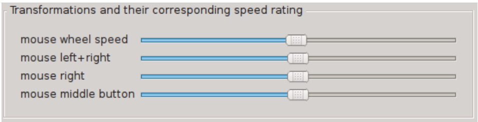


Figure 10.8.: Transformations and their corresponding speed rating

10.3.2. Scene Attributes

Defines a background color for the scene.

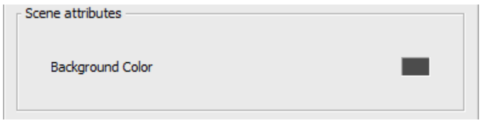


Figure 10.9.: Configurable scene attributes

10.3.3. Selected Points

Defines how big the marker (in this case a sphere) of a selected point is on a model.

Figure 10.10.: The diameter of a selected point marker

10.3.4. Stereo options

All scene windows can be displayed in a configurable stereo mode, for simple stereo glasses. Red/Blue and Red/Cyan are available. The eye disperity can also be configured.

Figure 10.11.: Stereo options

10.3.5. Level of Detail (LoD)

Scenes loaded with complex models may turn out to be difficult to navigate when the graphic card is overstrained while trying to run the models fluently.
TIP: A LoD can also help for grabbing Videos smoothly.

Figure 10.12.: Configurable scene attributes

To overcome this problem, the models can be simplified just for viewing, with no effect on internal calculations. Every model being loaded into the scene after adjusting will have the appointed LoD, if the level of detail is activated (slidebar value under 1000, see Figure 10.12). The LoD can also be set later at any time (see 15.0.1, page 53).

10.4. Collision

10.4.1. Collision groups

For every collision, a *collision step* is generated (read more at 4.3, page 13).

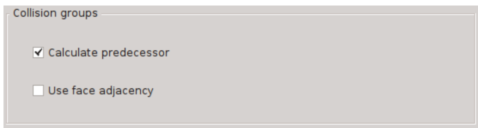


Figure 10.13.: Collision groups configuration

Sometimes, it is important to know the predecessor of a collision group (see figure 10.13), but because it takes time to calculate, the user can choose if the predecessors are to be included in the calculations.
If the *use face adjacency* check box is activated, the raw data of the model is used for the detection of neighboring faces. This will include flawed faces in the raw data though they exist. Otherwise, the neighboring faces will be determined through the faces edges.

10.4.2. Realtime collision test

Whenever a model position has changed, a collision detection occurs. If this check box is activated, the program will check for collisions while moving a model with the mouse through a scene.
If not activated, one model will move through the other. The number of vertices used for the real time collision detection can be reduced, because this will not effect normal collision detection or trajectory path calculations!

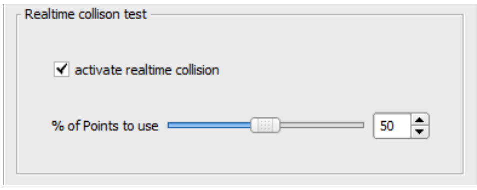


Figure 10.14.: Check for collisions while moving a model

10.5. Trajectory

10.5.1. Automated collision reports

OFA can create an automated collision report based on the pre-configured report template (see 19.3.2, page 76). The report will be shown after each run of a collision path.

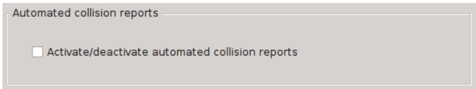


Figure 10.15.: Automated collision reports

10.5.2. System Beep on trajectory finish

After each run of a collision path, a system beep can be generated, to inform that OFA has finished the path calculation. A system beep sound has to be set in the operating system to allow this feature to work.

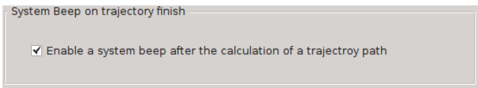


Figure 10.16.: System Beep on trajectory finish

10.6. Grab

10.6.1. Video

This menu contains the video's parameter besides width, height, and format (see 7.9, page 28).
First, it allows for the configuration of how much video is recorded before it is written on the hard drive. Second, there is the choice of how many frames per second the video should have. The third option determines the bit rate. Lastly, the user can select whether the video should be written out at once or if the temp file specified should be used, which is recommended.

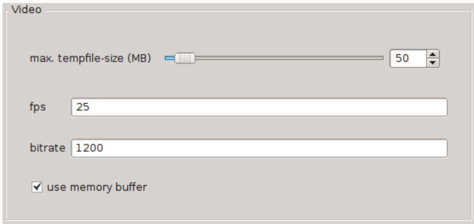


Figure 10.17.: Configure video options

10.7. Indentation

10.7.1. Indentation

To save memory, the refine calculation can be instructed to store only the last result, deleting any intermediate data. To save even more memory the number of results for any indentation or refine job can be limited. Be advised that limiting the numer of results per job, does restrict the update functionallity in case of a new face selection (see 20, page 78).

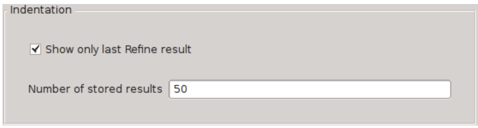


Figure 10.18.: Indentation options

10.7.2. Network

As any indentation calculation might need a huge amount of time, other available computers can help to shorten it (see 20.2, page 81). The "Listen" option makes the current OFA listen to any other OFA who might need help in calculation of a new indentation job. The port can be freely set to any free port. The clients can be specified per name (if the host is well known) or per IP. Also a client can be specified as hostname:port, to use another port then the default port 65432.

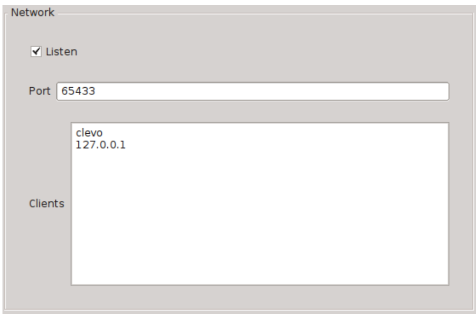


Figure 10.19.: Network options

10.8. Export

10.8.1. Decimal delimiter

The decimal delimiter can be specified for any export of data like in 17.1, page 63.

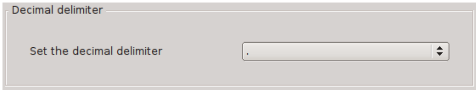


Figure 10.20.: Decimal delimiter options

11. The SceneViewer

The scene viewer is like a camera with the scene in its focus. The perspective can be freely chosen by the user with the following commands:[1]

- **Rotate the scene** by holding the right mouse-button clicked and moving the mouse.
- **Rotate around the view axis** by holding down the shift key and moving the mouse to the right or left.
- **Zoom In/Out** by using the mouse wheel.
- **Fast Zoom In/Out** by holding left+right mouse-button clicked and then moving the mouse.
- **Translation orthogonal to the view-axis** by holding the mouse wheel clicked and moving the mouse

A coordinate system as well as an axis graph were implemented for better orientation. The coordinate system has a grid on the x-y axis. The size of a grid adjusts itself to the size of the models.

The axis graph stays always in the lower left corner of the viewer and on top of the scene. It rotates with the scene.

11.1. Transforming Nodes

There are several possibilities to transform nodes. The node to be transformed must always be marked active within the scene tree! To move a node with the mouse within the scene, it is imperative to keep the *shift*-key pressed while performing the transformations. Once the key is released, background operations like collision detection will start.

Another way to transform a node is via the menu within the scene tree or via command in the console.

[1]Note that mouse movement is configurable under *Scene>Mouse Speed* (see 10.3.1, page 44).

12. The Command Console

12.1. Console

The console prints both commands entered in the command line and feedback from actions currently running.

These actions regarding the console are possible:

- **Save** the output of the console through the menu bar (see 7.7.3, page 29).
- **Clear** the console (removes the output). This option is available when right clicking in the console.
- **Mark a part** with the mouse and copy it.
- **Mark all output** in the console. This option is available when right clicking in the console.

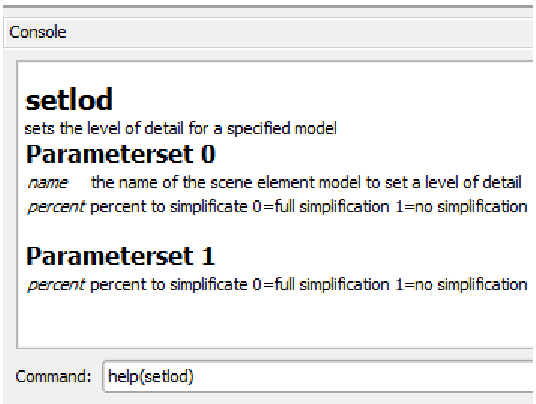


Figure 12.1.: Console with printed out help for setlod

12.2. Commandline

As mentioned before, it is possible to execute anything accessible through the graphical interface with a command entered in the console.

If you cannot remember a command, you can type the first few letters and then press *Ctrl+Space*. A list of commands is offered from which you can choose. If there is only one possible command, it will be completed automatically.

Most commands have at least one parameterset that they expect with the command. If those parameters are unknown, just add parentheses to the command and the command line will print out all possible parametersets. It will then ask for them one at a time.

Alternatively, the command *help()* will print out all parametersets with their available options. *help(setlod)*, for example, will show the help for *setlod*.

13. The Command Execution

13.1. Running Commands

Shows the currently executed command. In order to guarantee data integrity, some can not run parallel.

There are different types of commands:

- **Short Executed Commands:** They will perform a single task that won't take much computing time. (e.g. transform)
- **Long Duration Commands:** A single operation that takes some time. (e.g. collision detection)
- **Delegating Commands:** They operate a task that requires executing other commands (e.g. Trajectory; see Figure 13.1)
- **Loop Commands:** Those do not stop unless manually ordered to. (e.g. video grabbing, trajectory replay). These commands can either be paused / rerun or stopped completely 13.2.

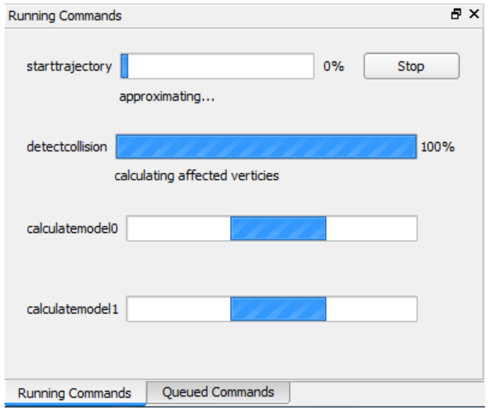


Figure 13.1.: Trajectory calculation with collision detection

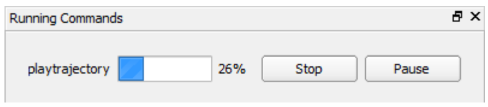


Figure 13.2.: Running video grabbing

13.2. Queued Command

Due to the serialized execution of some commands, a queue exits. Commands in here can get cancelled before they are executed.

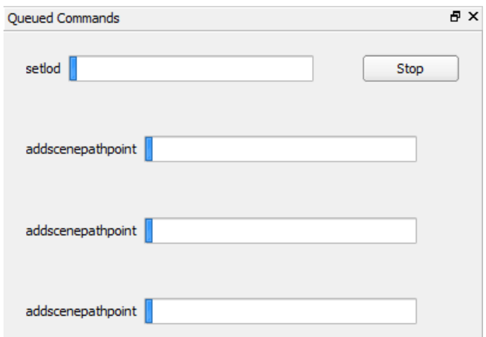


Figure 13.3.: Screenshot of a filled queue

14. The Point-/Light Setup

14.1. The Pointsetup

The point setup contains a list of all points created in the scene that have not been used (for lines etc.) before. The name of each point is generated from the model it was created on and its coordinates. It is not possible to rename points. They cannot be hidden; even when the corresponding model is hidden, the points are still visible.

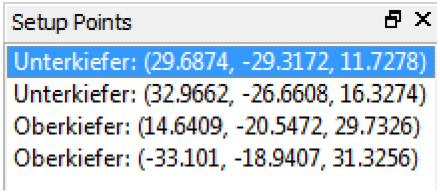


Figure 14.1.: Pointsetup with a list of points

If a model is deleted, all points related to this model are removed as well. Points can be discarded by marking and selecting the delete option by right clicking.

14.2. The Linesetup

In OFA, a line can either be a simple connection between two previously created points or it can be a row of joint lines. Each line within the list can be renamed by marking it and pressing *F2*. Also, line properties are changeable by double clicking on an entry.

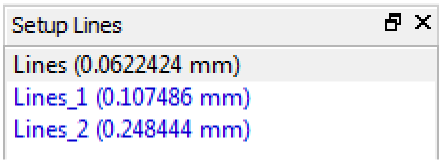


Figure 14.2.: Linesetup with a list of lines with their names and length

The length of a line is shown in the parentheses behind the name. Note that the measurement must be set correctly to an accurate value!

15. The Scene Tree

The scene tree gives an overview of all elements the user can manipulate in a project or scene. Points and lines appear in separate windows. Child nodes are effected by their parents features.

Two of the most frequent elements which effect others are light sources and clipping planes. Only if a model is a child of these, will the features be applied to the model (see Figure 15.2 and Figure 15.1). There is no limit in child nodes.

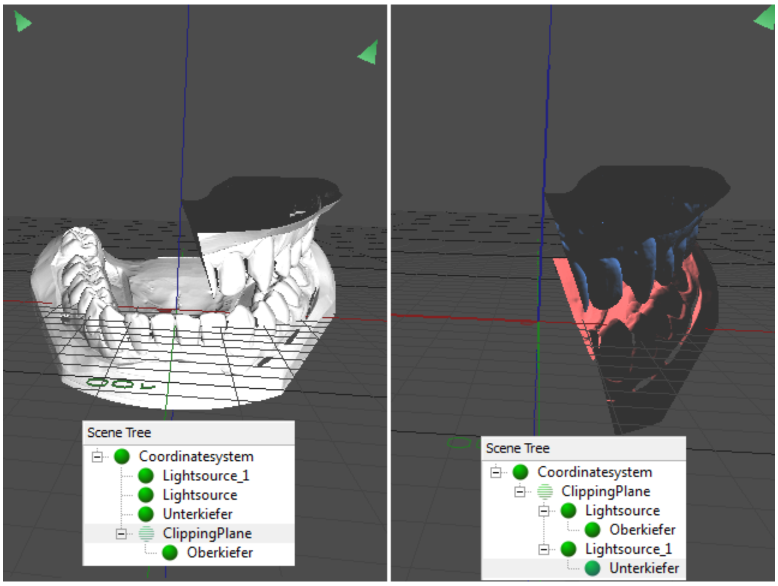


Figure 15.1.: Two examples with an additional hidden clipping plane

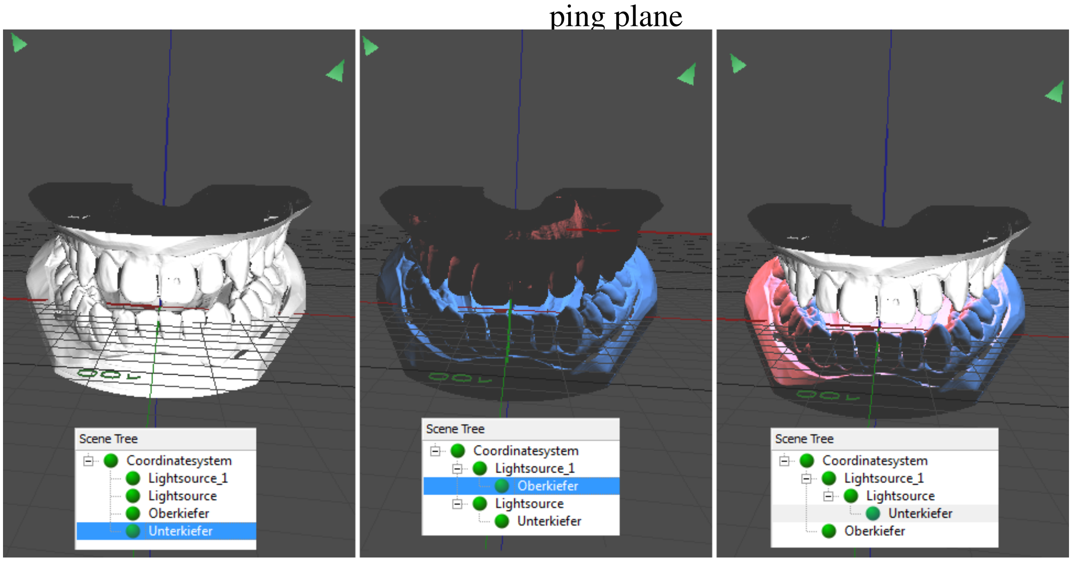


Figure 15.2.: Three tree examples with Lightsource (upper right, blue light) and Lightsource1 (upper left, red light)

15.0.1. Options per Element

Quick access to a number of operations is given by right clicking on a scene element (see figure 15.3, page 54).

- **Hide/Show Scene Element:** Decides whether the element is being drawn in the scene. The light disappears when hiding its source.
- **Hide/Show Collision:** Decides whether the collision groups of a model are shown in the scene. This option applies to models only.
- **Show/Hide Bounding Box:** Hides the element and shows its bounding box instead. Again this only applies to models.
- **Delete active element:** Removes the element **permanently** from the scene. This action cannot be undone!

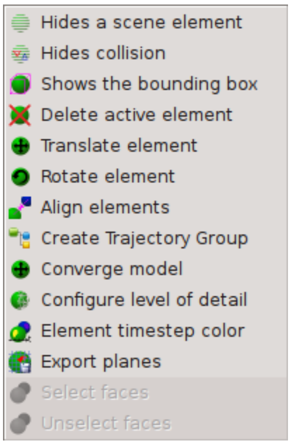


Figure 15.3.: The menu available for a scene element when right clicking it

- **Translate Element:** Translates an element. Enter the vector the model should be translated with. It can either be relative or absolute[1].
- **Rotate element:** Rotates an element. Enter a vector with the rotation angle either in degree or radiant. It can be either relative or absolute.
- **Align elements:** Moves the first selected element of multiple elements to the center and aligns all other with their pre-aligned (from the model file) coordinates, relative to the first selected (related to load-inplace, see also 7.1.1, page 20).
- **Create trajectory group:** Creates a new trajectory group with the selected elements.
- **Converge model:** Converges the model to an opposing model (see 20.5, page 83).
- **Configure Level of Detail:** Allows the user to choose the level of detail for the model active (see 10.3.5, page 45).
- **Element Timestep Color:** Allows the user to override the color settings per collision group and to assign each group in a time step the identical color. To do so, activate the check box in the new window. In the widget, there is a slider to choose a time step and a color chooser to select a color for the currently selected time step. This option applies only to the currently selected model.
- **Export Planes:** Allows the user to export fitted planes. They will be written out as a *.txt file with the origin vector first, then the normal of the plane and finally its name.
- **Select faces:** Selects all faces a selection element describes (selection sphere and plane).
- **Unselect faces:** Unselects all faces a selection element describes (selection sphere and plane).

[1]Relative= add vector to current position, Absolute= Translation to given points independent from current position.

Besides the operations mentioned above, there is also the possibility to rename a scene element. This action is accessible through marking the element and pressing *F2*.

15.0.2. Element Properties

Most of the scene elements have further accessible properties. They can be opened by double clicking on the element in the scene tree. For non-model elements, this often includes determining the color of the object or the color of its effects (e.g. lightning, see figures 7.3(b), page 23 and 7.2, page 22). In some cases, like with a fitted line, features like line size can be edited.

15.0.2.1. Model Properties

Because of the numerous amounts of information and options of the properties a model offers, they are explained below.

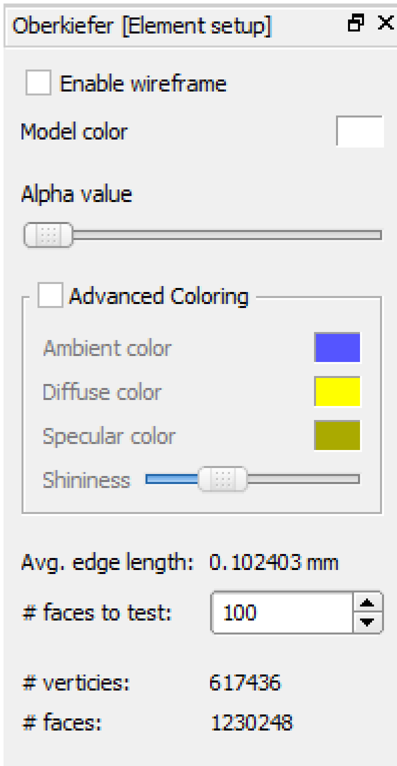


Figure 15.4.: Model Properties

- **Enable Wireframe:** Forces the model to be drawn without filling the faces, so only the face edges are visible. If a model has a very high resolution, it is necessary to zoom in to note the difference.
- **Model Color:** Defines the color of a model.
- **Alpha value:** Decides how transparent a model is. When the slider is moved to the left, the model is 100 percent visible. When it is on the far right side it is invisible.
- **Advanced Coloring:** Changes the material properties of the model. This only effects the display.
- **Avg. edge length** Shows the average edge length from the first x faces the model has, where x is the number entered in the *# faces to test* field. This value is important for the collision detection and the set distance (9.1.1, page 38).
  This value, however, is not be representative of the model, because there is no guarantee that the faces are the same size. In general, the higher the number of faces used for the calculation, the more precise the result. Due to the size of the models, this process requires too much time to calculate the total average edge length, thus the preset 100 faces makes this feasible.
- **# vertices / # faces:** Shows the total number of vertices and faces the model possesses.

Note that the *Model color* and *Alpha value* setup is disabled, when the model is loaded from an *Alias Wavefront Object*-File (*.obj), because these type of models defines there own material setup. But this is how colorized models can be loaded into OFA.

15.0.3. Grouping of models

In some cases there is not one model for multiple teeths, but one model per teeth available. To group those teeth, the right click menu on a model or a selection of model can be used (see 15.0.1, page 53). If a group is selected and any transform operation executed, it will be performed on the group and all of it's members. Also any path calculation will applied to any existing groups. Furthermore there are trajectory group settings for proper motion, see 16.4, page 60.

16. The Collision Path

In this chapter, a trajectory can be set up and the actual model movement along that path configured and calculated.

Movement from point to point is split in steps. The distance of each step can be defined in simple and advanced mode (see figure 16.1 and 16.3) in the field to the left of the *Start* button. The smaller the value, the more steps the path, resulting in a higher resolution of the path. Of course it also increases calculation time.

Once the calculation is finished, the slider enables the user to move the model along the trajectory path, showing the position it held at a certain time and the resulting collisions. If the check box is activated, the collision viewer shows the collisions of the currently selected time step.

Trajectories can be replayed (see 32, page 35).

16.1. Simple mode

For a simple collision path *collision path points* need to be set or imported (see 30, page 35). The list in the collision path widget (see Figure 16.1, page 57) contains all path points. The model chosen in the drop down box moves from path point to path point in the established order by starting trajectory calculation. It is possible to change this order simply with drag and drop.

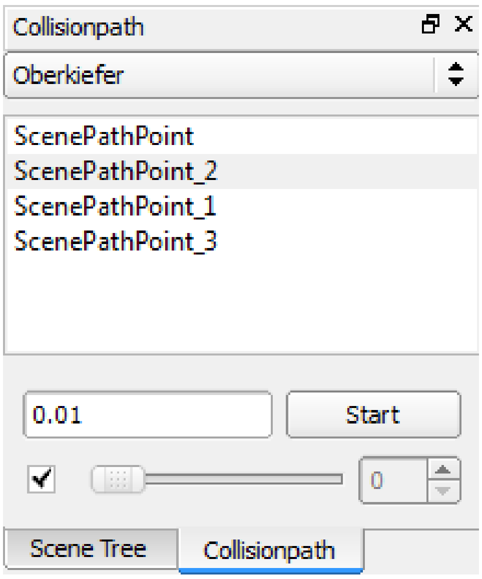


Figure 16.1.: Collision path widget with four path points

16.2. Advanced mode

The advanced mode calculates the path along pre-confiured triangles. The triangles simplify a temporomandibular- joint and axis in movement, from the start to the end. The first triangle is the inital position. It has to be specified by adding three points to a model or to a group of models in a trajectory group like in figure 16.2, where trajectory group (see 15.0.3, page 56) has been build and a triangle added. The first two points will specify the temporomandibular joint axis, represented with a thick cyan line.
The third point is the top of the triangle. The triangle can be added through the add triangle feature of the toolbar (see 34, page 35).

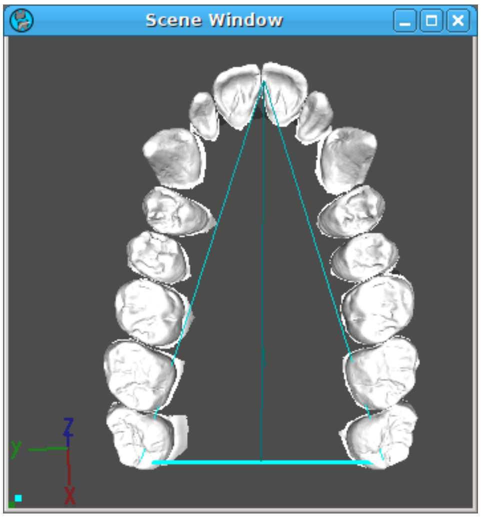


Figure 16.2.: Simplified temporomandibular-joint and axis

Another triangle can be added, by pushing the add triangle button again. After the creation of the second triangle, it can be selected and moved to the proper destination. Like the simple mode, multiple triangles can be added and the path will be along those triangles. The triangles will be attached to the trajectory group or model. In the path calculation, each triangle will be transformed to the next triangle by the shortest path, using translation and rotation if necessary.

The list in the advanced collision path widget contains all path triangles. The moving object is automatically choosen by the added triangle. As in simple mode the model or group of models will move from path triangle to path triangle. Also like in simple mode the order can be changed with drag and drop.
Another difference to simple mode is that before a calculation the first triangle should be double clicked. In this case, the position of the moving object will be resetted to the position it used relative to the triangle at it's creation time. Otherwise the current position of the moving object will be used, which allows to correct the position after a path calculation and before a new calculation.

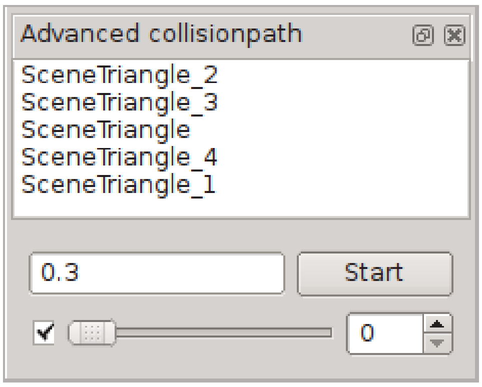


Figure 16.3.: Advanced collision path widget with path triangles

To limit the movement and rotation of the temporomandibular joint, there can be set specific constraints. A double click on the “SceneTrianglePath” Will open the constraint options. First the movement of each point of the axis can be constrained, by applying a radius to each or just one of them. As seen in figure 16.5, the radius can be displayed as a sphere to illustrate the constraint setting. Also color for the spheres can be changed. Furthermore the roation above (in direction of the normal of the triangle, dependen on the points used by its creation) and below the triangle can be constrained.

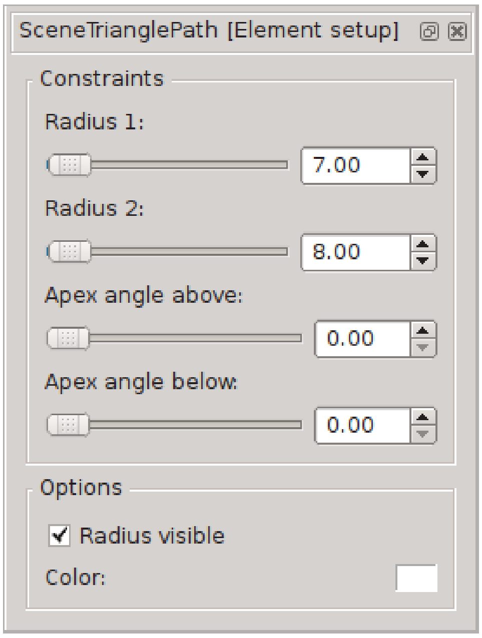


Figure 16.4.: Constraint options

Any created triangles will be checked against the constraints set and will turn to red, as in figure 16.5, if the check fails. The triangle can be moved and will turn back to cyan as soon as they are again inside the allowed constraints.

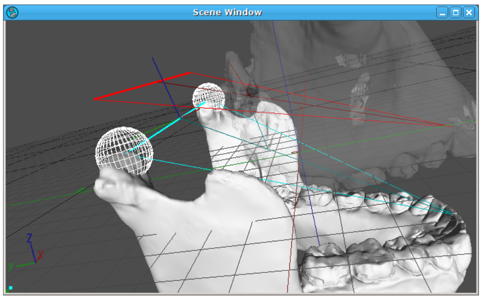


Figure 16.5.: Temporomandibular joint axis constraint

16.3. Results

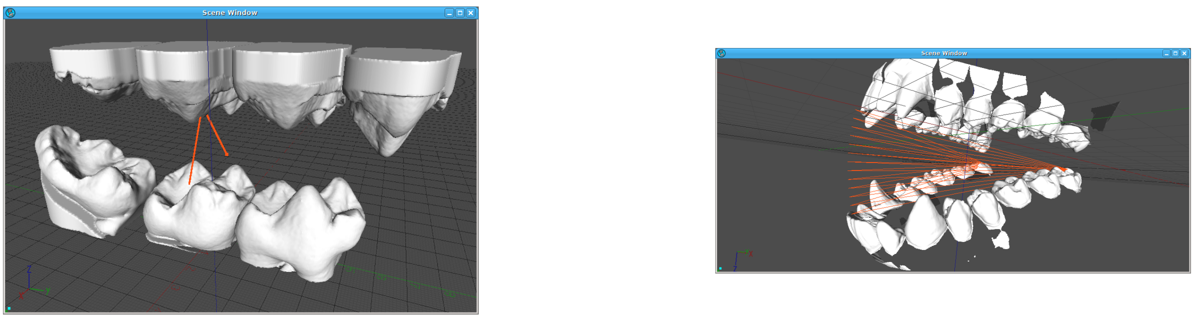


Figure 16.6.: Simple and advanced result path

Dependend on the trajectory mode the resulting path will be displayed in red points or red triangles between and on (if the point/triangle has been reached) the pre-configured collision path. At each displayed result point or triangle, a timestep is available. The shown result triangles can be reduced, see 9.2, page 40. Also for simple mode, the result path can be exported (see 7.5.2, page 26). Resulting path points with proper motion will be displayed in dark red.

16.4. Proper motion Setup

Proper motion allows models of a group to move away from a model it is colliding with, within a specified radius. It will move back as soon as possible. Proper motion is only available for Trajectory groups (see 15.0.3, page 56). With a double click on the trajectory group in the scene tree window, the proper motion properties for the trajectory group can be set.

For each member of the trajectory group a stepping and radius value is available. The radius defines the maximal distance from the original position, relative to the group center. So if the group is moving and there is no proper motion, the full radius is still available. With the stepping option, the distance per proper motion can be set. So if a collision occurs, the colliding model will try to move the defined distance away from the mode it is colliding with, until there is no collision or the accumulated stepping values will be beyond the radius.

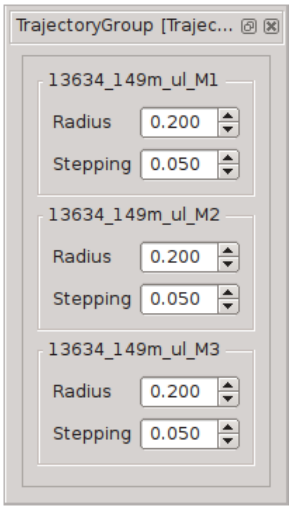


Figure 16.7.: Proper motion settings

17. The Collision Viewer

The collision viewer is a window that contains all information resulting from collision calculations. Each result is displayed as a *time step*.

- **View Collision Groups:** Allows the user to see either all collision groups resulting from a collision or just those of a single model.
- **Collison Group Table** Contains collision groups depending on the view selection.
  - **Model:** The model the collision occurred on.
  - **Name:** The name of the group. It can be changed by double clicking at it or pressing *F2*.
  - **Triangles:** The number of triangles forming this group.
  - **Inverted:** Inverts the normal of the fitting plane for the collision group (see 18, page 64).
  - **Right-Aligned:** Mirrors the projected normal along the referene plane direction.
  - **Color:** The color of the group. It can be changed.
  - **DIP:** The dip of the collision group normal in relation to the reference plane.
  - **DIP-Direction:** The dip-direction of the collision group normal in relation to the reference plane.
- **Area:** The size of all collision groups shown in the table.
- **Selected:** The size of the currently selected groups. Multiple selection is possible.

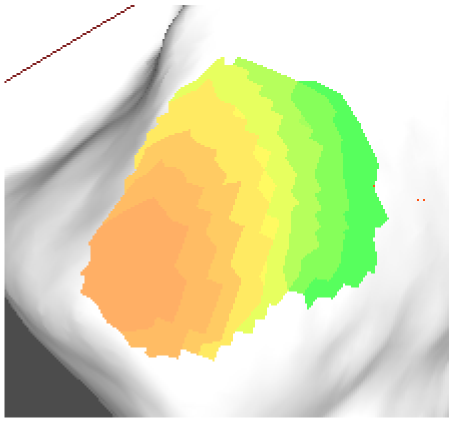


Figure 17.1.: A gradient of the collision groups of the last 24 steps with red being oldest, green newest step

- **Comment:** A comment for the collision step. This comment is not exported!
- **Time Steps:** A slider to choose from all recorded collisions.
- **Gradient:** Allows the user to merge up to a hundred collision groups preceding this one for show. Colors for the oldest and newest group are embedded.
- **Alpha:** Alpha value for the collision groups. When the slider is positioned to the left, this means 'invisible' and positioned to the right, this means 'one hundred percent visible'.
- **Gradient color:** The color for the oldest and the newest collision groups can be chosen. The program will interpolate a color gradient for the groups in between (see figure 17.1, page 61).
- **More:** Expands the window for further Options:
  - **Comment for Group:** Allows the user to write a comment on a collision group. This comment will stay with the group even if it is exported.
  - **Predecessor:** For this function, the *calculate predecessor* option (see 10.4.1, page 45) must be activated before the collision. It shows the respective preceding collision group of this one.

If predecessors are calculated, changes on *Name, Color, Inverted and Right-Aligned* will also be performed on all successors of the current group.

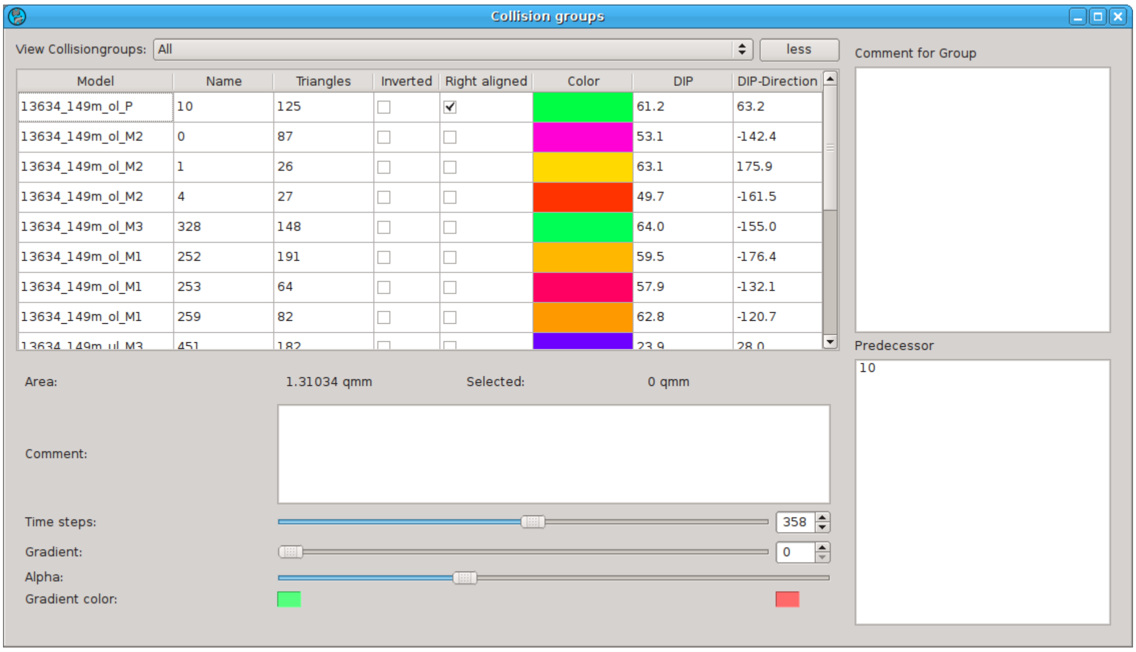


Figure 17.2.: Collision viewer expanded

17.1. Collision Group Menu

Within the *collision groups table*, there is a menu accessible by right clicking (see figure 17.3, page 63).

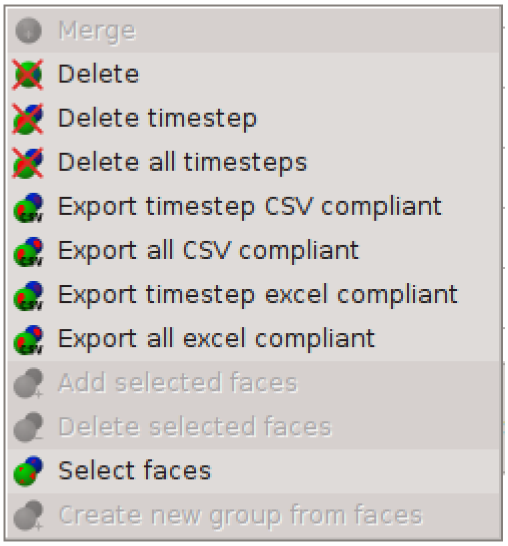


Figure 17.3.: The menu available for collision groups.

- **Merge:** When multiple groups of the same model are selected, they can be merged into one group.
- **Delete:** Deletes the currently selected collision group(s).
- **Delete time step:** Deletes the entire time step.
- **Delete all time steps:** Deletes all time steps contained in the scene viewer. **Warning:** This action is irrevocable.
- **Export time step CSV compliant:** Exports the data of a time step in a CSV[a]-file.
- **Export all CSV compliant:** Exports the data of all time steps in a CSV-file.
- **Export time step excel compliant:** Exports the data of a time step in a Microsoft Excel compliant CSV format.
- **Export all excel compliant**: Exports the data of all time steps in a Microsoft Excel compliant CSV format.
- **Add selected faces:** Adds faces that were manually selected in the scene to the selected group.
- **Delete selected faces:** Deletes faces manually selected from the group.
- **Select faces:** Renders the faces of the marked groups selected in the scene (e.g. to fit a plane from a collision group).
- **Create new group from faces:** Uses the selected faces to create a new group. Any faces still attached to another group will be unattached from that group.

[a] Comma-separated values; especially suitable for moving tables between different programs.

18. Scene reference plane

Some features in OFA need a reference plane to work at all, like the Stereoplott diagram (see 19.1.2, page 67). This plane is used to reference normal vectors in space against it. A reference plane has to be attached to a model. It can work either with:

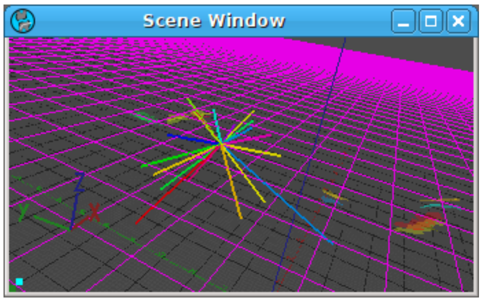


Figure 18.1.: Projected collision group normal vectors

- Collision data of the collision groups (see 4.2, page 13). This will use all faces of a collisiongroup and will reference the average normal vector against the reference plane, with all collision groups.

- Fitting planes. In this case the normal vector of the fitting plane will be referenced against the reference plane.

If there is any collision data, the current time step (see 4.3, page 13) with all available collision group data will be displayed (see 18.2) in relation to the scene reference plane if the Plane orientation angles window has been opened (see 8.5, page 34). The “name” is the name of the corresponding collision group of the current timestep.

Plane Orientation Angles

Name	DIP	DIP-Direction
1056	34.7	10.8
1411	42.8	11.2
4417	52.6	54.6
739	50.5	49.8
749	46.3	-164.9
491	39.6	-129.7
2533	35.0	-125.2
627	54.1	-164.5

Figure 18.2.: Plane orientation angles

Dip and Dip-Direction displays the data in relation to a Stereoplott and to the data shown in the collision viewer (see 17, page 61).

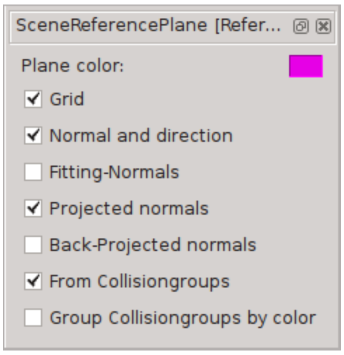


Figure 18.3.: Reference plane properties

With a double click on the reference plane in the scenetree, the properties of the plane can be configured. The color of the shown plane can be adjusted with the “Plane color” option. The “Grid” option will show or hide the plane. The normal and direction of the reference plane can be displayed in the scene. “Fitting-Normals” will show the normals of any fitting planes or from the current collision groups if “From Collisiongroups” is activated, as seen in figure 18.1.

“Project normals” will show the projected normals on the reference plane. “Back-Projected normals” will back project the projected normals on the fitting plane[1] and show them in the scene window. The collision group data can be grouped with “Group Collisiongroups by Color”. In this case, there will be only one normal for all CollisionGroup objects with the same color. The new normal is based on the average of all grouped normals.

[1]If collision groups are used, a fitting plane over all faces of the group will be used.

19. Diagram

The diagram feature in OFA supports many different diagram types to display any data calculated through the process of a collision path (see 16, page 57).

19.1. Diagram creation dialog

A new diagram can be created through the diagram menu (see 7.4.1, page 25). On conclusion of the dialog creation window, the diagram window will be shown (see 19.2, page 70).

19.1.1. Data selection

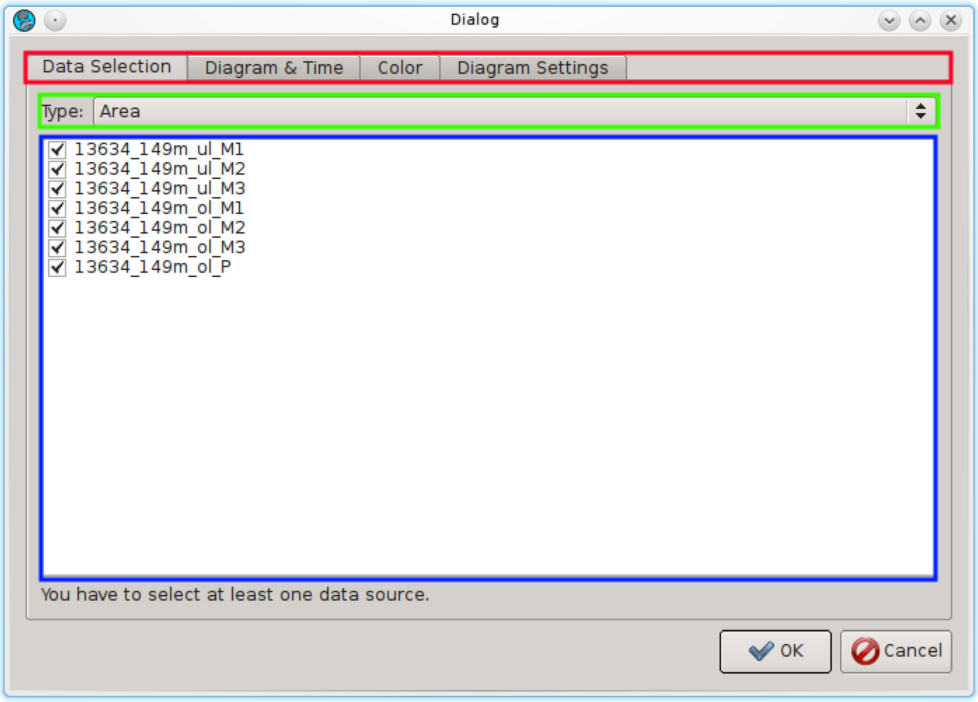


Figure 19.1.: Data selection

The colorations in figure 19.1 as follows:

- *RED:* The menues to specify the diagram. Every content of a menu is based on the selection of its former left menu.
- *GREEN:* The type of data to display.
- *BLUE:* The models to create the diagram for.

There are the following data types to evaluate as diagrams:

Type	Explanation
Area	The area of the collisiongroups
Faces	The number of faces (triangles) of the collisiongroups
Perimeter	The perimeter of the faces of the collisiongroups
Perimeter to Area Quotient	The perimeter of the faces devided through the area of the collisiongroups
Hierarchy[1]	The hierarchy of collisiongroups of one model
Steroplott[1]	The steroplott of the selected models to one time step (see 4.3, page 13)

19.1.2. Diagram and Time

The Diagram and time menu is available to specify the type of diagram to use to display the chosen data. The following diagram types are available:

- Linediagram
- Bardiagram
- Stackedbardiagram
- Circlediagram
- Areadiagram
- Hierarchydiagram[2]
- Stereoplottdiagram[3]

There are several options to augment the diagram:

Option	Explanation
Show Background	If the background is not shown, the background of the diagram will be transparent
Show XY-axis[4]	A x-and y axis can be displayed
Show Legend	A Legend of the diagram with the color assignment to the collisiongroups can be displayed
Headline	A headline can be placed at the top of the diagram

[1]Hierarchy and Steroplott are special diagrams and not a selection of data to display. The can not be evaluated through different diagrams shown in 19.1.2, page 67.

[2]This is a special diagram and can only be used with the data selection Hierarchy.

[3]This is a special diagram and can only be used with the data Stereoplott. Also a reference plane has to be added to the scene.

[4]This option is not possible for a Circlediagram or Stereoplottdiagram.

Also the data displayed on the y-axis can be scaled with the Y-Scaling option. The factor of 1.0 means no scaling at all.

The range of the date specified in 19.1.1, page 66 can be specified with the time step (see 4.3, page 13) range. Dependent on the type of diagram there will be from and to available. Circlediagram and Stereoplottdiagram will only display data to one specified time step.

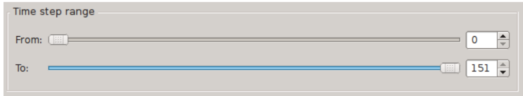


Figure 19.2.: Time step range

19.1.3. Color

The color menu will only be available for the Linediagram and Areadiagram, as they use one color for a model for the whole range specified in 19.1.2, page 67. All other diagram types use the color of the collisiongroup the data from is shown. This color can be changed in the Collison viewer (see 17, page 61).
For each model, a color is assigned per random. This color can be changed with a double click on the color.

19.1.4. Diagram Settings

The options in the diagram settings menu depend on the used diagram type. All possible options are the following.

Global options defines the width of any used line to display diagram data in pixel.

Figure 19.3.: Global options

XY-Axis options is available if the XY-Axis flag has been checked under diagram options (see 19.1.2, page 67). With show Y-axis the y-axis can be enabled or disabled. The possibility to enable cross beams are for easier reading of the y values, as on each cross beam, the corresponding value will be displayed on the y-axis. With the cross beam count, the number of lines to cut the y-axis can be set. Also with the enable axis arrows, arrows will be attached to the end of the axes lines, showing the direction. Additionally colors can be set for the x and y axis and for the displaying value text. The next options are for an optional text as inscription on the axes. The units in which the data is displayed can also be set, also per default, the correct unit will be set analog to the chosen data type.

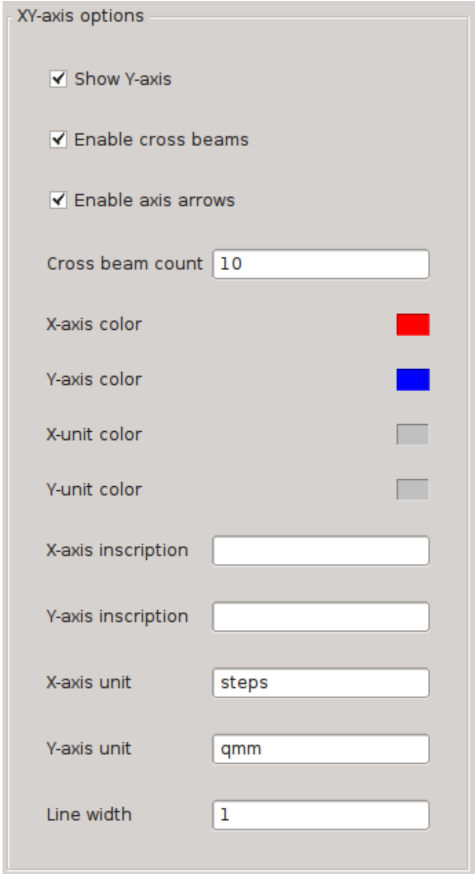


Figure 19.4.: XY-axis options

If the legend option has been selected, the legend text color can be chosen.

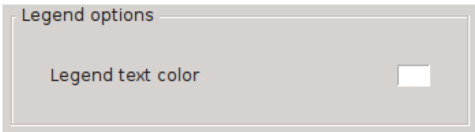


Figure 19.5.: Legend options

The text of the headline, which will be shown at the top the diagram, can be specified in the headline option. Also the color to use can be selected.

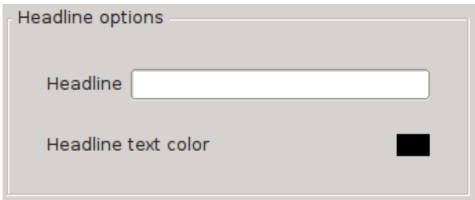


Figure 19.6.: Headline options

The background color can be selected in the background options, if the background has been activated.

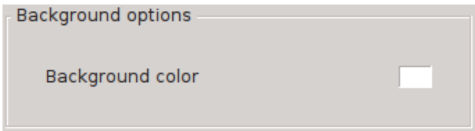


Figure 19.7.: Background options

The group text color will be used for the collision group names inside of the hierarchy diagram. The line width is for any lines used in the diagram itself.

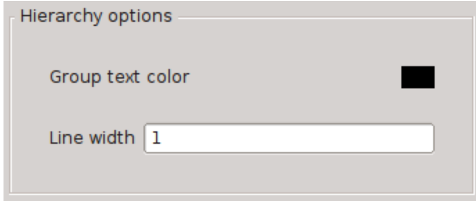


Figure 19.8.: Hierarchy options

At default, all vectors shown in the steroplott are alligned to PI (unit cirle). The can also be alligned to PI/2 with the allign top option. The use scene fitting planes will use the normal vector of any created fitting plane (see 8.7, page 35 and 18, page 64) instead of the normal vectors of the collision groups (see 4.2, page 13). Show angles will display the dip and dip-direction of any projected normal below of the diagram. Right, Top, Left and Bottom are the sides of the outer circle with their appropriate names, which are set by default to buccal, mesial, lingual and distal. Inner and outer ring color specify the color of the steroplott diagram rings. The ring description sets the color for the angles at the inner rings, describing the inclination to the reference plane. In case the collision groups are shown, there name can be displayed behind the projected normal with the show collision group names option. Additionally the line width for the rings and projected normal vectors can be set.

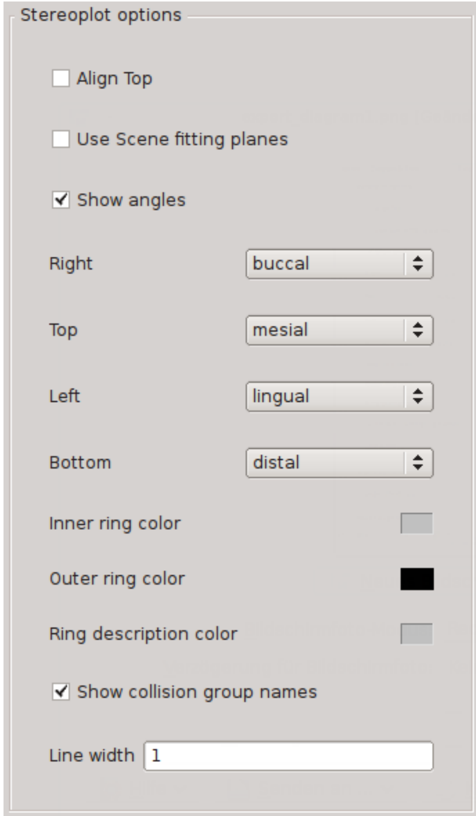


Figure 19.9.: Steroplott options

19.2. Diagram window

After the the conclusion of the diagram creation dialog, a window with the new diagram, like in figure 19.10[5], will be shown.
Coloration as follows:

- *GREEN:* Window manipulation options
- *BLUE:* File manipulation options
- *RED:* Diagram manipulation option

With the Window manipulation options the window can be manipulated in the following way:

[5] The shown window is a restored window of the default maximized window. To restore down from the maximized window, or to minimze it, the controls in the upper right edge can be used.

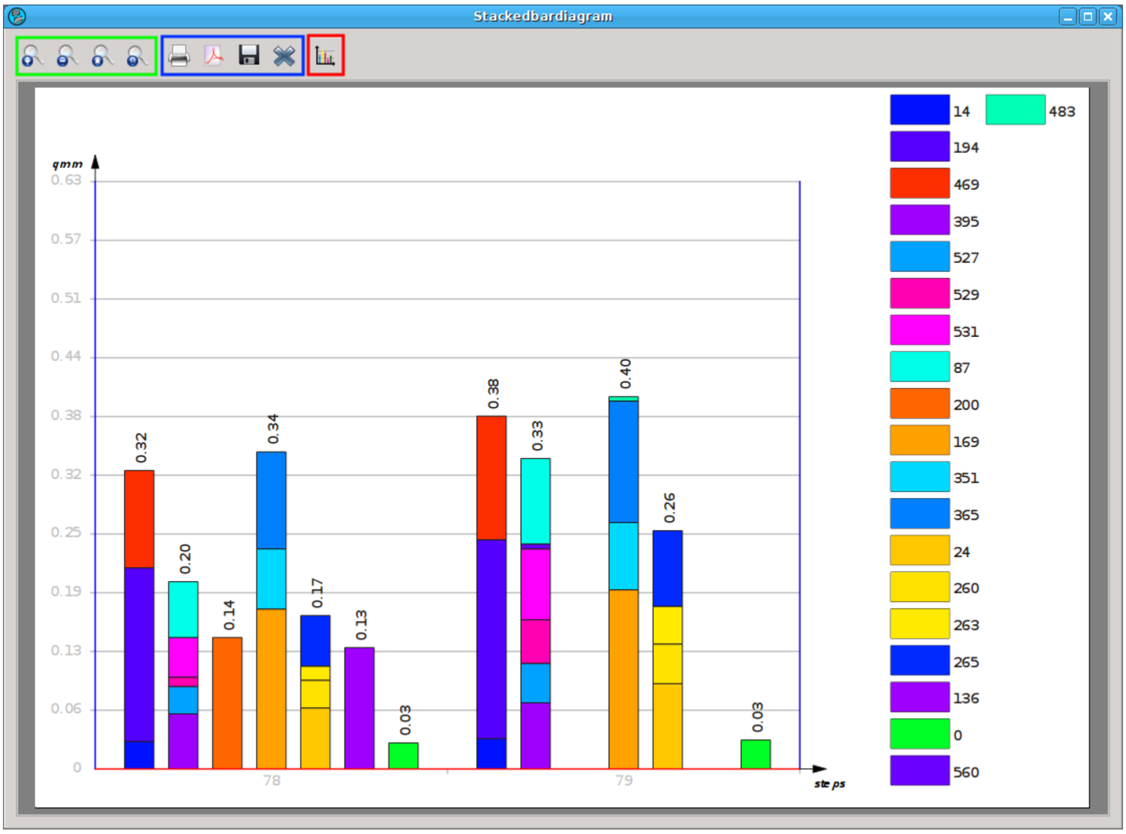


Figure 19.10.: Diagram window

- Zoom in *Ctrl + +*
- Zoom out *Ctrl + -*
- Zoom page width *Ctrl + 0*
- Zoom to page

Also the diagram window can be manipulated by mouse: Ctrl + mouse wheel will zoom in or out, pressing the left mouse button can be used to move in the zoomed window.
The following options can be used for file operations:

- Print, to print the file with a printer
- Export as PDF, with the diagram stored as a vector graphic inside of the PDF
- Export, to export it as jpg, png and other file formats
- Close, to close the window

With the button in the red colored area, the diagram settings can be reopend to change the selected settings.
In case a hierarchy diagram is displayed, a double click on a group in the diagram will open up the collision viewer (see 17, page 61) with the selected group.

For none vector formats, like jpg, DPI can be set at the bottom of the export dialog. Also the number of megabytes the exported picture will use is shown. Image resolution is restricted. To get a higher DPI resolution, the number of timesteps displayed in the diagram have to be shortened.

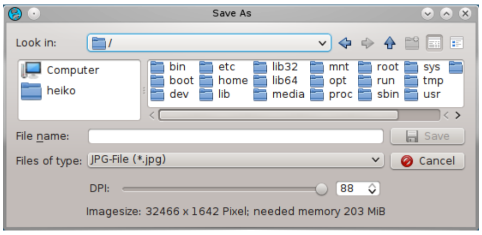


Figure 19.11.: Diagram export

Additionally the diagram can be opened during path calculation, to interpret the data live. It is recommended to disable the legend for this feature. If the bottom slider of diagram window is moved do the right edge, it will always be moved to the latest timestep.

19.3. Documents

The document feature can be used to rapidly create a document with data created in OFA, like diagrams or screenshots. There is a edit mode for documents and a view mode.

19.3.1. Creating a document

To create a new document the Diagram menu can be used (see 7.4.2, page 26). The document will be opened in edit mode and will look like in figure 19.12.

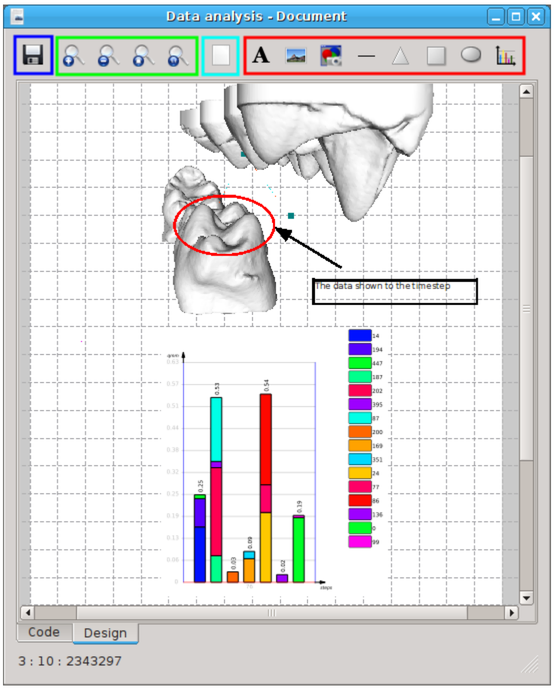


Figure 19.12.: A OFA document

Coloration as follows:

- *BLUE:* File manipulation option

- *GREEN:* Window manipulation options
- *CYAN:* Document manipulation option
- *RED:* Document items

The save option, will save the document in the OFA project file. It can be reopend through the documents feature (see 7.4.3, page 26) and 19.3.3, page 77). The window manipulation options are alike to the diagram window manipulation options (see 19.2, page 70). Through the empty page icon in the cyan colored area, a new page will be added to the end of the existing page(s). Also through a right click on the displayed page a menu will be opened where a page can be added, the current page can be removed and the properties of the current page can be specified.

The properties of the current page include the page size with it‘s width and height and orientation of the page, for portrait or landscape. Additionally the border margin of the page can be specified, displayed as a gray border in figure 19.12.

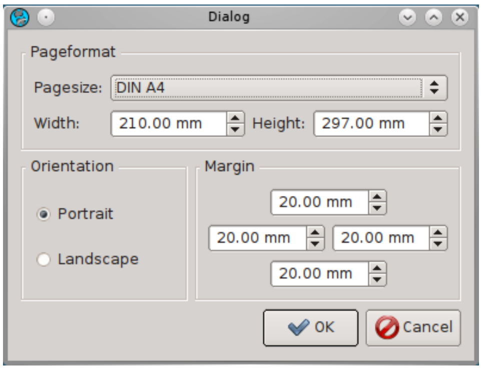


Figure 19.13.: Page properties

The following document items are available:

- Text *CTRL + T*, to insert a text block
- Image *CTRL + I*, to load and display an image
- Scene *CTRL + G*, to insert a screenshot of the current scene
- Triangle *CTRL + A*, to insert a triangle
- Rectangle *CTRL + R*, to insert a rectangle
- Circle *CTRL + C*, to insert a cirlce
- Diagram *CTRL +D*, to create and insert a diagram

19.3.1.1. Document item handling

Document items can be resized, moved or deleted. To resize a document item, it has to be selected with a left mouse button click on the item.

Multiple items can be selected with *SHIFT* while holding the left mouse button. If multiple items are selected, the can be grouped with *G* to manipulate the group instead of single items. If a group is selected it can be ungrouped with *SHIFT + G*.

A selected item will be shown with a green border. On each side and edge, a clickable square is available to resize the selection. The blue squares can be used for a proportional resize. Also the item can be moved on the page through a click on the selected item and moving it, while holding the left mouse button.

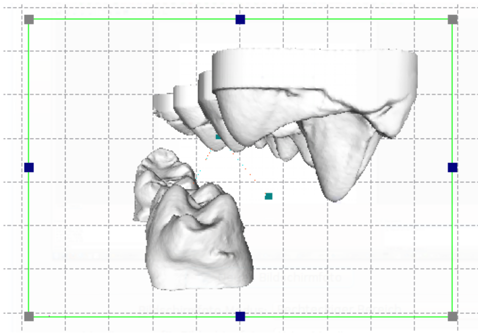


Figure 19.14.: Item selection

The right click context menu will list all item names below the mouse pointer. Each object can be selected or the properties accessed. The selection menu is very usefull if items are above each other.

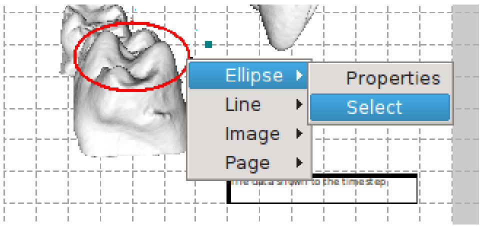


Figure 19.15.: Item context menu

The border properties can be used to create a border around an item. To make a border visible the respective border entry has to be set to 1. The color of the border can be specified through the color option. Also the border type can be choosen with the style option:

- dash line
- dash-dot line
- dash-dot-dot line
- dot line
- solid line
- none

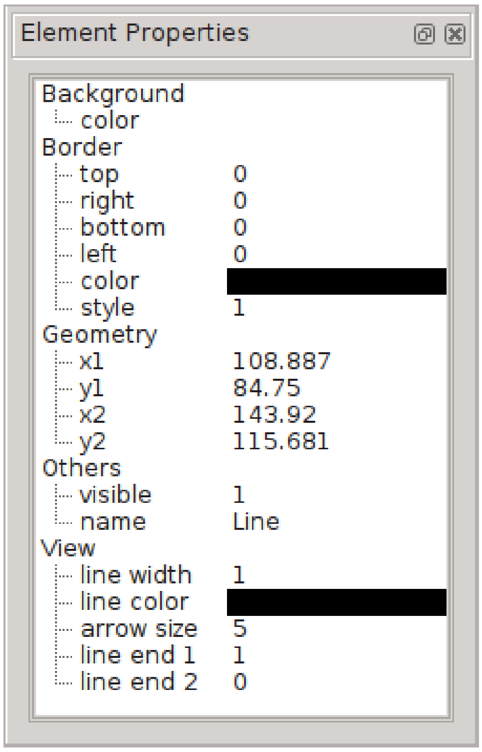


Figure 19.16.: Item properties

The properties of an item can be viewed with a double click on an item or a right click context menu of the item and the properties submenu below the item name.
“Geometry, ”Others“ and ”View“ properties may differ from item typ to item type. For images, circles and rectangles the x and y value will define the upper left point of the green

selection border. For a line, x1 and y1 will define the first point and x2 and y2 the second point. For a triangle likewise, with a third point for the third edge of the triangle. For those items with one point, width and height will define the area of the item.
The visible option in the ”Others“ menu will make the item invisible in view mode. In edit mode, it will still be visible, so that it can be manipulated.
The text item has also a text option with HTML text inside of the text option. Furthermore the name is displayed, to identify it in the context menu.
”View“ menu is available to specify line width, color or line ending in case of a line item. The line ending for line items can be set to arrow on ether end. The arrow size can be specified with arrow size. For rectangle, triangle and circle is also a fill color available. The background color will fill the bounding box around an object with the given color.

For text items, a double click will open a text input dialog. The properties can only be viewed through the right click menu. For newly created diagrams, a double click will reopen the diagram settings. For saved documents a double click on the diagram will open the properties, as the data used for the creation might not be the same.

The text editor for entering any text has basic text formating features. **Coloration as follows:**

- *RED:* File manipulation option
- *MAGENTA:* Clipboard and selection options
- *BLUE:* Undo/Redo option
- *CYAN:* Text style options
- *YELLOW:* Text type and size options
- *ORANGE:* Color options

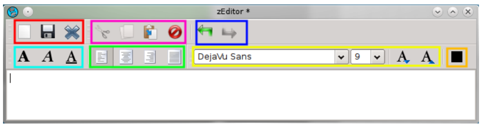


Figure 19.17.: Document text editor

With the file options, the text can be resetted with ”New“. Any changes can be saved with ”Save“. The editor can be closed with the ”Close“ option or by closing the window itself. The clipboard options will allow to cut or copy the current selection to use it somewhere else, or to paste any selection made before. The last button in this section will delete any selected text. The undo options can be used to undo or redo any changes made to the text. With the text style options, text can be set to bold, italic or underlined text. Also a compination of these options is possible by clicking or uncklicking them. The text orientation options can be used to left or right align, to center or justify the selected text. With the text type and size options, the font type for the selection and point size can be specified. The last option can be used to set the text foreground color.

As the document is open in edit mode, the code (written in XML) itself can be edited with the code tab in the lower left (see figure 19.12). This should not be necessary but can be done for advanced document handling.

19.3.2. Report

A report can be used to quickly create data about the collision groups and time steps created of a collision path (see 16, page 57). The report can be created through the diagram menu (see 7.4.4, page 26). The report will include all diagrams set through the report wizard in the tools menu (see 7.7.5, page 30).

The wizard will, step by step, set a template for a diagram. The first step is the diagram type and the following steps are all like the tabs in the diagram creation dialog (see 19.1, page 66). Only the time setting is not available as the time can not be pre-estimated. The "back" button can be used to change any settings made.

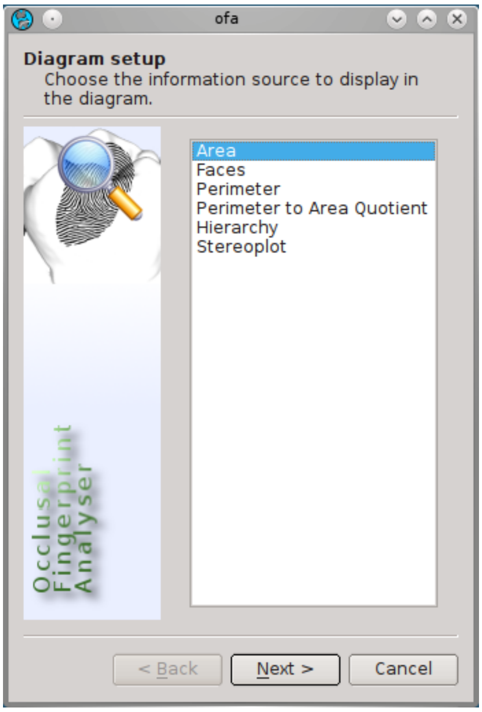


Figure 19.18.: Report Wizard

The last page of the diagram wizard can be used to add another diagram to the report, with the use of the "Add another diagram" button. "Finish" will save the report template which will be applied to each created report. A report can be created automatically after each calculation of a collision path. To activate it see 10.5, page 46 for more information.

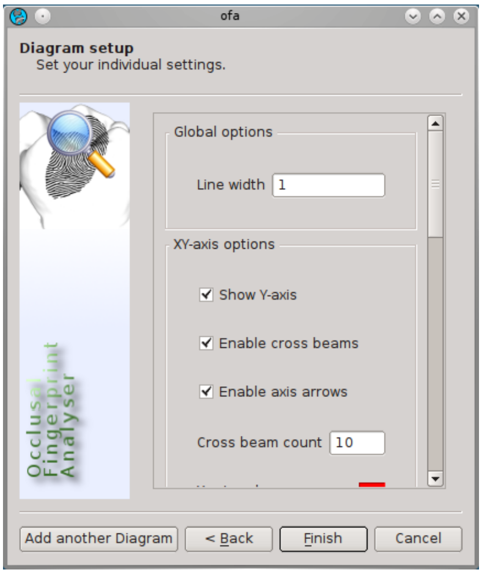


Figure 19.19.: Finishing a report wizard

19.3.3. Manage documents

An overview of all documents can be opened through the documents menu (see 7.4.3, page 26).

All created reports and documents will be displayed in the overview. They can be manipulated through the right click context menu on any document. "Show" will display the document in view mode, "Edit" in edit mode. The rename function can also be used by the shortcut *F2*. "Delete" will delete the document unrecoverable. Also a double click on the document will open it in view mode.

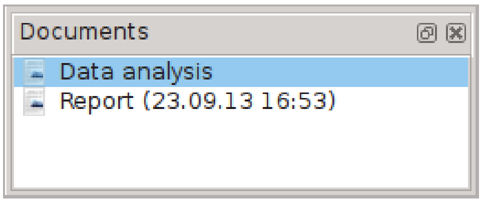


Figure 19.20.: Documents overview

A document opened in view mode, is like a pdf view. The document can be edited by using the first upper left "Edit" button or by the shortcut *CTRL + F2*. All view mode functionallity is similar to the Diagram window (see 19.2, page 70).

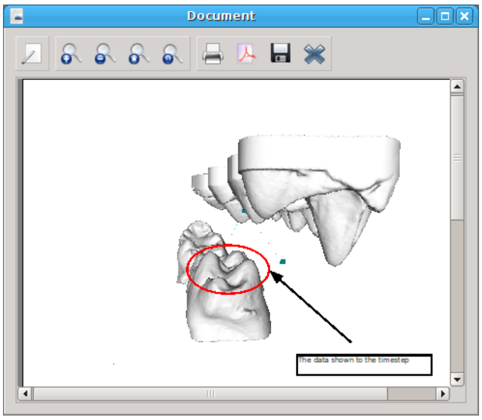


Figure 19.21.: Document view mode

20. Indentation

As often, the maximal indentation is used as a pathpoint for the trajectory, the indentation feature has been introduced in OFA. It can be used to calculate the maximal indentation, based on the maximal collision between selected model faces. The result will always be an approximated value to the maximal indentation.

20.1. Indentation setup

The Indentation setup is available through the toolbar (see 8.8, page 36). The first step is to select the static and the moving model or group. The most important selection is the choice of the grid. The grid will be used to calculate possible positions, like it can be seen in figure 20.1 with the gray points on the model. For each gray point, a collision calculations will be made.

There are multiple grid options:

- **full sphere,** will create a sphere grid with the accumulated bounding box diagonal of both models or groups as circumference. This will be a very large grid and will take a huge amount of time to calculate but can be combined with an indentation plane.
- **center center cylinder,** will create a vector between the two models or groups centers and cylinder based on that vector. This method may be inaccurate and is very depending on a good center position.
- **center center bounding box static,** will create a grid on a plane based at the center of the static model or group with a normal pointing to the nearest plane of the opposing model or group. This method is a good choice if the model or group is properly surrounded by the bounding box.
- **center center bounding box mobile,** similar to the preceding method with the center of the mobile model or group.

If an intendation plane is created through the last button on the intendation planes option in 20.1 or before through the toolbar (see 7.3.7, page 24) there are more grid options possible:

- **half sphere above intendation plane,** will use the full sphere grid and cut it through the indentation plane, by discarding every grid point below the plane.
- **identation plane based,** will create a grid on the indentation plane.

If the static element is a trajectory group there will be another option available:

- **trajectory group plane based,** will create a grid based on the plane fitted through all centers of the static element trajectory group.

The *size* option will define the percentage of the radius used. As the radius is very large (through both bounding box diagonals) it can be reduced very much. *Distance between points* defines the distance between the points in the grid and also the distance between multiple planes if used. *Offset* and *parallel planes* can be used if any plane based grid is used. *Offset* will move the planes along the planes normal by the given value. *Parallel planes* will create the given number of planes parallel to the first one.

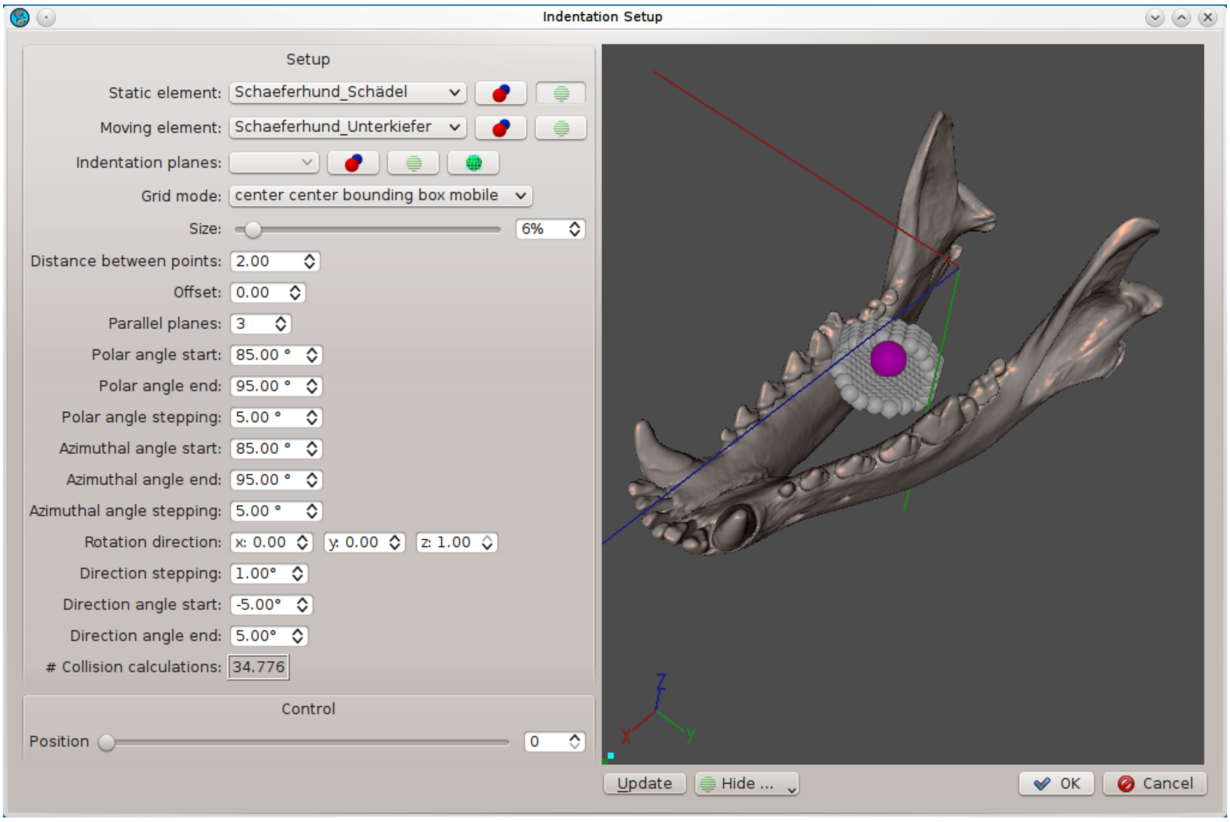


Figure 20.1.: Indentation setup gui with a grid and rotations

Furthermore rotations can be defined which will be used on each grid position. As there is no information in the model file about where the chewing surface is, a direction vector has to be defined. For example, a direction vector of $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ is shown in figure 20.2. The possible roations are displayed as blue points.

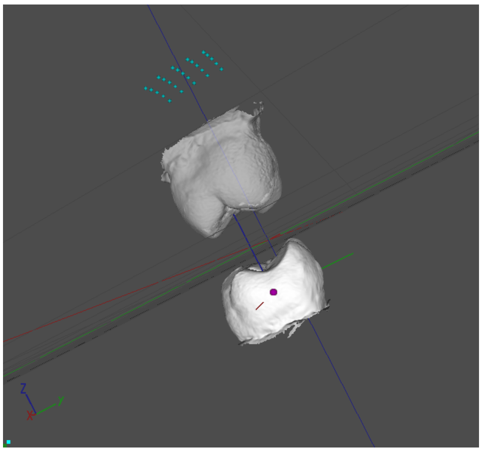


Figure 20.2.: Rotation points based on a given direction vector

The roations are defined in spherical coordinates with *azimuthal* angle θ start, end and stepping. The same for *polar* angle ϕ. An example for *polar* is displayed in 20.3 with a start value of $45°$, a stepping of $45°$ and end $135°$. In the case of the example two roation points would be used, as $90°$ is the direction itself. If the same values would be used for *azimuthal* and *polar* there would be eight roations and with that nine calculations per position if there is no direction rotation defined. Recommended would be a range between $85°$ and $95°$ as any more would rotate the model or group unnatural if it has been pre-aligned. Setting start and end value of *azimuthal* or *polar* to $90°$ will do no rotation for the specified angle.

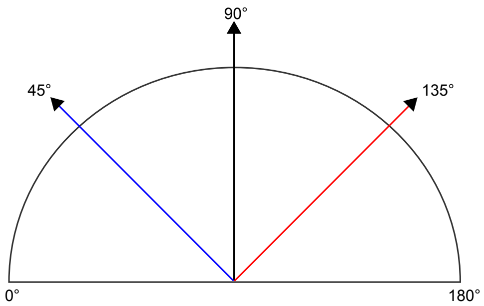


Figure 20.3.: Rotation schematic

The direction rotation rotates around the direction vector as displayed in 20.4. The settings are similar to *azimuthal* and *polar*, only that $0°$ for start and end would be no self rotation instead of $90°$ with *azimuthal* or *polar*.

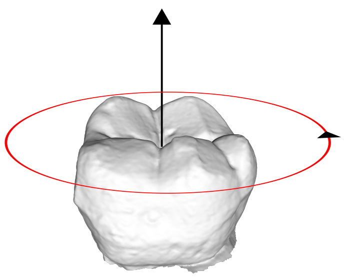


Figure 20.4.: Self rotation schematic

The number of calculations is calculated through the following formula: *number of calculations = number of positions * number of rotations * number of self rotations*. Furthermore, all possible positions can be tested by using the *Position* slider.

By using the "Ok" Button of the indentation setup, a so called *gridjob* will be started, containing all configured information. If a face selection has been made beforehand only these faces will count if they have collision. The area displayed in the result will reflect this.[1]

20.2. Network

If Bonjour/Avahi is installed the OFA service will automatically be announced in the local network and any started OFA will be used for indentation if it has no current open project. Otherwise OFA will try to connect to all configured IP addresses (see 10.7.2, page 47). On computers with more than three CPUs, multiple local processes can increase the performance of the indentation calculation. For example, one or two additional clients with four (physical) cores is a good choice. To test this feature, it is recommended to watch the core load, to check if too much or too few clients have been started.

20.3. Calculated gridjobs

Already calculated grid jobs can be reopend through the toolbar (see 8.8, page 36) or the menubar (see 7.8, page 30). The Gridjobs window will display the date and time of the available gridjobs and any iteration steps of a refine calculation.

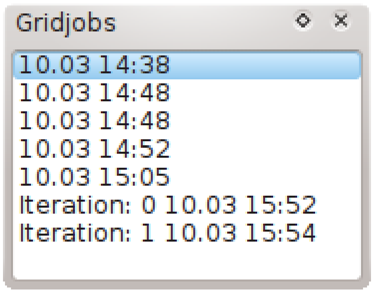


Figure 20.5.: Gridjobs window

A right click on an entry opens the following options:

- **Delete**, to delete the grid job and the saved result points.
- **Open**, to reopen the indentation setup (see 20.1, page 78) with the selected values.
- **Refine**, to refine the results (see 20.4, page 82).

A double click on an entry will open the result gui. Also refinement steps with the calculated result points are listed.

[1]If a face selection has been made, a later different face selection might not show the correct results if the number of stored results has been set to greater then 0 (see 10.7.1, page 47).

20.3.1. Gridjob results

The result of the Indentation calculation will be a list of positions with the achieved collisions for each position, if the position is valid. The gridjob results will be opened automatically after a indentation or refine calculation.

The result list is orderd by the collision area of the position. For each entry, the actual position, rotation vector and self rotation is displayed. Furthermore a refince checkbox is available to improve the position (see 20.4, page 82). Refince can be started by using the right click context menu. The update button will update all positions to any current face selection (see 7.3.5, page 24 or 8.4, page 34 face selection).

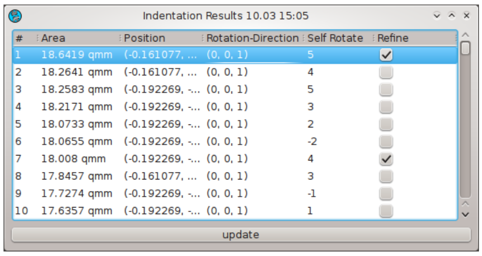


Figure 20.6.: Indentation results

20.4. Refine

Refine is an optimization operation like simulated annealing or genetic algorithms to improve the current results by using the best current values. The algorithm will expand the position in each direction with a decreasing range for each iteration and analog for all degree settings.

TIP: With refine, it is possible to create a wide-meshed grid and improve any good looking positions.

The refine dialog shows as its first option the *gridjob name* for the grid job to improve. *Max iterations* will define how many iterations there will be for a position. *Degree stepping* is for *azimuthal* and *polar* settings. It works closely together with *degree multiplier* and *degree reduction*. If, for example, a degree stepping of 3° has been defined, the degree multiplier will mutiply this by the given value. If a value of 2 has been set, refine would start at the current rotation +-6°. Degree reduction will reduce the degree stepping with each iteration, making the rotation narrower. This is simillar for self rotation with *self degree stepping*, *self degree reduction* and *self degree multiplier*. Just as it is similar to the distance settings, only that the inital distance value will be taken from the gridjob distance. *Top usage* defines how many of the best values should be improved. For example, a value of 5 will use the first five results of gridjob result list and the five best from each following iteration.

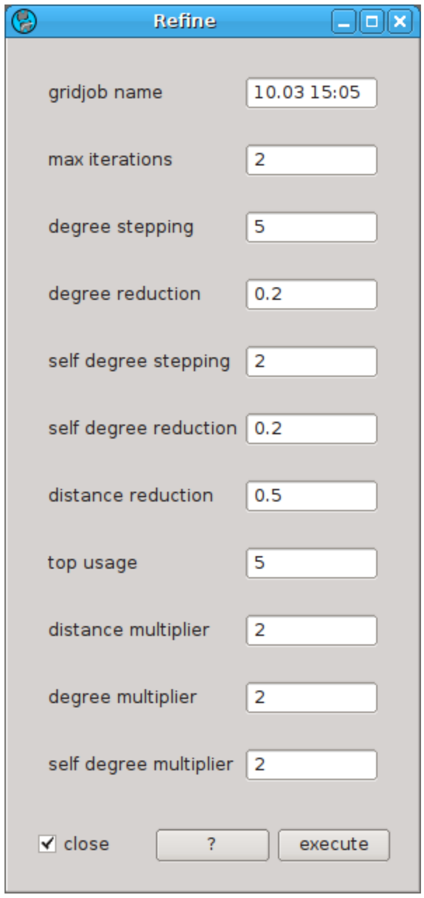


Figure 20.7.: Refine dialog

20.5. Converge

The converge option can be used any time and has been programmed to finalize a calculated position, but it can also be used after a trajectory path has been calculated on a choosen timestep. To use the converge algorithm, a model has to be selected. Through using the right click context menu of the scenetree (see 15.0.1, page 53) converge will start. If any collision data is available (through indentation or trajectory) the normal vectors of the colliding faces will be used to determine the convergence vector. Otherwise the center of the nearest opposing model will be used. The converge algorithm will stop if a collision of the choosen element has occurred.

List of Figures