NMSBUILDER

v2.1

User Guide

Abstract

This is the manual of the nmsBuilder software. It is intended to be a guide for the user, and we hope it can help rather than confuse you. Please be aware that the manual is work in progress. It may not be complete or updated, and we cannot ensure all the information in it is 100% correct.

nmsBuilder is not intended for clinical use, nor for commercial purposes (e.g. consultancy), nor for redistribution. You use it at your own risk, and you are responsible for the interpretation of any results.

Being a research tool under continuous development, nmsBuilder may not work exactly as documented, and sometimes it may not work at all (loop, crash, etc.)! Please report to us if you find any bugs (experts@nmsbuilder.org), or would like to suggest improvements (info@nmsbuilder.org).

Contents

1 License

nmsBuilder v2.1 use license agreement, license grant and restrictions on use.

The software named nmsBuilder v2.1 and any accompanying documentation, manuals or data (hereafter collectively "software") is property of Istituto Ortopedico Rizzoli and the copyright law protects it.

A revocable, limited, non-transferable, non-assignable, non-exclusive use license is hereby granted by Istituto Ortopedico Rizzoli, (hereafter "licensor") to the user named below (hereafter "user") for the limited use of the Software as long as the Software is used in accordance with all the provisions of this Agreement.

software is intended only for non-commercial research and education. It may not be used in relation to any commercial or military activity. software use must be acknowledged in all publications, presentations, or documents describing work in which software is used.

It is understood by user that software is made available only on the condition that software will not be modified in any way or otherwise transformed into any other form. user shall not modify, translate, reverse engineer, decompile, or disassemble software or any portion of software. user shall not distribute software or create and distribute derivative works based on software. User shall not duplicate in any form, except for archival purposes, the programs, program statements or any other materials relating to software supplied by licensor to user.

User shall not remove any copyright notices or other proprietary notices or labels from software.

The rights and license granted user hereunder to use software are restricted solely and exclusively to user and may not be assigned, rented, subleased, sub-licensed, sold, offered for sale, transferred, disposed of, encumbered or mortgaged.

Title and all ownership and proprietary rights to the software and any derivative works thereof (including without limitation copyright, patent, trademark, and trade secret rights) shall remain in licensor.

Since the software is provided free of charge, the software is provided on an "as is" basis. Accordingly, licensor does not make and hereby expressly disclaims, any and all representations or warranties of any kind with respect to the software, including without limitation any implied warranties of merchantability, fitness for a particular purpose, and non-infringement, and any warranty with respect to the accuracy or applicability of the software or of data or algorithms contained in the software.

Licensor accepts no responsibility for, and makes no representations as to, the accuracy or reliability of any report, output, or data prepared by or with the assistance of software. The entire risk as to the use, quality and performance of software is born by user. Should software prove defective in any way, user, and not licensor, assumes the entire cost of any service and repair. licensor is not obligated to render any technical assistance or support in the use or operation of software or to correct or fix any defects or bugs in the software.

Under no circumstances shall licensor or its affiliates, officers, directors, contractors, students, employees and agents, be liable to Licensee or any other person for any damages, losses or liabilities whatsoever or any general, special, incidental or consequential damages of any character, including without limitation damages for loss of good will, lost profits, work stoppage, computer failure or malfunction, or any and all other monetary damages or losses. In no event shall licensor or its affiliates, officers, directors, contractors, students, employees, and agents be liable to user with respect to any claim or demand made against user by any other person.

Licensee agrees to indemnify and hold licensor, and its affiliates, officers, directors, contractors, students, employees, and agents, harmless of, from and against all claims, costs, expenses, damages, and liabilities, including reasonable attorney's fees, actually or allegedly resulting from or incident to the use, operation, or storage of software.

In view of the unique and proprietary nature of software, it is understood and agreed that licensor's remedies at law for a breach by user of user's obligations under this Agreement may be inadequate and that licensor shall, in the event of any such breach, be entitled to equitable relief (including, without limitation, provisional and permanent injunctive relief) in addition to all other remedies provided under this Agreement or available to licensor at law.

The License under this Agreement may be terminated at the sole discretion of licensor at any time by licensor. The License under this Agreement will automatically terminate upon user's failure to comply with any of the terms and conditions of this Agreement. Upon termination of the License under this Agreement for any reason, user shall make no further use of software and user shall destroy all copies of software.

nmsBuilder is partially based on OpenALBA and OpenSim.

1.1 ALBA Licence

This is an open-source copyright as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.

This software is provided by the copyright holders and contributors 'as is' and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the regents or contributors

ALBA is partially based on:
VISUALIZATION TOOL KIT (VTK) (**1**)
WXWINDOWS LIBRARY (**2**)
GDCM (GRASSROOTS DICOM) (**3**)
ITK LIBRARY (**4**)
XERCESC LIBRARY (**5**)
CRYPTO++ LIBRARY (**6**)

1. **The Visualization Toolkit (VTK) copyright**

2. **wxWindows Library Licence**

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS LIBRARY LICENCE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. see the GNU Library General Public Licence for more details.

EXCEPTION NOTICE

- As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3 of the Licence document.
- The exception is that you may use, copy, link, modify and distribute under the user's own terms, binary object code versions of works based on the Library.
- If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.
- If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

3. **GDCM (Grassroots DICOM) licence**

Copyright (c) 2006-2016 Mathieu Malaterre Copyright (c) 1993-2005 CREATIS (CREATIS = Centre de Recherche et d'Applications en Traitement de l'Image) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither name of Mathieu Malaterre, or CREATIS, nor the names of any contributors (CNRS, INSERM, UCB, Universite Lyon I), may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright holder or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

4. **ITK Library licence**

The Insight Segmentation and Registration Toolkit (ITK) was initially developed under contract to the National Library of Medicine at the National Institutes of Health. ITK is partially derived from VTK and VXL, hence some code is copyrighted accordingly (see VTKCopyright.txt and VXLCopyright.txt).
The ITK copyright is as follows:
Copyright (c) 1999-2003 Insight Software Consortium All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the Insight Software Consortium, or the names of any consortium members, or of any contributors, may not be used to endorse or promote products derived from this software without specific prior written permission.
- Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.

This software is provided by the copyright holder and contributors “as is”’ and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a

PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

See also the ITK web site: http://www.itk.org for more information.

5. **XercesC Library licence**

   Copyright 1999-2004 The Apache Software Foundation.

   Licensed under the Apache License, Version 2.0 (the ”License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software

   distributed under the License is distributed on an ”AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

6. **CRYPTO++ Library licence**

   Compilation Copyright (c) 1995-2003 by Wei Dai. All rights reserved. This copyright applies only to this software distribution package as a compilation, and does not imply a copyright on any particular file in the package.

   The following files are copyrighted by their respective original authors, and their use is subject to additional licenses included in these files.

   mars.cpp - Copyright 1998 Brian Gladman.

   All other files in this compilation are placed in the public domain by Wei Dai and other contributors.

   Permission to use, copy, modify, and distribute this compilation for any purpose, including commercial applications, is hereby granted without fee, subject to the following restrictions:

   (a) Any copy or modification of this compilation in any form, except in object code form as part of an application software, must include the above copyright notice and this license.

   (b) Users of this software agree that any modification or extension they provide to Wei Dai will be considered public domain and not copyrighted unless it includes an explicit copyright notice.

(c) Wei Dai makes no warranty or representation that the operation of the software in this compilation will be error-free, and Wei Dai is under no obligation to provide any services, by way of maintenance, update, or otherwise.

THE SOFTWARE AND ANY DOCUMENTATION ARE PROVIDED ”AS IS” WITHOUT EXPRESS OR IMPLIED WARRANTY INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL WEI DAI OR ANY OTHER CONTRIBUTOR BE LIABLE FOR DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

(d) Users will not use Wei Dai or any other contributor’s name in any publicity or advertising, without prior written consent in each case.

(e) Export of this software from the United States may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

(f) Certain parts of this software may be protected by patents. It is the users’ responsibility to obtain the appropriate licenses before using those parts.
If this compilation is used in object code form in an application software, acknowledgement of the author is not required but would be appreciated. The contribution of any useful modifications or extensions to Wei Dai is not required but would also be appreciated.

1.2 OpenSim Library License

The OpenSim API is a toolkit for musculoskeletal modeling and simulation.

See http://opensim.stanford.edu and the NOTICE file for more information. OpenSim is developed at Stanford University and supported by the US National Institutes of Health (U54 GM072970, R24 HD065690) and by DARPA through the Warrior Web program.

2 What is nmsBuilder?

nmsBuilder is a freely-available software application to perform the pre-processing phase of subject-specific musculoskeletal modeling from three-dimensional geometries and to create models in the OpenSim file format. Specifically, nmsBuilder can import different kinds of biomedical data, interactively visualize them, fuse them in a coherent representation, manipulate landmarks, surfaces and volumes, define and calculate musculoskeletal parameters, and finally create and export musculoskeletal models for the OpenSim software. nmsBuilder is developed at the Bioengineering and Computing (BiC) of the Rizzoli Orthopaedic Institute (Bologna, Italy).

3 How to cite nmsBuilder?

Please cite the following paper when using nmsBuilder for your research:

Valente G, Crimi G, Vanella N, Schileo E, Taddei F. nmsBuilder: freeware to create subject-specific musculoskeletal models for OpenSim, Comput Methods Programs Biomed[1]

4 System Requirements

To use this software you just need a PC running Microsoft Windows (we are sorry but currently other operative systems are not supported). The minimum system requirements are:

- Operating System: Windows (XP or later versions)
- Processor: 1.0 GHz
- RAM: 1GB (3 GB recommended)
- Graphics Card: Compatible with OpenGL 1.2
- Hard Disk: 40MB

5 nmsBuilder basics: software framework

To allow the user to interact with and visualise data, we embedded the core algorithm of nmsBuilder in a software framework based on the Visualisation Toolkit (www.vtk.org) and other specialised libraries. Within this framework, the main entities exposed to users are:

- Data objects, called also Virtual Medical Entities, VMEs for short in the following;
- Views, that provide interactive visualisation of the VMEs;

- Operations, that create new VMEs or modify existing ones (special Operations are the Importers, that let you import and convert into a VME almost any biomedical dataset, and the Exporters that can convert the VME into files formatted according to common standards);

- Graphical User Interface (GUI) components;

Each session of nmsBuilder , including data, can be saved and reloaded. When saving, object properties and relationships among objects are stored in an XML file with extension .nms . This .nms file links to a complementary folder, in which the datasets relative to each object are stored in binary format.

5.1 VMEs

VMEs provide the data representation. A VME represents a data entity. There are many VME types, that can be used to represent almost any biomedical image or signal. VMEs are organized in a hierarchical tree (VME Tree or data Tree) and they are all composed by a dataset, a matrix that defines the pose of the VME with respect to its parent in the VME Tree, and a set of metadata that provides all the attributes of the VME.

5.1.1 Root

Root is the base VME of the datatree. This vme is used as entry point for the creation of all other VMEs. This VME cannot be created by the user but is created when nmsBuilder is launched or a new .nms is created (*Path: File → New*)

5.1.2 Group

Group is a VME used for grouping and tree management propouse. You can create this VME in the tree contextual menu (Par. 6.2.3.3). The name of the group can be changed in the VME tab.

5.1.3 Volume

Volume is a VME that contains a volumetric 3d dataset tipically obtained from a CT or MRI acquisition. The data contained in this dataset is stored in a regular grid of voxels. This VME can contains two type of data:

- Structured Points: this is a topological and geometrical regular array of points. In this case all voxels have the same size.

- Rectilinear Grid: represents a geometric structure that is topologically regular with variable spacing in the three coordinate directions x-y-z. In this case the size of voxels is not constant.

5.1.4 Surface

Surface is a VME that contains a surface dataset made of polygon meshes. Visual properties can be edited in the Visual props tab.

5.1.5 Landmark

Landmark is a VME that contains a landmark defined by the 3D coordinates of a point.

5.1.6 Landmark Cloud

Landmark Cloud is a VME that contains a group of landmarks defined by the 3D coordinates of the points. The radius and the resolution of all the landmarks can be edited in the VME tab.

5.1.7 Parametric Surface

Parametric Surface is a VME that contains an implicit surface parameterizable. This VME can be defined in the VME tab. The user can choose the type of the new surface (sphere, cone, cylinder, cube, plane and ellipsoid) and set the corresponding parameters. For example, if the user creates a sphere surface, the parameters to set are radius and the number of triangles that are generated by the curvature resolution. Visual properties can be edited in the Visual props tab.

5.1.8 Reference System

Reference System is a VME that contains an orthogonal reference system. This VME can be defined in the VME tab in three ways:

1. Pick the Normal radio button (which is the default option) to define a Reference System that inherits the parent's pose matrix.

2. Pick the Select Origin radio button to define an origin by hitting the Origin button and choosing a Landmark VME from the data tree.

3. Pick the Select Plane radio button to define both an origin (see point 2) and a XY plane by hitting the Point 1 and Point 2 buttons and choosing two other Landmark VMEs from the data tree.

The default dimension of the visualized Ref Sys can be edited in the scale input field of the VME tab. Visual properties can be edited in the Visual props tab.

5.1.9 Anatomical Ref Sys

Anatomical Ref Sys is a VME that contains an orthogonal reference system created from a predefined set of anatomical landmarks accoring to the recommendations of the International Society of Biomechanics [2]. The default dimension of the visualized Ref

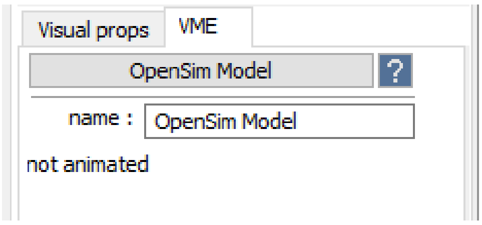
Sys can be edited in the scale input field of the VME tab. Visual properties can be edited in the Visual props tab.

5.1.10 Meter

Meter is a VME that contains length and angle measurement information. This VME can be defined in the VME tab. It takes as input two or three VMEs and gives as output the distance or the angle between them. It is possible to measure a distance between two points, between a point and a line, or the angle between two lines, choosing them from the mode drop-down menu. To measure the distance between two VMEs (line distance mode), the user selects the VMEs with the start and end buttons. To measure a distance of a point from a line (line distance mode), the user selects the point with the start button and defines the line with two VMEs selected through the end and 2nd point buttons. The angle (line angle mode) is measured between the two lines that origin at the start point and are defined by the other two. If the user moves one of the Meter origins, the meter updates itself automatically. If the plot profile checkbox is activated, a histogram dialog is opened. With the Probed button the user can selected a volume from the tree and see the histogram of the volume densities along the meter line. Visual properties can be edited in the Visual props tab.

5.1.11 OpenSim Model

OpenSim Model is a VME that contains all the information on the OpenSim objects necessary to create the models (Figure 1).



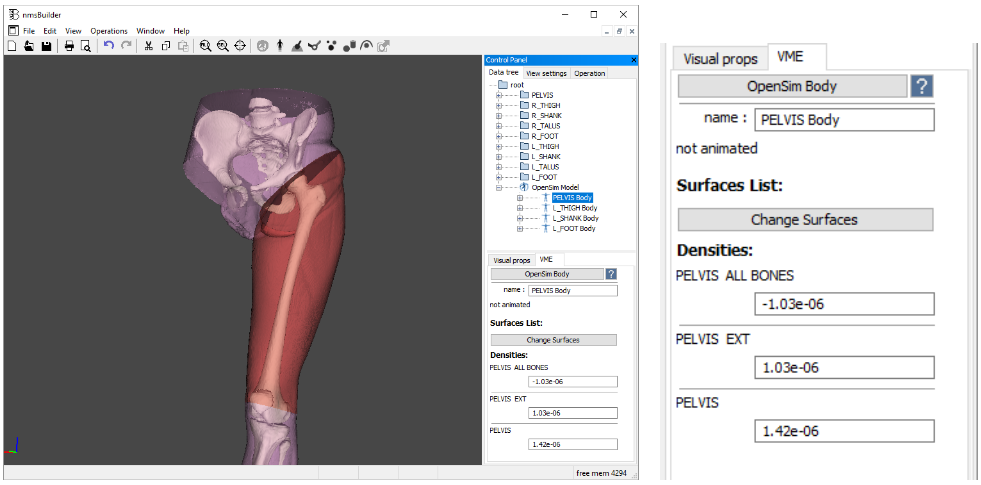
name : OpenSim Model

not animated

Figure 1

5.1.12 OpenSim Body

OpenSim Body is a VME that contains the information on OpenSim bodies. In the VME tab, the user can change the corresponding surfaces by hitting the Change Surfaces button, and the values of the body density (in kg/mm3) typing in the corresponding input fields (Figure 2).



Visual props | VME

OpenSim Body

name : PELVIS Body

not animated

Surfaces List:

Change Surfaces

Densities:

PELVIS ALL BONES -1.03e-06

PELVIS EXT 1.03e-06

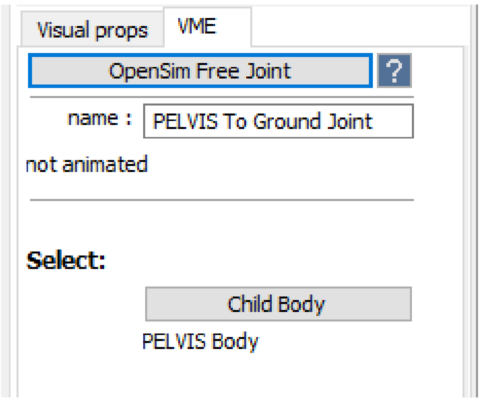
PELVIS 1.42e-06

(a) (b)

Figure 2

5.1.13 OpenSim Free Joint

OpenSim Free Joint is a VME that contains the information on OpenSim joints connecting a body with the ground. The user can change the body by hitting the Child Body button in the VME tab (Figure 3).



Visual props | VME

OpenSim Free Joint

name : PELVIS To Ground Joint

not animated

Select:

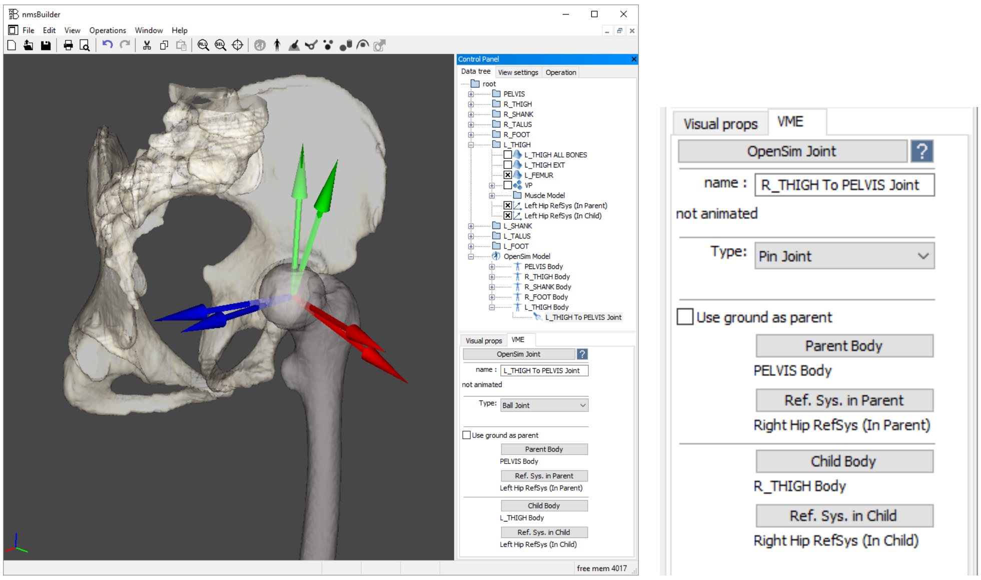
Child Body

PELVIS Body

Figure 3

5.1.14 OpenSim Joint

OpenSim Joint is a VME that contains the information on OpenSim joints connecting two bodies (parent and child). In the VME tab, the user can modify the joint contraints from the Type drop-down list, and modify the bodies and reference systems defining the joint by hitting the corresponding buttons (Figure 4).



name : L_THIGH To PELVIS Joint
not animated
Type: Ball Joint
☐ Use ground as parent
Parent Body: PELVIS Body
Ref. Sys. in Parent: Left Hip RefSys (In Parent)
Child Body: L_THIGH Body
Ref. Sys. in Child: Left Hip RefSys (In Child)

(a)

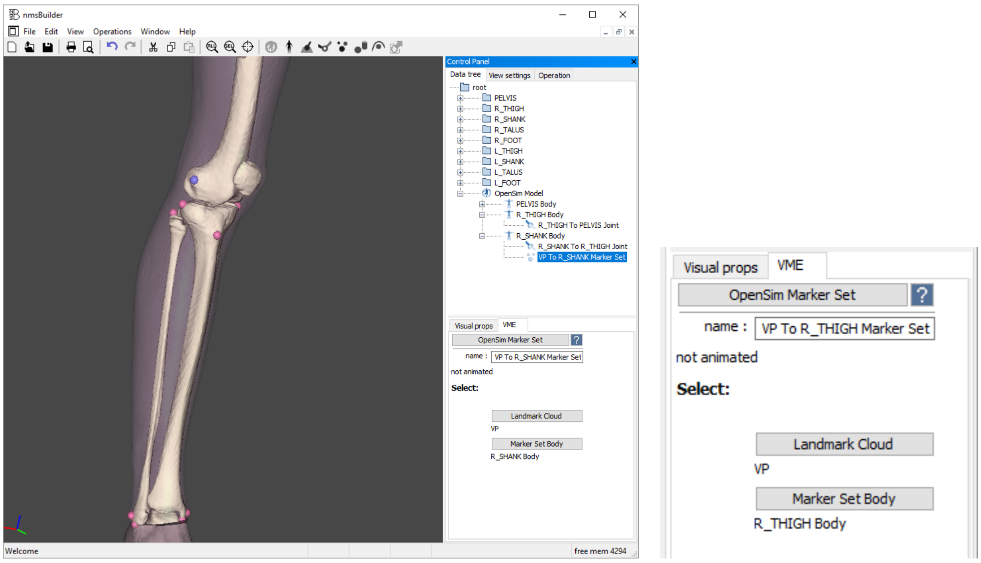
name : R_THIGH To PELVIS Joint
not animated
Type: Pin Joint
☐ Use ground as parent
Parent Body: PELVIS Body
Ref. Sys. in Parent: Right Hip RefSys (In Parent)
Child Body: R_THIGH Body
Ref. Sys. in Child: Right Hip RefSys (In Child)

(b)

Figure 4

5.1.15 OpenSim Marker Set

OpenSim Marker Set is a VME that contains the information on the markers attributed to OpenSim bodies. The user can modify the marker set and the body which it belongs to, by hitting the Landmark Cloud and Marker Set Body buttons in the VME tab (Figure 5).



name : VP To R_SHANK Marker Set
not animated
Select:
Landmark Cloud: VP
Marker Set Body: R_SHANK Body

(a)

name : VP To R_THIGH Marker Set
not animated
Select:
Landmark Cloud: VP
Marker Set Body: R_THIGH Body

(b)

Figure 5

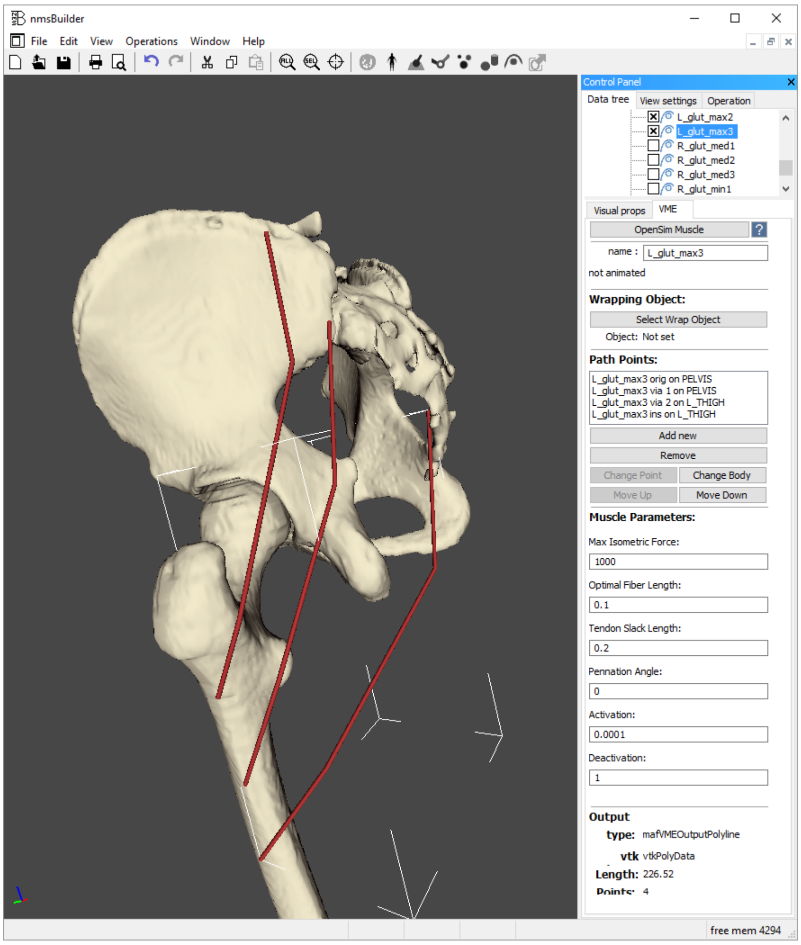
5.1.16 OpenSim Muscle

OpenSim Muscle is a VME that contains the information on the OpenSim muscle actuator. The VME can be defined in the VME tab (Figure 6):

1. Wrapping Objects: a previously-defined wrap object can be used, by hitting the Select Wrap Object button.

2. Path Points: multiple points (origin, intermediate, insertion) can be added, removed, changed and moved to define the muscle path, by hitting the corresponding buttons.

3. Muscle Parameters: musculotendon-specific parameters for contaction and activation dynamics can be defined by typing in the correspinding input fields. Default values are set automatically.



name : L_glut_max3

not animated

Wrapping Object:

Object: Not set

Path Points:

L_glut_max3 orig on PELVIS
L_glut_max3 via 1 on PELVIS
L_glut_max3 via 2 on L_THIGH
L_glut_max3 ins on L_THIGH

Muscle Parameters:

Max Isometric Force: 1000

Optimal Fiber Length: 0.1

Tendon Slack Length: 0.2

Pennation Angle: 0

Activation: 0.0001

Deactivation: 1

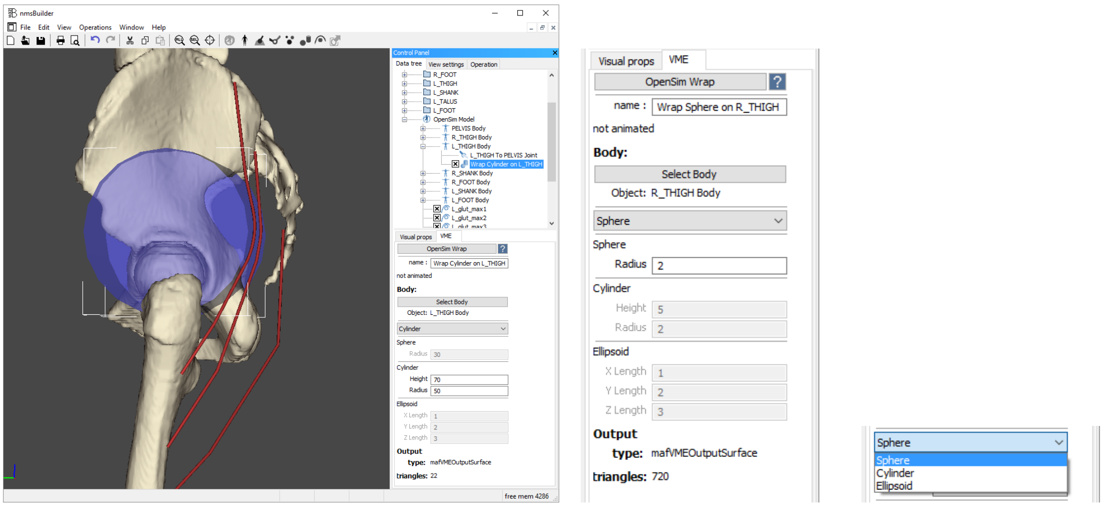
Output

type: mafVMEOutputPolyline
vtk vtkPolyData
Length: 226.52
Points: 4

Figure 6

5.1.17 OpenSim Wrap

OpenSim Wrap is a VME that contains an implicit parametric surface including the information for an OpenSim wrap object associated with a muscle path. In the VME tab, the user can specify the OpenSim body which the wrap object belongs to, by hitting the Select Body button, choose the type of wrap object from the drop-down list (sphere, cylinder and ellipsoid) and set the parameters by typing in the corresponding input fields (Figure 7).



(a)

name : Wrap Cylinder on L_THIGH
not animated
Body:
Object: L_THIGH Body
Cylinder
Sphere: Radius 30
Cylinder: Height 70, Radius 50
Ellipsoid: X Length 1, Y Length 2, Z Length 3
Output
type: mafVMEOutputSurface
triangles: 22

(b)

name : Wrap Sphere on R_THIGH
not animated
Body:
Object: R_THIGH Body
Sphere
Sphere: Radius 2
Cylinder: Height 5, Radius 2
Ellipsoid: X Length 1, Y Length 2, Z Length 3
Output
type: mafVMEOutputSurface
triangles: 720

(c)

Figure 7

5.2 Views

Views allow data examination. There are various views that permit to examine the VMEs. A View provides an interactive representation of the VME Tree. For each view the system maintains a display list of the VMEs that are currently rendered in that view. The Views available in NMSBUILDER are: Surface, Orthoslice, RX-CT, Arbitrary, DRR, Global Slice and Isosurface (Figure 8).

The user can control several properties of the active View through the View Settings tab in the upper part of the Control Panel.

5.3 Operations

Operations create new VMEs or modify existing ones. At any time the user can select a VME and then choose one of the available Operations. An Operation may accept only a particular type of VME as primary input. At any time, it is possible to run only those operations that accept the currently selected VME (Figure 9).

Operations that need more than one input may request additional VMEs to the user.

Operations are modal: when an operation is running, it is not possible to run another operation and it is not possible to change the selected VME.

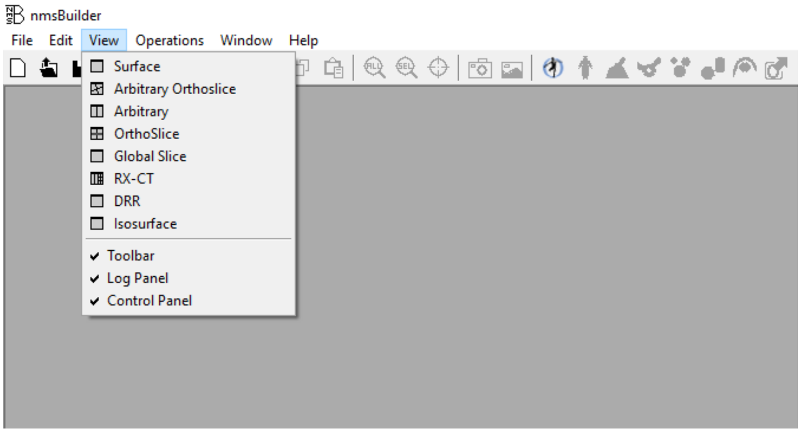


Figure 8

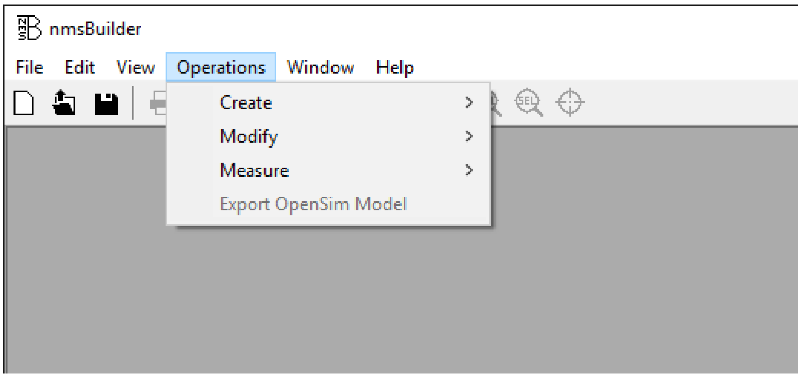
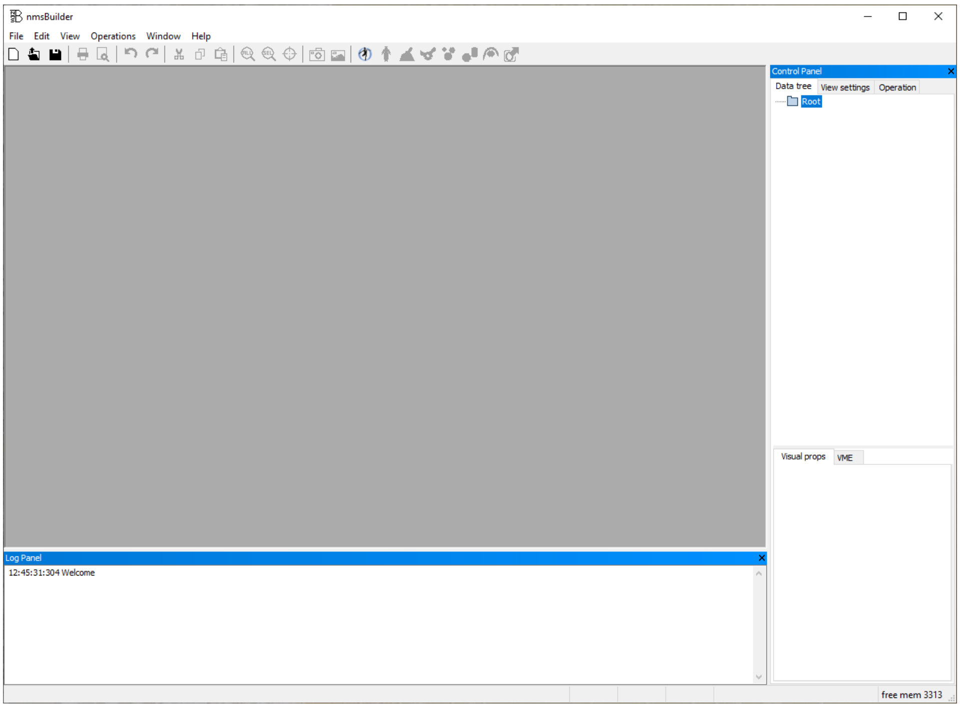


Figure 9

The user can control the behavior of the operation through the Control Panel. Operations support the Undo/Redo feature.

6 nmsBuilder basics: Graphical User Interface (GUI)

When launched the application interface looks like Figure 10.



```
12:45:31:304 Welcome
```

Figure 10

6.1 User Interaction

In all visualization windows the mouse interaction with the available objects happens as follows (see Fig. 11):

- **left mouse button** to *rotate*
- **shift + left mouse button** or **central mouse button** (or press down the scroll wheel) to *translate/pan*
- **right mouse button** or **scrool wheel** (only on Windows 10) to *zoom in/out*.

6.2 GUI environment

The basic structure of the GUI is composed of:

- Menu Bar

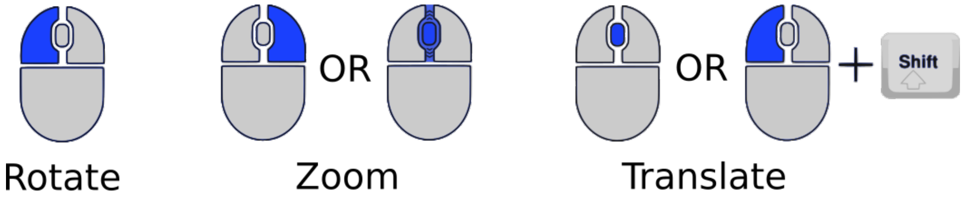


Figure 11

- Main Working Area
- Lateral Control Panel, showing the VME hierarchical structure
- Log Panel, for the system messages.

Contextual menus (upon clicking right mouse button) are also available for Views, VMEs, and for the VME tree.

6.2.1 Menu Bar

6.2.1.1 File This item contains all the commands related to input/output operations; in the application skeleton the basic features are open/save/new commands to respectively load, store, or initialize a new .nms session file. All importers/exporters will be described in detail in the corresponding sections.

6.2.1.2 Edit This item contains the commands to cut/copy/paste/delete any VME from the tree. Undo/redo commands are also available. Find VME allows the user to search a VME by name in the data tree. Each command has a keyboard shortcut, displayed in the menu. Finally, some settings of the application can be chosen (changes will take effect when the application restarts).

6.2.1.3 View This item contains the list of the available views. To add a new view, simply hit View and select the desired view. It is also possible to select which of the other bars (Toolbar, Log Panel, Control Panel) should be visible in the global application window. All views will be described in detail in the following.

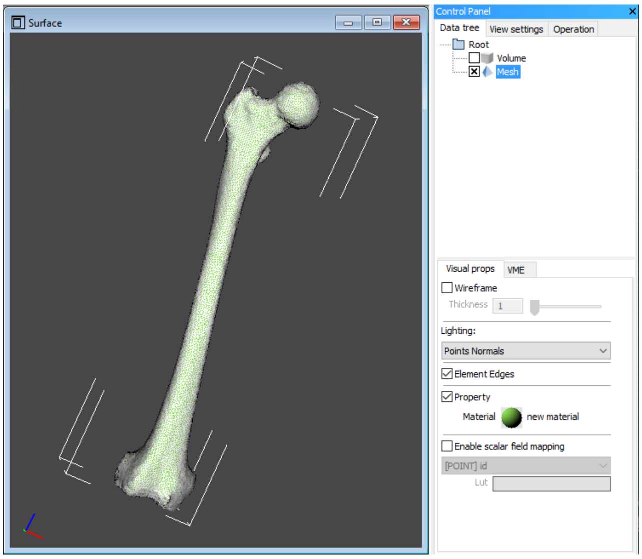
6.2.1.4 Operations This item contains a list of available operations within the application; to perform an operation, first select your input VME (if any), then select the desired Operation in Operations menu (if an operation can not be run with the selected VME as input, the operation name appears in grey). All operations will be described in detail in the corresponding sections.

6.2.2 Control Panel

The Control Panel is formed by three sub-windows:

6.2.2.1 Data Tree This sub-window shows the loaded VMEs, with their hierarchical structure. For the selected VME three other tabs are active (in the bottom part of the Control bar): it shows all VMEs loaded in the application, with their hierarchical structure. The user can select only one VME at a time. For the selected VME, two other tabs, shown below the tree, are active:

- *Visual props* shows the rendering properties of the selected VME according to the active view (Figure 12).



☐ Wireframe
Thickness 1
Lighting: Points Normals
☑ Element Edges
☑ Property
Material ● new material
☐ Enable scalar field mapping
[POINT] id
Lut

Figure 12

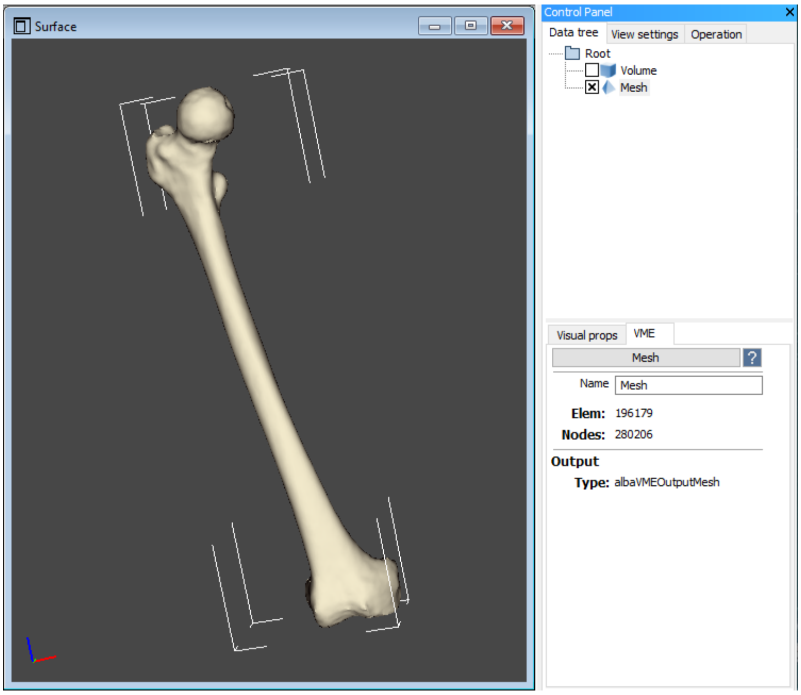
- *VME* shows the attributes of selected VME (such as the name, the encryption, the number of sub-entities such as cells or points it is compesed of, etc) (Figure 13).

Any VME is represented by:

- colour
  - Blue: the correspondent VME is loaded in memory and can be used
  - Grey: the VME cannot not be displayed in the active view
- icon
  Changes according to the type of VME

The display check box near the VME icon can be a square or a circle. If it is a circle, the view is mutually exclusive for that type of VME, which means that only one VME of that type can be display in the selected view.

6.2.2.2 View Settings This sub-window shows the settings of the active view. These can be consistently different depending on the type of active view. For example, camera parameters will be shown when a Surface view is active, slice or look-up table settings when an Orthoslice view is active. (Figure 14).



☐ Volume
☒ Mesh

Name: Mesh

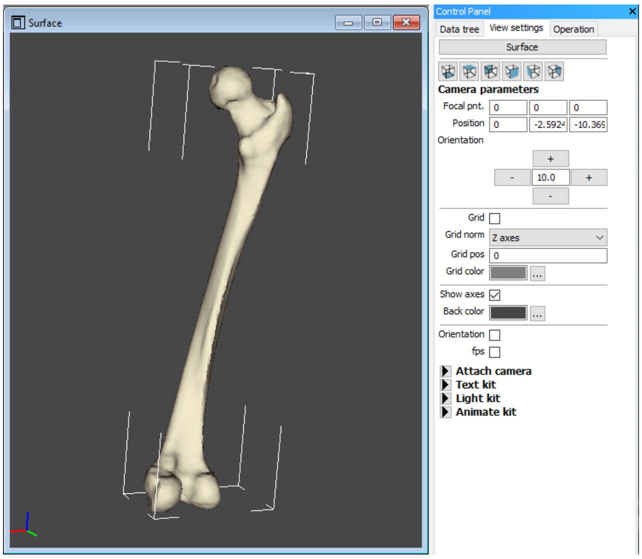
Elem: 196179

Nodes: 280206

Output

Type: albaVMEOutputMesh

Figure 13



Camera parameters

Focal pnt. 0 0 0

Position 0 -2.5924 -10.369

Orientation 10.0

Grid ☐

Grid norm Z axes

Grid pos 0

Show axes ☑

Orientation ☐

fps ☐

Figure 14

6.2.2.3 Operation

It shows the settings of the operation in progress.

6.2.3 Contextual Menus (i.e. mouse right-click)

6.2.3.1 View contextual menu

It can be activated pressing with right mouse button in any view background. It provides specific actions for the active view (Figure 15):

- *Rename view* allows to assign a specific name to the current view;
- *Normal size/maximize* allows to restore the size view to maximized or normal; in compound views this command is available for the view as a whole or for the selected sub panel;
- *Save as image* allows to save the content of the view in a standard image format (bmp, jpg, tiff, etc.); in compound views this command is available for the view as a whole or for the selected sub panel;
- *Export as VRML* allows to save the content of the rendered view in VRML format.

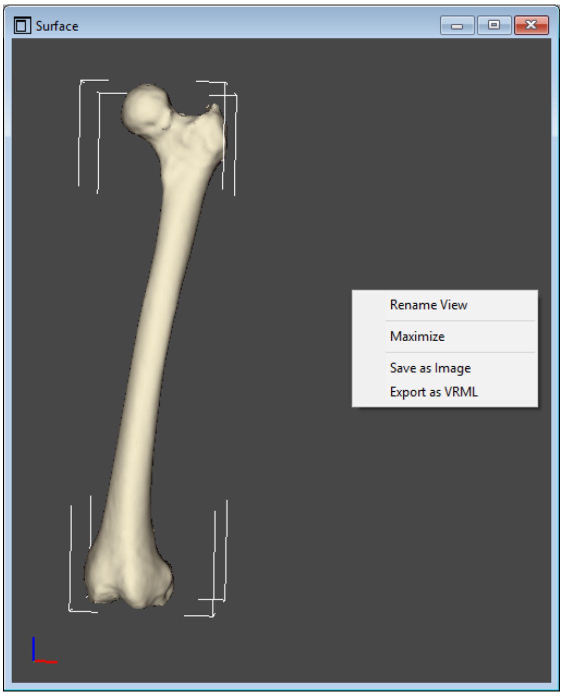


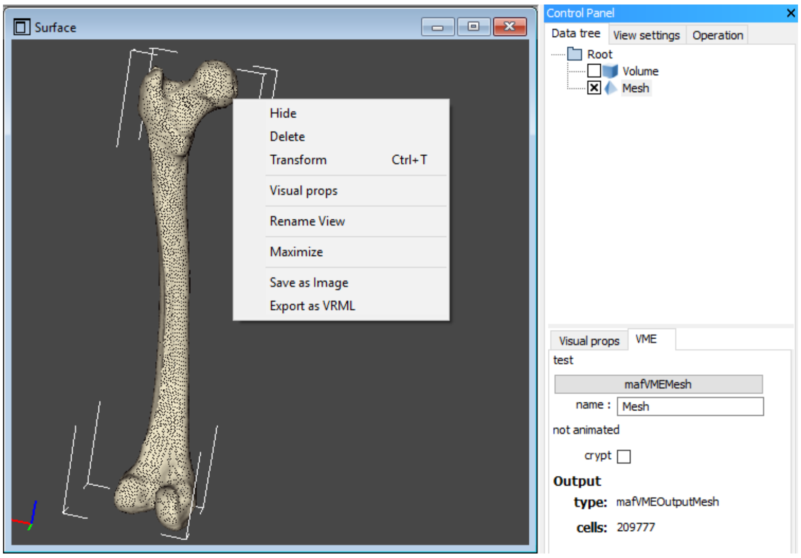
Figure 15

6.2.3.2 VME contextual menu

It can be activated pressing with right mouse button on any VME displayed in any view (provided it is also the one selected in the VME tree) (Figure 16). It adds to the actions of the View contextual menu, some other actions that are specific for the VMEs:

- *Hide* removes from the display list the selected VME;

- *Delete* deletes from the Data Tree the selected VME;
- *Transform* launches the Transform operation on the selected VME.
- *Visual props* opens a GUI exposing the visual properties of the VME according to the selected View, or (in compound views) the view sub-panel.



test

mafVMEMesh

name : Mesh

not animated

crypt □

Output

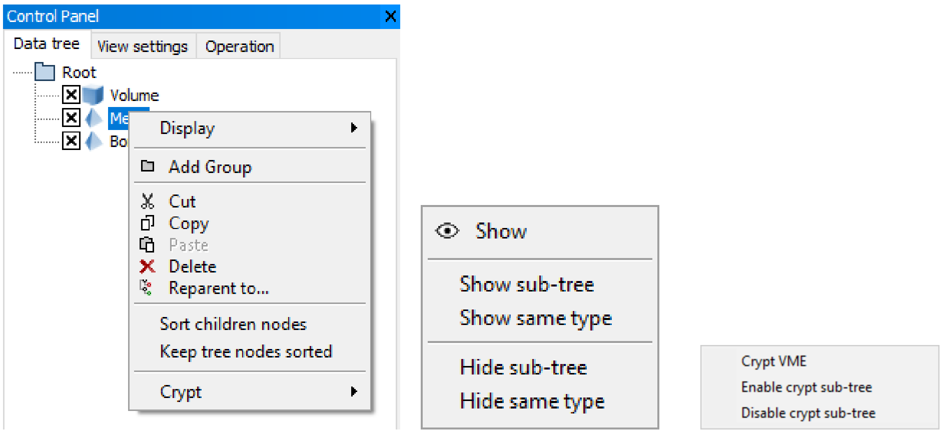
type: mafVMEOutputMesh

cells: 209777

Figure 16

6.2.3.3 Tree contextual menu It becomes active upon a right mouse click, on any VME in the VME tree (Figure 17). It provides the following functionalities:

- *Display (present if a view is opened)* This sub menu (Figure 17b) provides the following functionalities used to show/hide VMEs:
  - *Show/Hide* show or hide the current VME
  - *Show sub-tree (present if a view is opened)* displays in the current view all the VMEs of the sub-tree related to the selected VME;
  - *Show same type* it is present if the selected VME can be displayed in the active view. It displays in the current view all the VMEs of the same type of the selected one;
  - *Hide sub-tree* removes from visualization in the active view all the VMEs of the sub-tree of the selected VME;
  - *Hide same type* removes from visualization in the current view all the VMEs of the same type of the selected one;
- *Add Group* Adds a Group (See Par. 5.1.2) to the tree, as a son of the selected VME.



(a) (b) (c)

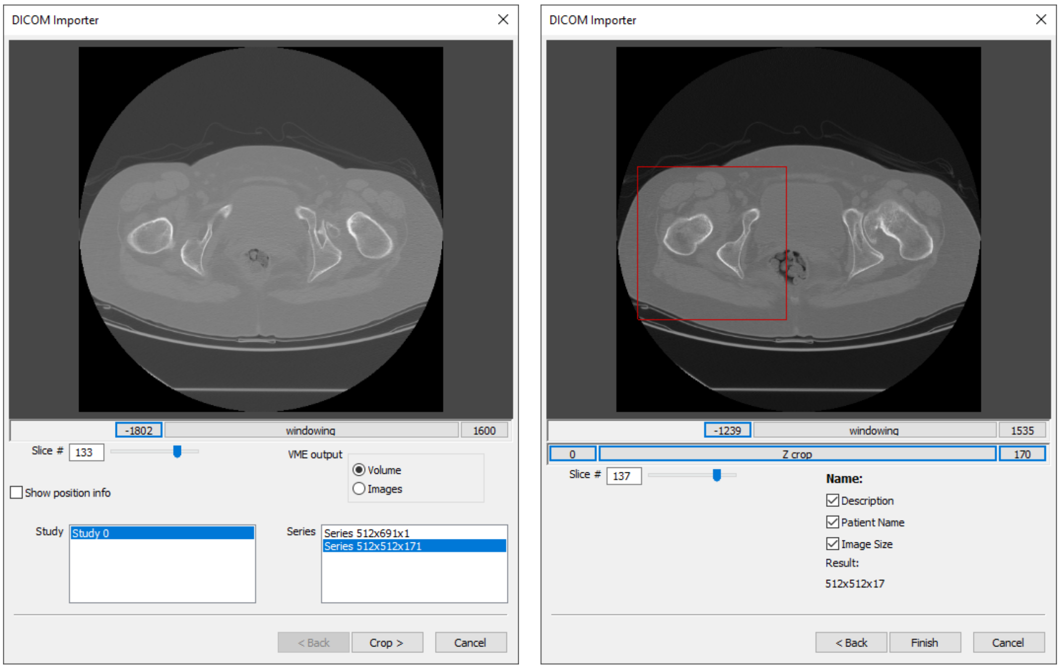
Figure 17

- *Cut* Cuts the current VME (and the relative subtree); the VME is removed from the tree and placed in the clipboard.
- *Copy* Copies the current VME (and the relative subtree); the VME is placed in the clipboard, but not removed from the tree.
- *Paste* Pastes the content of the clipboard, a a son of the selected VME.
- *Delete* Deletes the current VME (and the relative subtree).
- *Reparent to...* Reparents the selected VME to another VME, without changing its absolute pose
- *Sort children nodes* Sorts children nodes in alphabetical order.
- *Keep tree nodes sorted* Keeps the alphabetical sorting even when new VMEs are added to the data tree.
- *Crypt* This sub menu (Figure 17c) provides the functionalities for VME encrypting:
  - *Crypt* Applies encryption to the selected VME;
  - *Enable crypt sub-tree* Applies encryption to all VMEs of a sub-tree.
  - *Disable crypt sub-tree* Removes encryption from all VMEs of a sub-tree.

7 nmsBuilder Key Commands

7.1 Data Importers and Exporters (menu File)

The Import and Export classes, accessible through the menu File, allow the user to import/export data to/from the application, in different formats. When in use in nmsBuilder, the files are stored in the internal nms format.



(a) (b)

Figure 18

Importers/Exporters are grouped according to the type of data they are dealing with. In nmsBuilder, the user can import medical images, surfaces, landmarks or previously stored vtk files. Imported entities can also be exported.

7.1.1 Import Dicom

Path: File → Import → Images → DICOM

This importer loads medical images stored in DICOM format, and creates as output either a Volume or Image VME (most times, a Volume VME).

The user has first to select, from a standard window for folder selection, the directory in which the Dicom files are stored. Once the folder has been selected, a new wizard window appears in which the user can subsequently:

1. see the images preview. In this windows you can
   - (a) perform windowing on the images
   - (b) select the output type
   - (c) show/hide slice position and orientation info
   - (d) choose the number of study id if multiple ones were present in the folder
   - (e) choose the number of series id if multiple ones were present in the selected study

2. proceed to the optional “Crop” stage, in which cropping is performed in the slices plane by drawing with the mouse the crop area, and in the direction orthogonal to the slices by moving the “Z crop” slider. The in-plane crop area can be changed by moving each side of the area, and controlled over the different slices with the “Slice #” slider (Figure 18b). In this stage you can compose the VME name output by enabling “Patient Name” and “Image Size” check boxes and complete the process by clicking “Finish”.

7.1.2 Import Raw Volume

Path: File → Import → Images → Raw Volume

This importer can load a raw binary volume file, i.e. a stack of images already stored in a single file (Figure 19).

A Volume is created in the data tree as a result. The user needs to choose the file name (browsing with the “open” button), specify the endianity, the scalar type (char, short, int, float, double) through which the images are represented, the number of components for each pixel value (e.g. 1 in a greyscale image, 3 in a RGB Volume), and to specify if data are signed or not (check-box). The user has also to enter the dataset dimensions (number of pixels) and the spacing in mm/pixel in all directions. The import can be limited to a part of the original volume, defining a volume of interest (VOI) through x, y and z pixel intervals (VOI x, VOI y, VOI z fields). If the spacing along the z-spacing axis is not uniform the user can specify a text file from which the Z-coordinates are read (browsing with “z-coord load” button). The Z coordinates file should be a text file whose format is like the following (the coordinates should be written in ascending order):

```
Z coordinates:
-109.1
-107.8
-104.0
...
```

Moreover, the user can set the header size to be skipped in the import process. A “guess header” function is available through the “guess” button. Finally, a Volume is created in the tree upon pressing the ok button.

7.1.3 Import STL

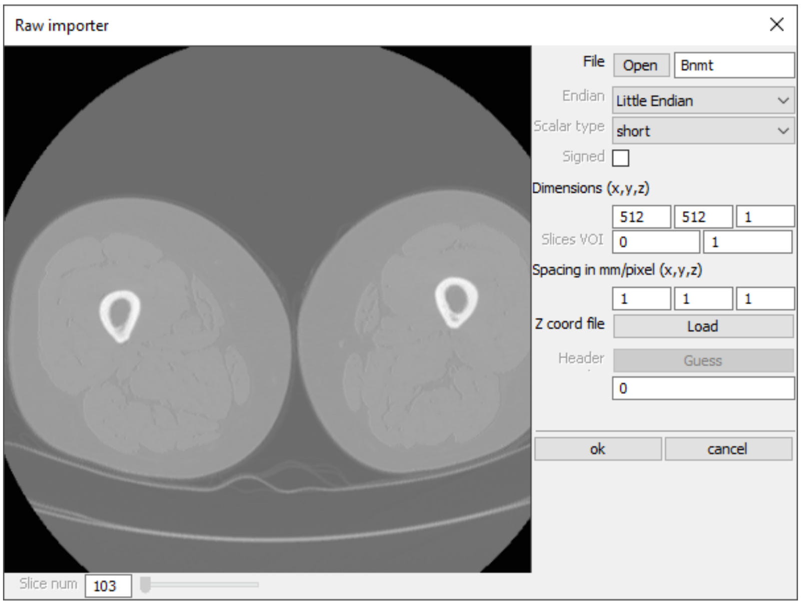
Path: File → Import → Surfaces →STL

It imports a STereoLithography (STL) file, which contains a polygonal surface model. A Surface VME is created in the data tree.

7.1.4 Import VTP

Path: File → Import → Surfaces →VTP (from OpenSim)

It imports a 3D geometry in vtp file format, which contains a VTK XML-based description of a polygonal surface model. A Surface VME is created in the data tree.



File: Bnmt
Endian: Little Endian
Scalar type: short
Signed: □
Dimensions (x,y,z): 512 512 1
Slices VOI: 0 1
Spacing in mm/pixel (x,y,z): 1 1 1
Z coord file
Header: 0
Slice num: 103

Figure 19

7.1.5 Import VTK

Path: File → Import → VTK

The VTK importer loads data stored in vtk format. The type of VME created into the data tree depends on the information contained into the vtk file.

7.1.6 Import Landmark

Path: File → Import → Landmark

It imports a landmark cloud from an ASCII file. The imported landmarks are stored in the data tree in a Landmark Cloud VME with the same name of the imported file, which represents a landmark cloud where each row contains the landmark coordinates.

When the operation is launched, the user can choose the separator (comma, space, semicolon, tab) and if the file to be imported should be tagged (i.e. contains the names of the landmarks) or not (picking "Coordinates only" is for not tagged files). For example, a tagged file with comma as separator should be formatted as follows:

```
RASIS, 108.513, -116.382, -178.429
LASIS, -92.9863, -121.055, -183.088
RPSIS, 41.1026, -230.853, -146.008
LPSIS, -37.8802, -231.691, -144.048
SACRUM, 1.13252, -235.337, -148.725
```

If the file is not "tagged", the expected ASCII file should not include the text fields, but only the coordinates of each point.

7.1.7 Export STL

Path: File → Export → Surfaces → VTK

It exports a Surface VME in STereoLithography (STL) file format (both binary and ASCII).

7.1.8 Export VTP

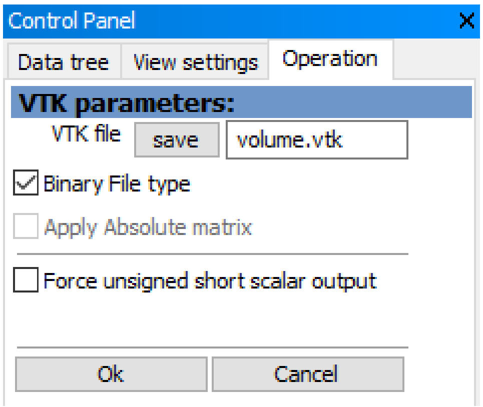
Path: File → Export → Surfaces → VTP (for OpenSim)

It exports a Surface VME in vtp file format, which contains a VTK XML-based description of a polygonal surface model.

7.1.9 Export VTK

Path: File → Export → VTK

Each VME can be exported as its vtk representation. The user can choose whether to export the binary or the ASCII file (Figure 20).



VTK parameters:

VTK file: volume.vtk

☑ Binary File type

□ Apply Absolute matrix

□ Force unsigned short scalar output

Figure 20

7.1.10 Export Muscle

Path: File → Export → Muscle

It exports an OpenSim Muscle VME in ASCII file. The default output file contains muscle parameters and muscle path information. The user can choose whether to include or not muscle parameters by checking or unchecking the corresponding box in the Export panel, as well as name and directory of the file.

7.1.11 Export Landmark

Path: File → Export → Landmark

It stores a Landmark Cloud VME in ASCII file. The output format contains, in each row, the landmark name followed by its coordinates, using comma as separator, as in the example of Par. 7.1.6.

7.2 Views (Menu View)

Every VME can be visualized and managed through a combination of views and operations. Many instances of the same view with different VMEs displayed and/or different views can be opened at the same time according to the user's needs, and to memory usage limits of the computer. Views can be single, or composed by different sub-views (visualization panels), (Figure 21). Each visualization panel views can be (i) perspective, i.e visualizing a VME in space through a camera that can be roto-translated or zoomed, or (ii) slice, i.e. slicing a VME through a defined plane (Figure 21).

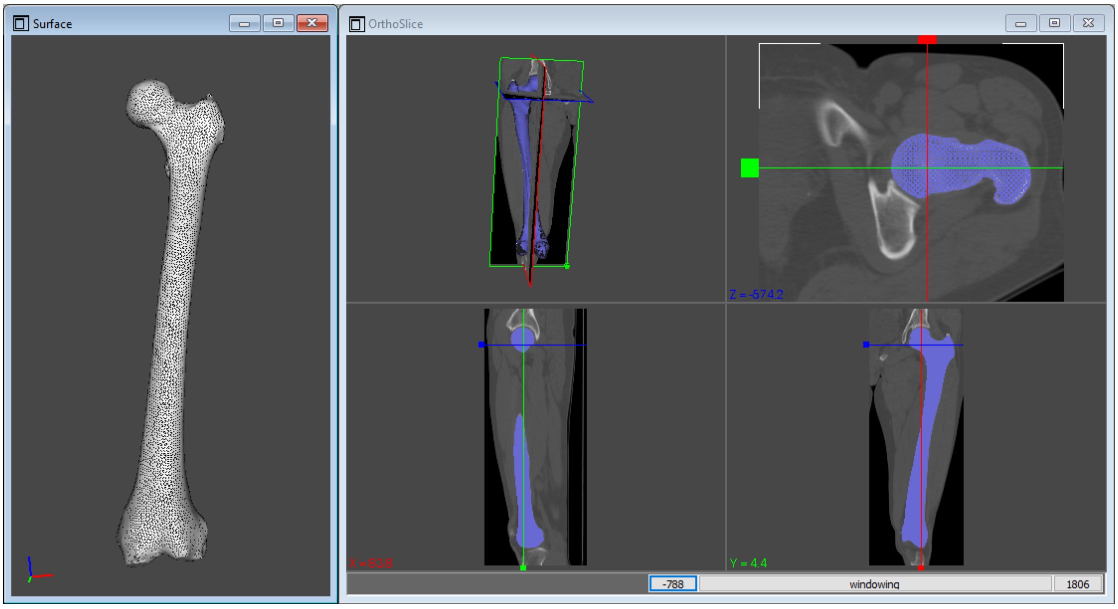


Figure 21

Special interaction objects (Gizmos, i.e. particular handles instantiated by a View or an Operation to allow direct interaction with the mouse) are present in the slice views to facilitate the definition and the fine tuning of the view settings (Figure 22).

7.2.1 View Surface

The Surface View is a three-dimensional perspective render window (Figure 23).

The Surface View supports in nmsBuilder the visualization of:

- Volumes, displayed only with their bounding box.
- Surface ("Visual props" panel of the control bar, below the VME tree, or "Visual props" in the contextual menu). For further explanation see Surface visual properties in the paragraph 7.2.4.

The view properties can be changed from the "View settings" panel (Figure 24):

- "camera parameters": the user can control the camera with text entries rather than with the mouse, if a predefined exact camera position has to be set;

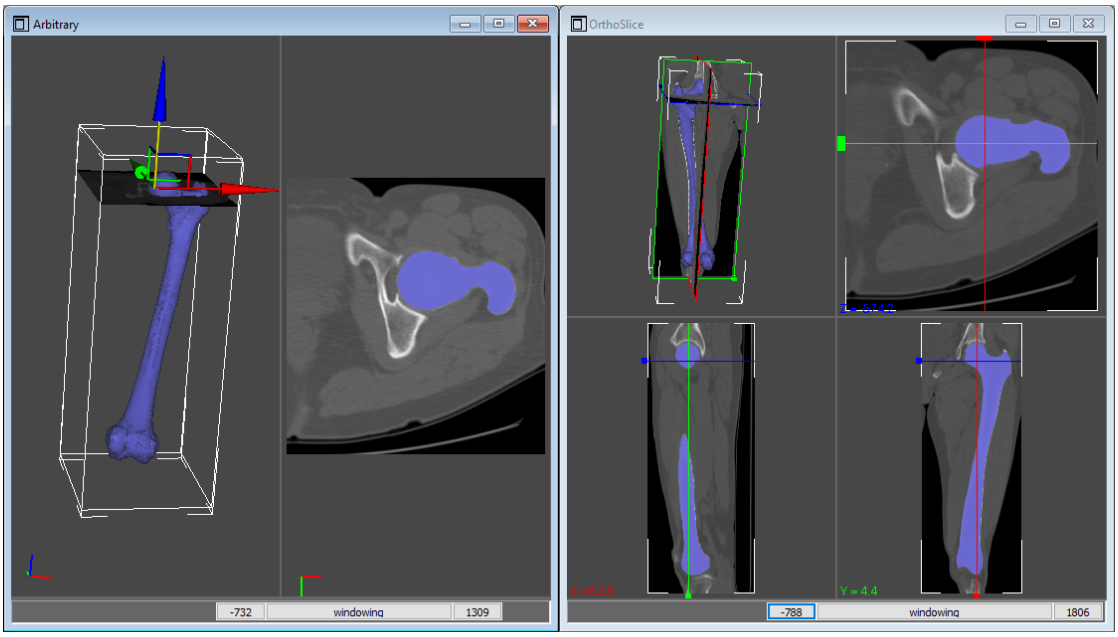


Figure 22

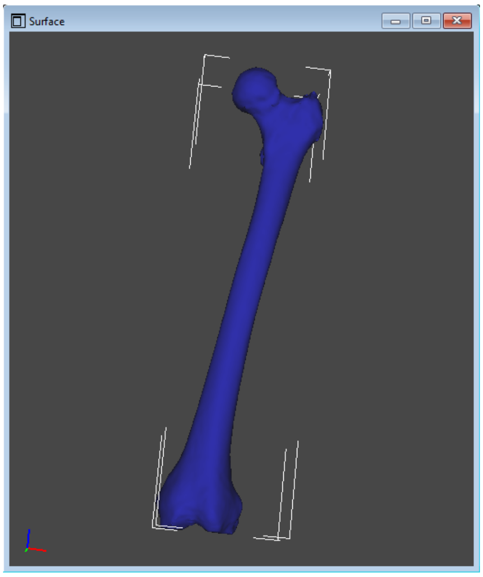
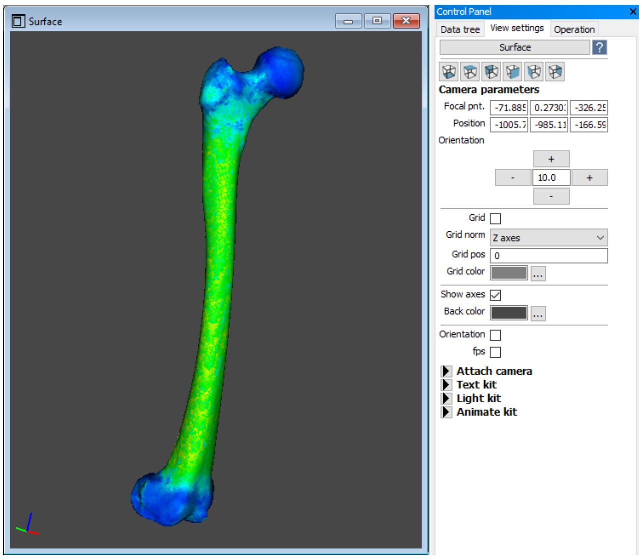


Figure 23

- “grid” commands make it possible to visualize the grid or the axes and modify their colours
- “show axes” shows or hides the reference system gizmo
- “back color” lets the user set the background colour. The other settings below are part of Surface View in ALBA but are of little use in NMSBUILDER.



Focal pnt. -71.885 0.2730 -326.25
Position -1005.7 -985.11 -166.59
10.0
Grid □
Grid norm Z axes
Grid pos 0
Grid color ...
Show axes ☑
Back color ...
Orientation □
fps □

Figure 24

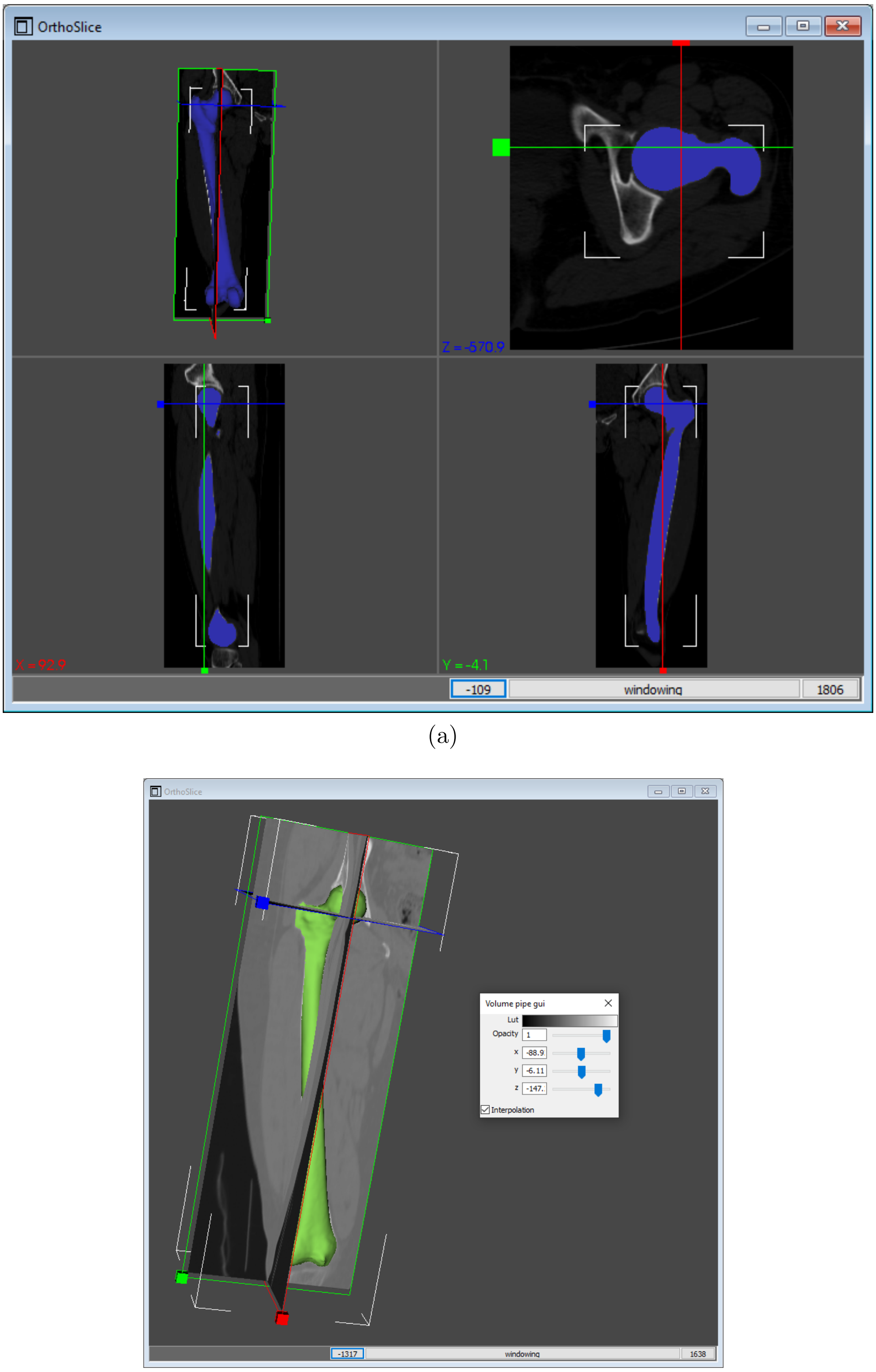
7.2.2 View OrthoSlice

The OrthoSlice View is a compound view made of four panels and three gizmos. A perspective panel shows a volume intersected by the three orthogonal planes and the other three panels show a parallel view of each other slice (Figure 25a). The Orthoslice View supports the visualization of:

- Volume, one at a time. Windowing for the volume scalar is available. It applies to all view panels.
  Volume visual properties (“Visual props” panel accessible through contextual menu): the user can change or define the LUT of the volume scalars, and its opacity in the visualization; the x,y,z quotes can also be changed manually (Figure 25b)
- Surface, even more than one simultaneously, with a 3D rendering in the perspective view panel, and the sliced contour in the orthogonal slice panels.
  Surface Visual properties: for further explanation look at the paragraph 7.2.4.

The view properties can be changed from the “View settings” panel (Figure 26):

i. “layout”: the user can change the layout of the panels in the view.



Lut

Opacity 1

x -88.9

y -6.11

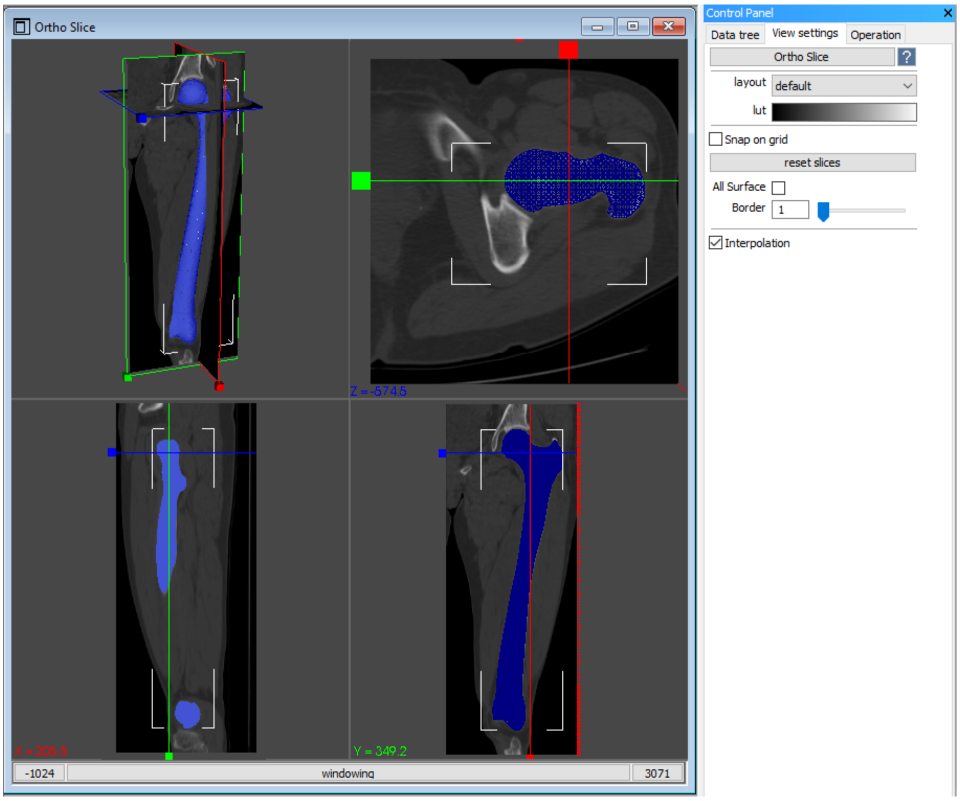
z -147.

☑ Interpolation

(a)

(b)

Figure 25



layout: default

lut

□ Snap on grid

All Surface □

Border 1

☑ Interpolation

Figure 26

ii. “LUT”: the user can set or define the look-up table to be assigned to the scalars of the Volume currently visualized.

iii. “Snap on grid”: the user can select whether to avoid interpolation between slices and perform slice cuts only at values of the original grid.

7.2.3 View RX-CT

The RXCT View is a compound view made of eight 2D panels: two planar projections in the XZ and YZ planes, and six slices in the XY plane, whose Z-quotes are identified in the two planar projections by six gizmos (Figure 27).

The RX-CT View supports the visualization of:

- Volume (one at a time), displayed in the projection panels as a digitally reconstructed radiograph, usually resembling anatomical planes projections (e.g. Anterior/Posterior and Medial/Lateral), and sliced along the Z direction in the slice panels.
  Volume visual properties (“Visual props” panel accessible through contextual menu): the user can change or define the LUT of the volume scalars, and its opacity in the visualization; the z quote can also be changed manually (Figure 28).

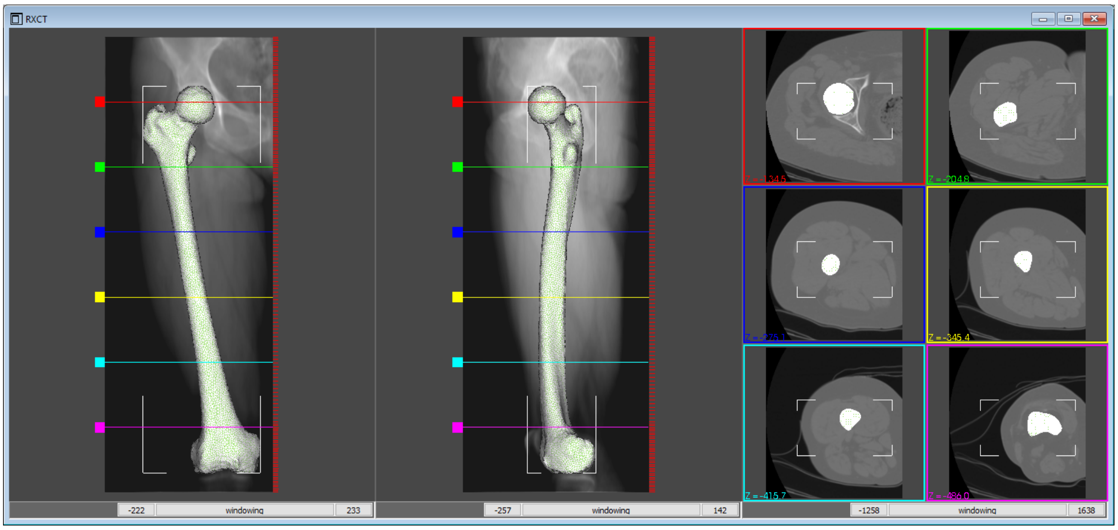
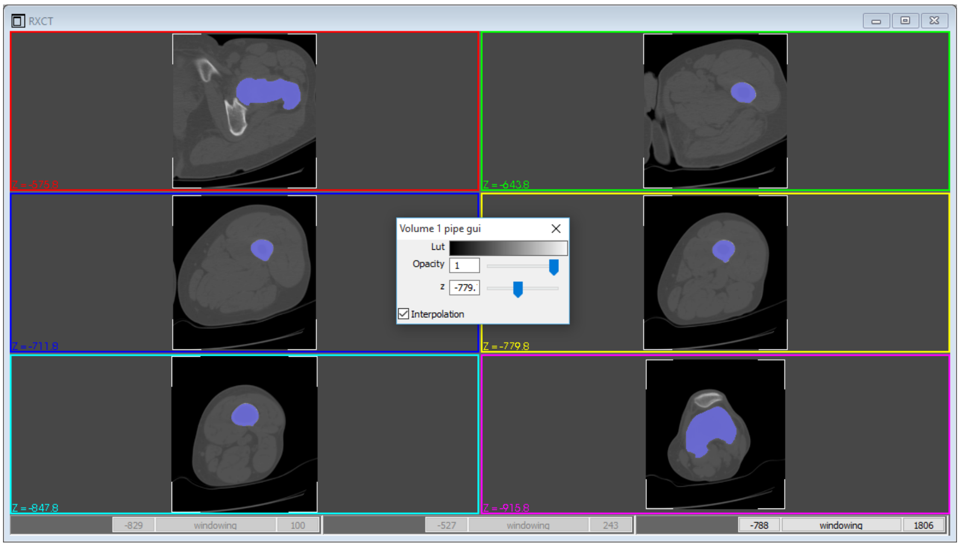


Figure 27



Volume 1 pipe gui

Lut

Opacity 1

z -779.

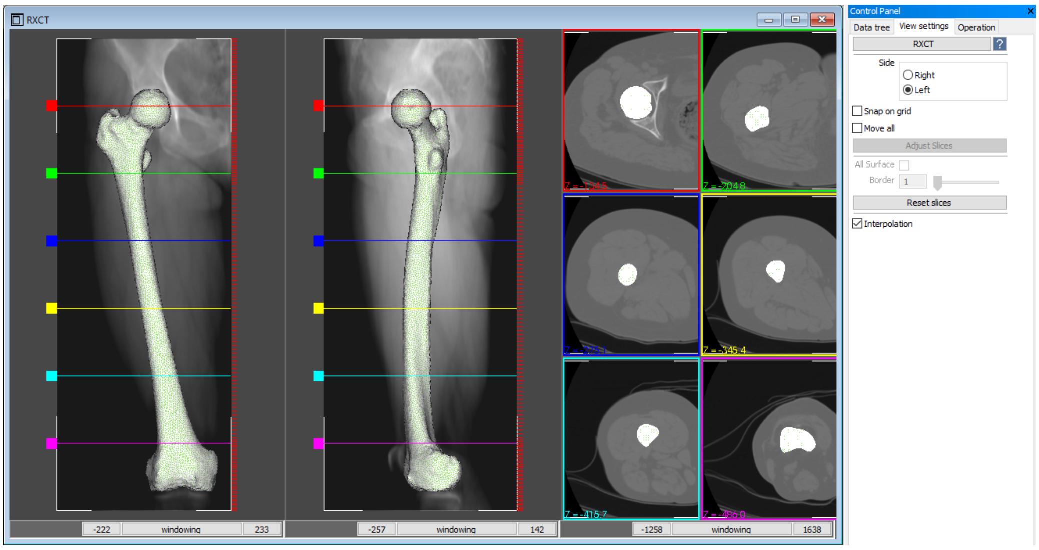
☑ Interpolation

Figure 28

- Surface(even more than one simultaneously), as a 3D rendering in the projection panels and as sliced contours in the slice panels.
  Surface Visual properties: for further explanation look at the paragraph 7.2.4.

The view properties can be changed from the “View setting” panel (Figure 29):

- “Side” (Left/right) changes the projection direction of the lateral projection
- “Snap on grid”: the user can select whether to avoid interpolation between slices and perform slice cuts only at values of the original grid. The user can change the layout of the panels in the view
- “Move all”: by checking it, the user moves all gizmos simultaneously
- “reset slices” brings back all slices to their original positions



Side ○ Right ● Left

□ Snap on grid

□ Move all

All Surface □

Border 1

☑ Interpolation

Figure 29

7.2.4 View Arbitrary

The Arbitrary View is a compound view made of two panels. In the left panel, the user can change using gizmos the position and the orientation of an arbitrary plane; in the right panel the corresponding volume section is showed in a 2D parallel (slice) view (Figure 30).

The Arbitrary View supports the visualization of:

- Volume, one at a time. Windowing for the volume scalar is available. It applies to both view panels.

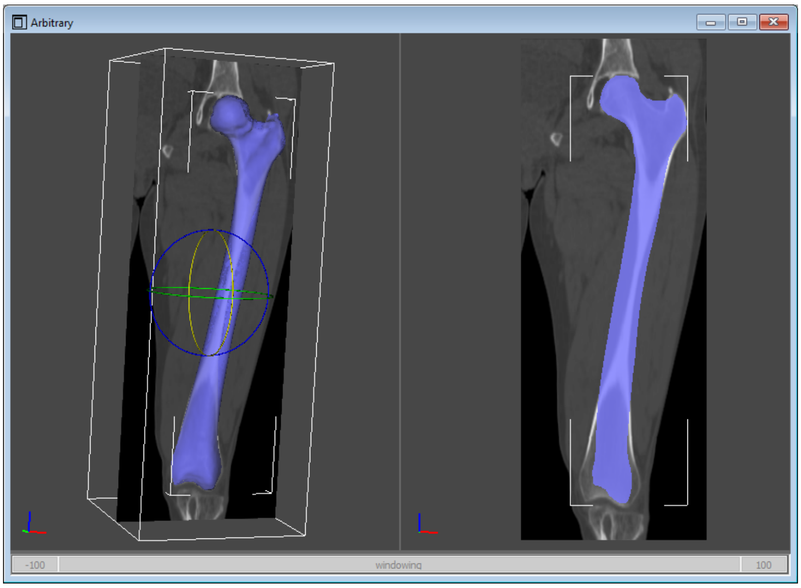


Figure 30

- Surface , even more than one simultaneously, with a 3D rendering in the left volumetric view panel, and the sliced contour in the right slice view panel.
  Surface Visual properties (“Visual props” panel accessible through contextual menu): in this panel, the user manages the visualization options of the selected Surface . The following command/options are available (Figure 32).

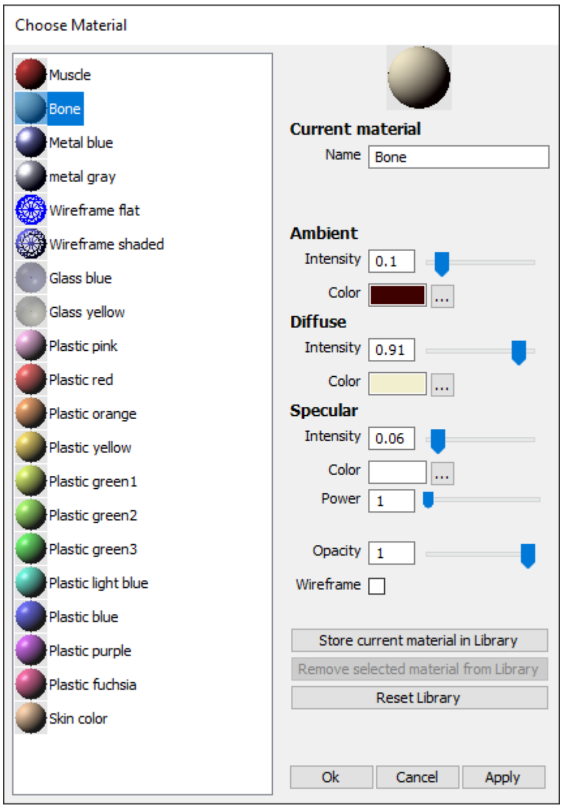
  - Wireframe: this checkbox enables wireframe visualization. The user can adjust the edges thickness, by editing the number or acting on the dedicated slider.
  - Lighting: from the drop down menu, the user can change the way normals are calculated (on the cells or on the vertices). “Points Normals” (default) gives a smoothed visualization, while “Cells Normals” shows discontinuities at edges.
  - Element edges: this checkbox enables the visualization of the edges.
  - Property: this checkbox applies several rendering properties, called “Material” for brevity. Under the checkbox there is a circle with the material preview. Upon clicking on it, another window is opened (Figure 31), in which it is possible to:

    - Choose a material from a library
    - Modify the current material
    - Store or remove the current material from the library
    - Reset the library at the default list

    The changes at the current material are saved in the list only by clicking “Store current material in Library” but the material appearance is automatically applied by clicking “OK”. It is possible to modify the following material properties:

  - ∗ Name of the material
  - ∗ Ambient color and intensity
  - ∗ Diffuse color and intensity
  - ∗ Specular color, intensity and power
  - ∗ Opacity
  - ∗ Wireframe

– Enable scalar field mapping: this checkbox is active if the selected Surface has one or more scalar fields associated to its vertices/cells. If checked, it enables the mapping of a scalar field. The user can select from the drop down menu the scalar field to visualize. Then, clicking on the Lut (Look-up table) bar a window opens, letting the user choose/modify the contouring color map, value range, number of contours, and color/value associations.



Choose Material

Current material

Name: Bone

Ambient
Intensity: 0.1
Color

Diffuse
Intensity: 0.91
Color

Specular
Intensity: 0.06
Color
Power: 1

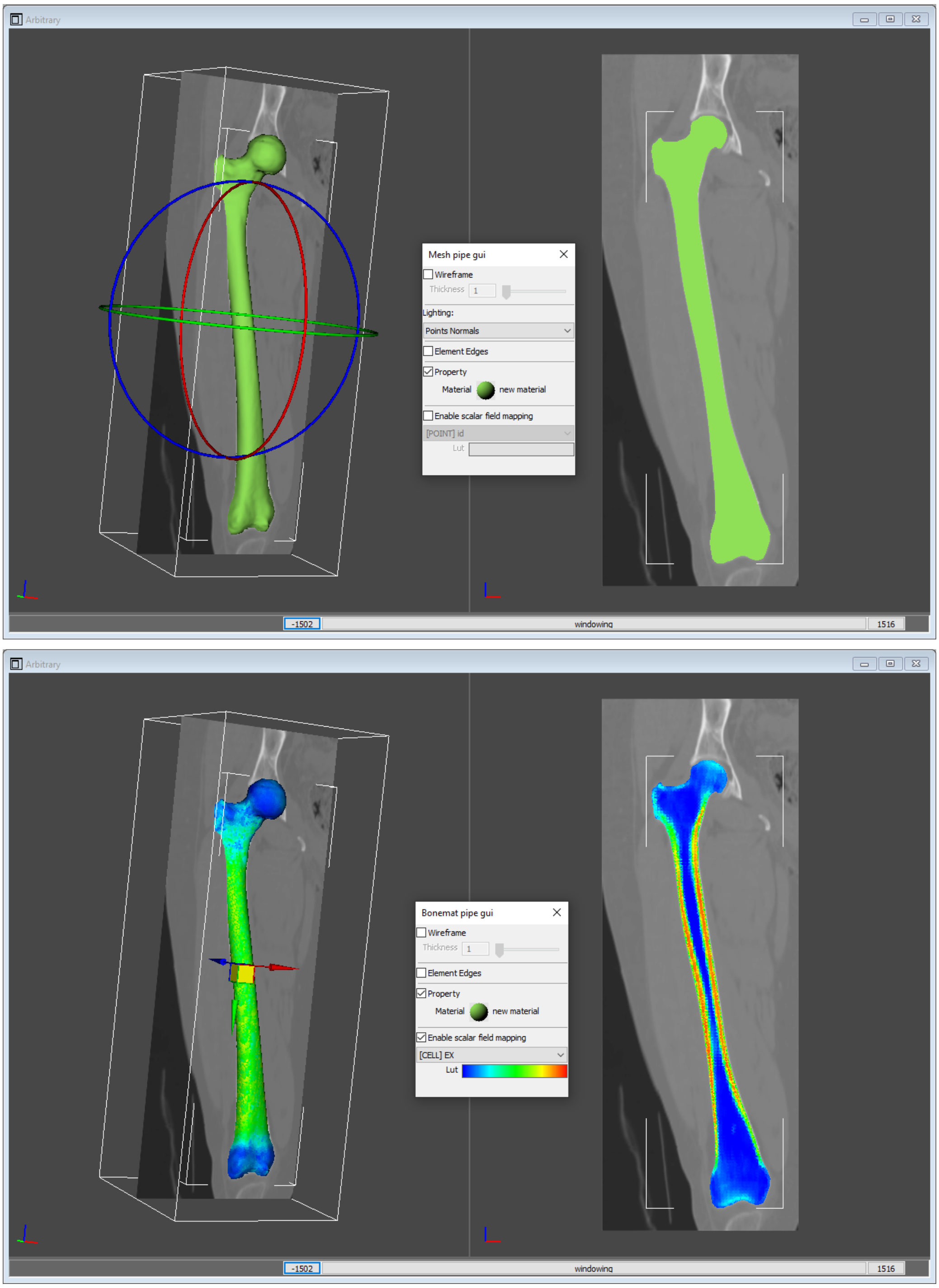
Opacity: 1

Wireframe: □

Figure 31

The view properties can be changed from the "View settings" panel (Figure 33):

i. The user can change the position and orientation of the visualized slice through Gizmo or Text interaction.
"Gizmo Interaction": it is possible to translate or rotate the slice with three arrows (for the translation) or three rings (for the rotations). By selecting the desired



Mesh pipe gui

☐ Wireframe

Thickness 1

Lighting:

Points Normals

☐ Element Edges

☑ Property

Material new material

☐ Enable scalar field mapping

[POINT] id

Lut

Bonemat pipe gui

☐ Wireframe

Thickness 1

☐ Element Edges

☑ Property

Material new material

☑ Enable scalar field mapping

[CELL] EX

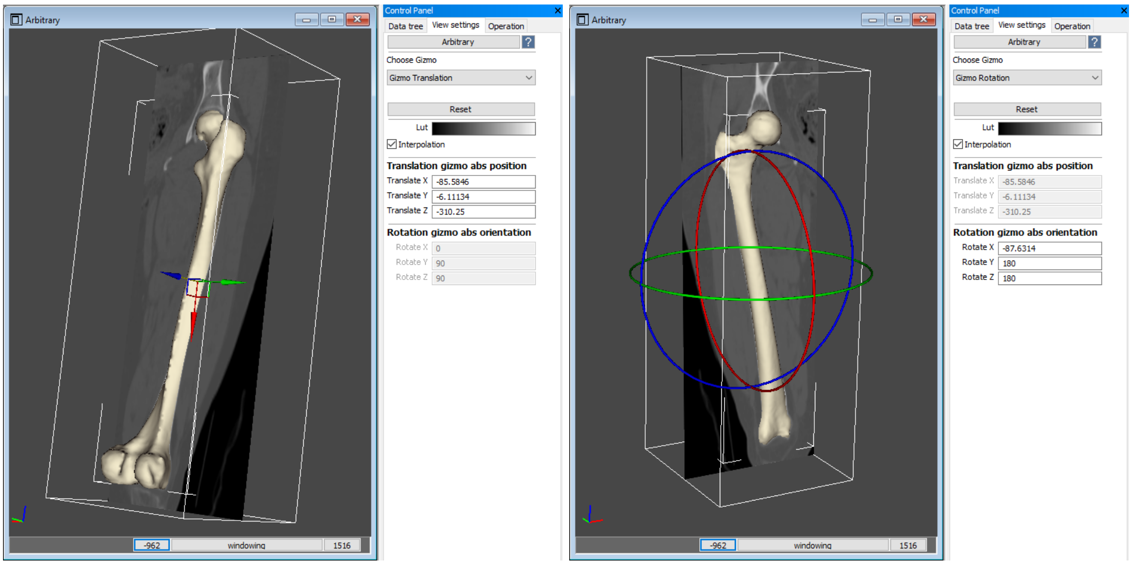
Lut

Figure 32

gizmo and holding pressed the left button of mouse, the gizmo moves and the plane is updated accordingly.
"Text Interaction": The user can write the value in the corresponding text box, and restore the initial plane position with the "reset" button.

ii. "Reset" button: the user can reset the initial position of the slice

iii. "LUT": the user can set or define the look-up table to be assigned to the scalars of the Volume currently visualized.



Choose Gizmo: Gizmo Translation
Lut
☑ Interpolation
Translation gizmo abs position
Translate X -85.5846
Translate Y -6.11134
Translate Z -310.25
Rotation gizmo abs orientation
Rotate X 0
Rotate Y 90
Rotate Z 90
-962 windowing 1516

Choose Gizmo: Gizmo Rotation
Lut
☑ Interpolation
Translation gizmo abs position
Translate X -85.5846
Translate Y -6.11134
Translate Z -310.25
Rotation gizmo abs orientation
Rotate X -87.6314
Rotate Y 180
Rotate Z 180
-962 windowing 1516

Figure 33

7.2.5 View Arbitrary Orthoslice

The OrthoSlice View is a compound view made of four panels and three gizmos (Figure 34). It is intended as a union of the functionalities of the arbitrary view and the orthoslice view, and supports the visualization of volumes, surfaces and landmarks.

There is the perspective panel showing a volume intersected by the three orthogonal planes and the other three panels showing a parallel view of the arbitrary slice. The gizmos on the orthogonal panels allow to rotate and translate the arbitrary slice. Specifically, clicking on the plane boxes will allow to translate the selected plane (Fig 35 top left and right), clicking inside the boxes will allow to move both planes (Fig 35 bottom left), and clicking outside the boxes will allow to rotate the selected plane (Fig 35 bottom right)

7.2.6 DRR

The Digitally Recontructed Radiography (DRR) View is a special volume renderer window that displays a Volume VME as the result of the DRR algorithm, while a Surface VME is rendered with a 3D representation.

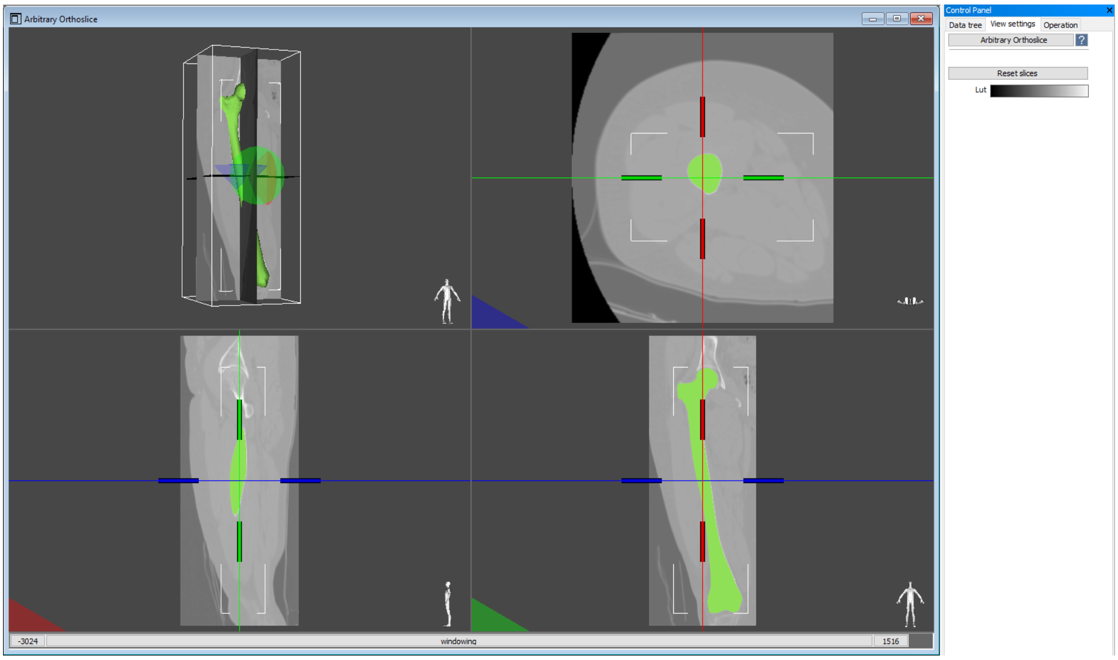


Figure 34

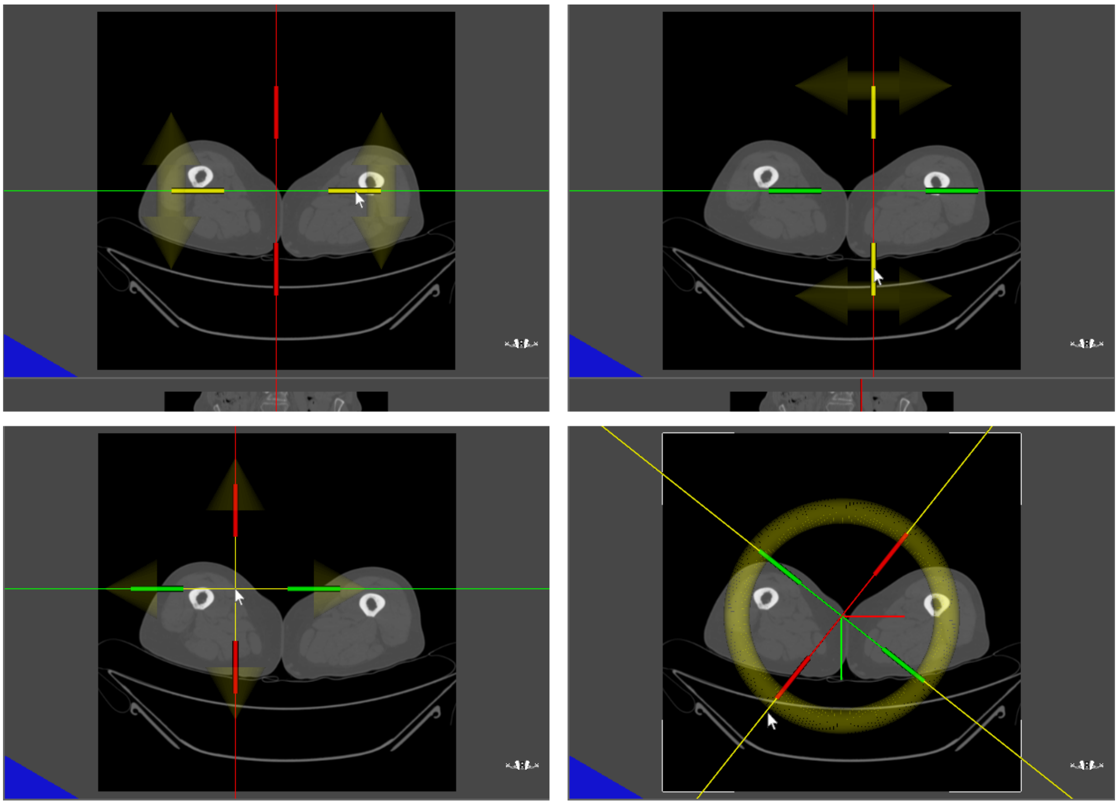


Figure 35

In the visual props panel, the DRR settings can be used to change the generated DRR (e.g.. colour, or the level of the project with the gamma value). The camera settings can be used to modify the position of the projection instead of using the mouse interaction.

7.2.7 Global Slice

The Global Slice View shows a slice obtained from the intersection of one or more volumes with a user selectable coordinate plane.

In the View Settings it is possible to choose the orientation of the plane with respect to the coordinate system, select the position of the plane in the chosen direction, change the transparency with the opacity slider or the LUT of the selected volume. An indication of the slicing plane position (origin and director cosines) is reported in the bottom left corner of the view.

7.2.8 Isosurface

In the Isosurface View, a Volume VME is displayed as a surface extracted with an iso-density contour filter. The user can interactively change the density threshold (contour) and the transparency threshold (alpha) in the visual props panel.

7.3 Operations (menu Operations)

7.3.1 Create

7.3.1.1 OpenSim This set of operations allows the creation of OpenSim models, by creating different VMEs that store the data necessary to define OpenSim objects The operations that can be launched depend on the data that the user have created in the data tree.

7.3.1.1.1 OpenSim model This operation creates a VME that contains all OpenSim objects. This operation is preliminary to any other OpenSim operation.

7.3.1.1.2 Body This operation creates a VME that contains the data to create an OpenSim body: the geometry and the density. The user can include a single-surface body or a multiple-surface body (i.e., a surface group made of two or more VMEs). When launched, the user has to select the surface VME or surface group from the data tree, and manually enter the value for the densities in the VME tab of the control panel, which are used to calculate the inertial parameters in the export phase.

7.3.1.1.3 Free Joint with Ground This operation creates a VME that contains the data to create a free joint (6 degrees of freedom) between any selectable Child Body in the VME tab of the control panel and the Ground.

7.3.1.1.4 Joint This operation creates a VME that contains the data to create different types of joints (Pin, Ball or Custom) between two bodies. When launched, the user has to select the type of joint, the parent body and the corresponding reference system defining the joint in the parent body (Ref. Sys. in Parent), and the child body and the corresponding reference system defining the joint in the child body (Ref. Sys. in Child)

7.3.1.1.5 Marker Set This operation creates a VME that contains the data to create a marker set on a body. When launched, the user has to select the marker set body and the landmark cloud.

7.3.1.1.6 Wrap Object This operation creates a parametric VME that contains the data to create a wrap object in OpenSim, which belongs to a body and can be used to wrap muscle paths. When launched, the user has to select the body, the type of object (sphere, cylinder, ellipsoid), and can edit the geometrical parameters of the object in the VME tab of the control panel.

7.3.1.1.7 Muscle This operation creates a parametric VME that contains the data to create a muscle object in OpenSim. When launched, the user has first to define the muscle path by selecting at least two points from the data tree. Once the object is created, the user can rename the muscle object, modify the path, add wrap objects (see Par. 7.3.1.1.6) and manually assign the musculotendon dynamics parameters from the VME tab of the control panel.

7.3.1.1.8 Auto Create Muscles This operation automatically creates OpenSim muscle objects (see Par. 7.3.1.1.7) from the corresponding landmarks that define each muscle path. The names of the landmarks identifying the path have to be defined in the following format:

muscle name orig
muscle name via 1
muscle name via 2
muscle name via n
muscle name ins

When launched, the operation automatically creates all muscle objects corresponding to the landmarks found in the data tree. The user can rename the muscle object, modify the path, add wrap objects (see Par. 7.3.1.1.6) and manually assign the musculotendon dynamics parameters from the VME tab of the control panel.

7.3.1.2 Group The Group is a particular VME that defines a branch of the data tree. By default it is created as child of the selected VME in the tree and is named group, then the name can be changed in the VME tab.

7.3.1.3 Add Landmark This operation creates a landmark cloud as a child of the VME selected when the operation is launched. The user can interactively pick the landmark on the selected VME or manually enter the coordinates in the Position text boxes from the Landmark Props tab. By clicking on an existent landmark name, the label becomes blue and the position can be modified by either re-picking the landmark or manually changing the coordinates. It is possible to remove a landmark through the Remove button and manually enter the name of the landmark by typing in the Name text box. In addition, the user can load a dictionary with the landmark names through Load dictionary and save the current landmark names to a dictionary through Save dictionary.

7.3.1.4 Reference System This operation creates a reference system (Reference System VME) as defined in Par. 5.1.8.

7.3.1.5 Anatomical Landmark Cloud This operation creates a Landmark Cloud VME from a predefined set of anatomical landmarks according to recommendations of the International Society of Biomechanics [2]. Once launched, the user can choose a specific landmark cloud preset from a drop-down menu (i.e., show all, right hip, left hip, right knee, left knee, right ankle and left ankle). The system will automatically recognize the presence of the landmarks of the preset in the subgroup, by highligthing them in red. The user can add the landmarks that are not present (in grey) following the procedure described in Par.7.3.1.3.

7.3.1.6 Anatomical Reference System This operation creates joint reference systems from anatomical landmark clouds according to recommendations of the International Society of Biomechanics [2]. Once launched, the user can choose the lower-limb joint (i.e., right hip, left hip, right knee, left knee, right ankle and left ankle), and the system will automatically visualize the corresponding landmarks necessary to create the reference systems. If missing, the user can pick the landmarks from the data tree by hitting the corresponding button. The outcome are two anatomical reference systems per joint, i.e., one in the parent body and the other in the child body.

7.3.1.7 Parametric Surface This operation creates a parametric surface whose parameters can be defined as in Par. 5.1.7.

7.3.1.8 Freeze VME This operation freezes procedural VMEs by fixing once for all VME parameters, such as position and displayed dimension of a reference system, or type and parameters of a parametric surface. The type of VME created from the operation depends on the input VME, i.e. a parametric surface or a Reference System gives as output a surface VME.

7.3.1.9 Distance Meter This operation creates a meter whose parameters can be defined as in Par. 5.1.10.

7.3.1.10 Average landmark This operation creates a landmark VME that results as the average position of the selected landmark cloud.

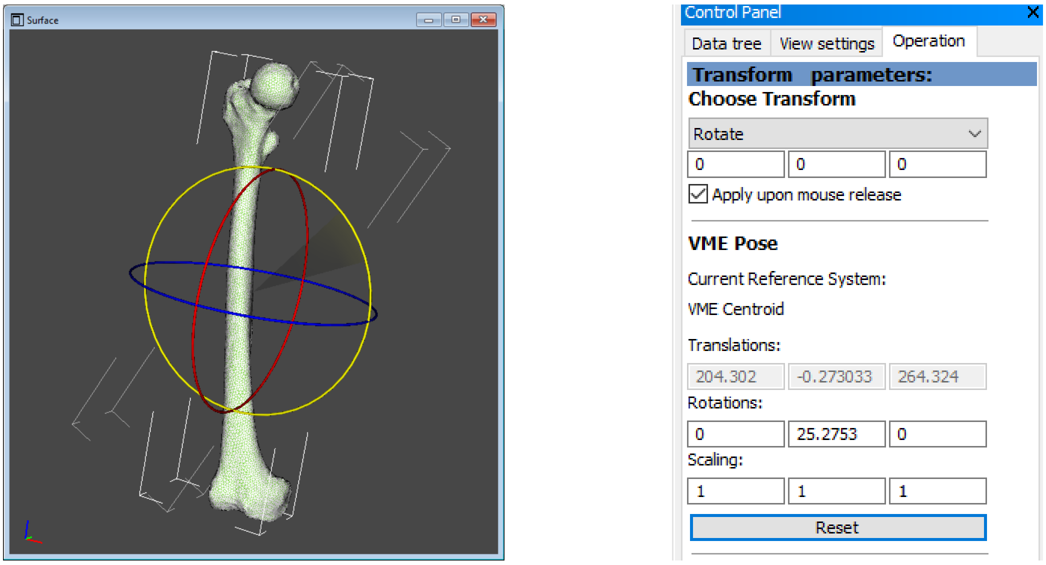
7.3.1.11 Extract Isosurface This operation creates a surface VME that results as the iso-density surface (isosurface) extracted from a selected volume VME, by selecting an iso-density threshold. The user can change the threshold grey level to define the iso-density surface (called contour value). It is possible to navigate the volume to check the surface contour and to pick a new threshold value on the volume slice. The volume slice can be removed from the visualisation (slice checkbox) or changed of position with the corresponding slice position slider. The optimize option (activated by default) automatically applies a filter to the surface before creating the VME in the data tree, which deletes the closed surfaces internal to the extracted surface.

7.3.2 Modify

7.3.2.1 Transform This Operation allows the user to perform affine transformation (roto-translation and non-uniform scaling) on the VMEs. The input is any VME in the data tree. The output is a modified VME, with a matrix pose applied to it (please note the original dataset, stored in .vtk format, is not changed, just a pose matrix is applied to it whenever visualizing or interacting with it).

The user interface (Operation panel in the data tree) lets the user apply the transformation either by inserting text or interacting with Gizmos (Figure 36).

There are two ways to perform the transformation: (i) Choose Transform, i.e. selecting Translate, Rotate or Scaling from the drop-down menu and assign (either by texty entry or gizmo interaction) the values; (ii) VME Pose, i.e. typing the values to achieve for translations, rotations and scaling.



Transform parameters:

Choose Transform

Rotate

0 0 0

☑ Apply upon mouse release

VME Pose

Current Reference System:

VME Centroid

Translations:

204.302 -0.273033 264.324

Rotations:

0 25.2753 0

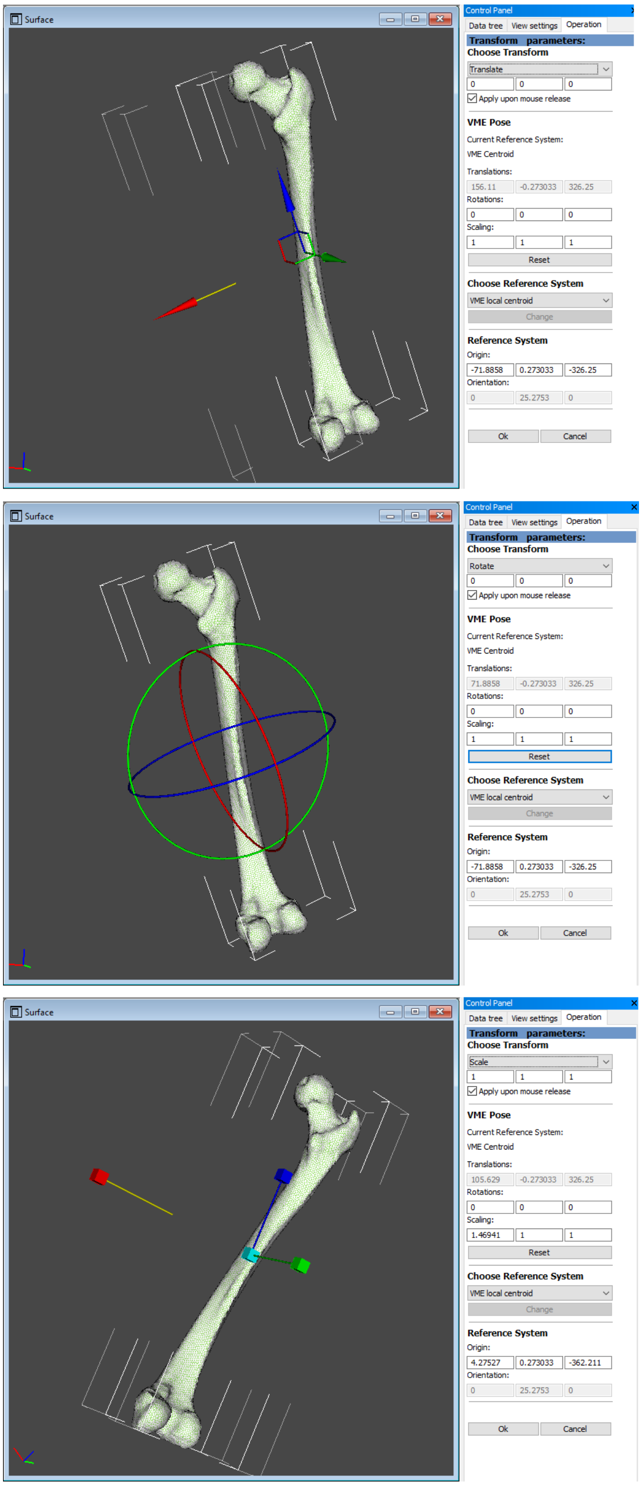
Scaling:

1 1 1

Reset

Figure 36

If interacting with Gizmos, the user has first to select the transformation mode (translate, rotate, scale) (Figure 37).



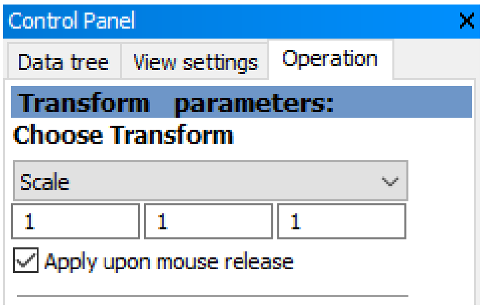
Transform parameters:
Choose Transform: Translate
0 0 0
☑ Apply upon mouse release
VME Pose
Current Reference System: VME Centroid
Translations: 156.11 -0.273033 326.25
Rotations: 0 0 0
Scaling: 1 1 1
Choose Reference System: VME local centroid
Reference System
Origin: -71.8858 0.273033 -326.25
Orientation: 0 25.2753 0

Transform parameters:
Choose Transform: Rotate
0 0 0
☑ Apply upon mouse release
VME Pose
Current Reference System: VME Centroid
Translations: 71.8858 -0.273033 326.25
Rotations: 0 0 0
Scaling: 1 1 1
Choose Reference System: VME local centroid
Reference System
Origin: -71.8858 0.273033 -326.25
Orientation: 0 25.2753 0

Transform parameters:
Choose Transform: Scale
1 1 1
☑ Apply upon mouse release
VME Pose
Current Reference System: VME Centroid
Translations: 105.629 -0.273033 326.25
Rotations: 0 0 0
Scaling: 1.46941 1 1
Choose Reference System: VME local centroid
Reference System
Origin: 4.27527 0.273033 -362.211
Orientation: 0 25.2753 0

Figure 37

The user can activate the "Apply upon mouse release" option (Figure 38) that allows him to visualize the transformation only after the mouse release and not interactively following mouse movement.



Transform parameters:

Choose Transform

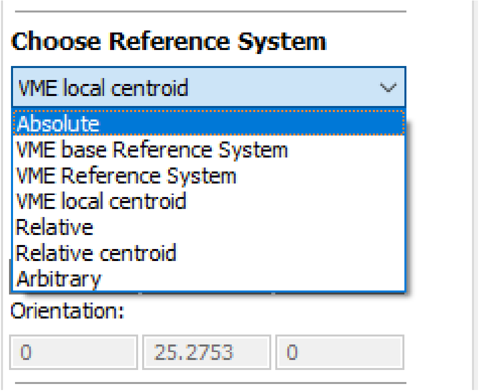
Scale

1 1 1

☑ Apply upon mouse release

Figure 38

The VME Pose parameters shown in the text are interactively updated when moving Gizmos. The pose parameters and the Gizmos can be shown in the following reference systems (Figure 39):



Choose Reference System

VME local centroid

Orientation:

0 25.2753 0

Figure 39

- Absolute, i.e. the global reference system
- VME base Reference System, i.e. the local reference system of the selected VME at the beginning of the operation, so that the VME base reference system is fixed and the user can control the transformation entity on the text entries at each instant.
- VME Reference System, i.e. the local reference system of the selected VME. The values in the VME Pose will always be 0 for Translations and Rotations, and 1 for Scaling.
- VME local centroid, i.e. a reference system centered at the centroid of the selected VME; the user can modify the coordinates of the reference system origin to control the transform entity
- Relative, i.e. a reference system specified by the user among the VMEs present in the data tree

- Relative centroid, i.e. a reference system centered at the centroid of a VME specified by the user among the ones present in the data tree
- Arbitrary, i.e. an arbitrary reference system that the user can define from the global reference system, by modifying the coordinates of the reference system origin.

7.3.2.2 Register Landmark Cloud This operation registers a source landmark cloud onto a target landmark cloud by using the Singular Value Decomposition (SVD) method[3]. This registration provides as output a new landmark cloud generated from a transformation that, applied to the source cloud, best fits the landmarks coordinates with respect to the target ones. The registration is performed by matching landmarks that have the same name in the two landmark clouds. This implies that the two landmark clouds must have a sub-set of landmarks with common names. It is possible to perform the operation through three types of registration: rigid, similarity, or affine (from the reg. type box):

- Rigid: translation and rotation only are allowed
- Similarity: translation, rotation and scaling are allowed
- Affine: any affine transformation is allowed

The follower surface button allows to optionally select a surface VME that will get applied the same transformation of the registered cloud.

The weighted registration button allows to assign weights to the landmarks to be used in calculating the registration (default is 1).

7.3.2.3 Register Surface This operation registers a source surface or landmark cloud onto a target surface or landmark cloud. The operation uses the Iterative Closest Point (ICP) technique for surface registration. The registration operation output is a new VME generated from an affine transformation that, if applied to the source VME, makes the points of the source fit the target ones. The user can set the iteration number of the algorithm (in conv. step inside the operation tab).

7.3.2.4 Filter Surface This operation takes as input a surface VME and applies to it a combination of vtk surface filters (smooth, decimation, etc.). The user can use different type of filters and set the parameters for each one. For example, with the smooth filter it is possible to define the iterations number.

7.3.2.5 Deform Surface This operation deforms a surface according to differences between two polylines and a chosen deformation method. It takes as input a surface VME and two polylines. The operation opens a pop-up window where the GUI allows the user to:

- Automatically create a base curve to deform
- Select two curves (the original curve and the deformed curve) from the data tree
- Edit both the original and the deformed cure by adding, moving, deleting and other editing options on curve points

• Show/Hide curves and surfaces in the preview window

• Select alternative deformation methods (Blanco's method, Modified Blanco's method and simple warping) • Saving both the original and the deformed curve

7.3.2.6 Surface Mirror This operation mirrors a static surface along any of the coordinate axes.

7.3.2.7 Crop Volume This operation extracts a rectangular volume of interest from a given volume. The user has two ways for selecting the region of interest (ROI):

• Using the interactive gizmos: through the mouse, the user selects the face handle to move and generates a parallelepiped to crop.

• Using the text boxes: the user writes the dimensions of the ROI. Both the handles and the ROI can be taken off the visualisation by unchecking the corresponding boxes.

7.3.2.8 Volume Resample This operation resamples the given volume VME on a regular grid. The user can remove from the display both the gizmos for the bounding box dimensions definition and for the bounding box orientation, using respectively the check box Show handle and show gizmo transform. The dimensions of the bounding box can also be set by inserting numerical values into the text boxes in the ROI selection part, while in the ROI orientation part the user can first select from the combo box if he/she wants to translate or rotate the bounding box. Translations and rotations can also be set by inserting numerical values into the Bounding Box Origin and Bounding Box Orientation text boxes. The user can set the output volume spacing as well as the auto-spacing of volume.

7.3.3 Measure

7.3.3.1 2D Measure This operation interactively performs the following measurements onto 2D representations of VMEs (i.e. on the CT Slices of the RXCT View).

• Points: the distance between two points

• Lines: the distance between two parallel lines

• Angle by lines: the angle between two independent lines

• Angle by points: the angle between two lines with a common point

• Indicator: creates an arrow with a label

Default units are mm for distances and degrees for angles. Any measurement can be stored in the .nms file with a description.

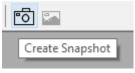
7.3.3.2 VOI Density This operation computes the volume and the average volume density in the region defined by the surface, from a volume VME and a surface VME.

7.3.4 Export OpenSim Model

This operation creates the OpenSim model file from the selected OpenSim Model VME (Par. 5.1.11). Once launched, the user can choose the file name and directory, and the

.osim model file will be created. The model file includes the current information that are included in the created OpenSim objects.

7.4 Snapshot Manager

The user can create and manage snapshots of the main working area any time. From the tool bar, hitting the create snapshot button (

) will create a snapshot with a white background, while manage snapshot will allow to manage all the snapshots created, using the options delete, save image as, save all, and print (Figure 40). In addition, the user can personalize the image resolution and the background color any time using the snapshot settings (Figure 41):

Path: Edit → Settings → Snapshot

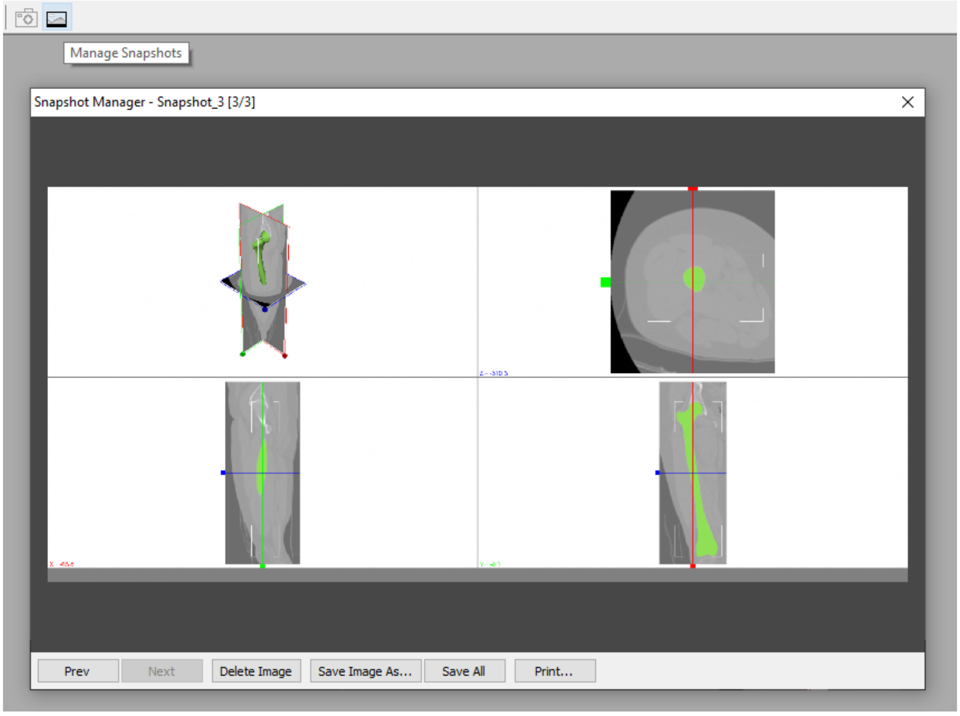
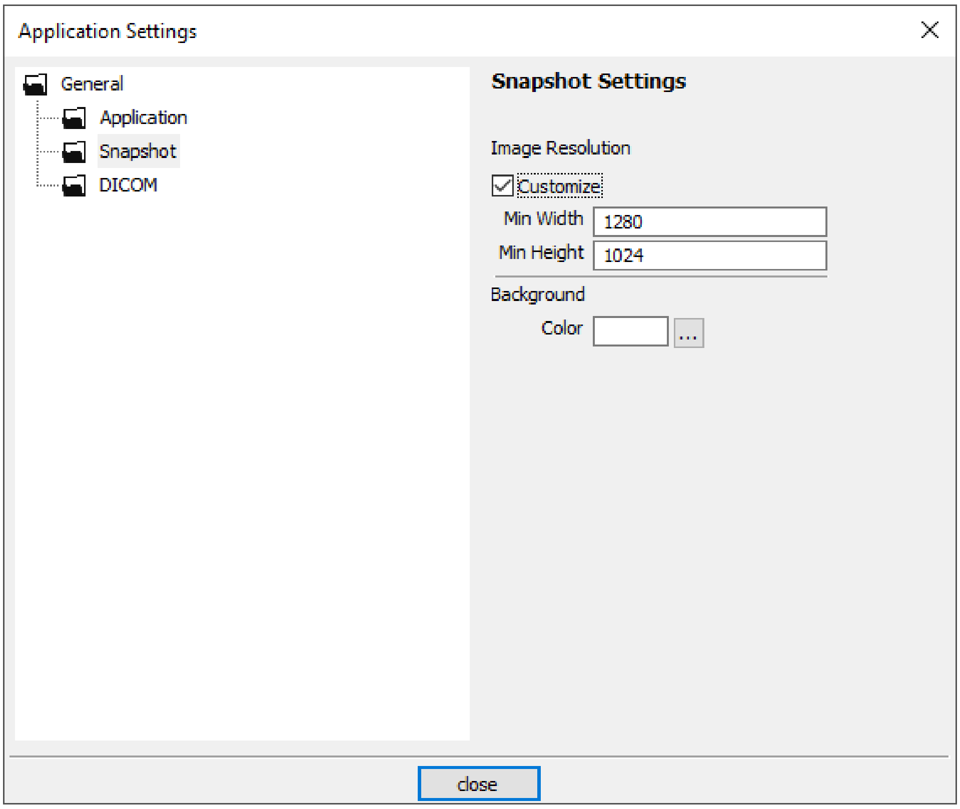


Figure 40



Snapshot Settings

Image Resolution

☑ Customize

Min Width 1280

Min Height 1024

Background

Color

Figure 41

References

[1] Giordano Valente, Gianluigi Crimi, Nicola Vanella, Enrico Schileo, and Fulvia Taddei. nmsBuilder: freeware to create subject-specific musculoskeletal models for OpenSim. *Comput Methods Programs Biomed*, 152: 85-92, 2018. 3

[2] Ge Wu, Sorin Siegler, Paul Allard, Chris Kirtley, Alberto Leardini, Dieter Rosenbaum, Mike Whittle, Darryl D D'Lima, Luca Cristofolini, Hartmut Witte, Oskar Schmid, and Ian Stokes. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion–part I: ankle, hip, and spine. International Society of Biomechanics. *J Biomech*, 35(4):543–548, 2002. 5.1.9, 7.3.1.5, 7.3.1.6

[3] Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. 7.3.2.2