Noname manuscript No.
(will be inserted by the editor)

A Visualization Tool for OpenSim

Nicolas Pronost · Anders Sandholm · Daniel Thalmann

Abstract This paper outlines the functionalities of a visualization tool for the neuromuscular platform OpenSim. The goal is to allow scientific visualizations of simulation data directly embedded within the OpenSim GUI. Basically, this tool creates visualization tasks defining and displaying relationships between musculoskeletal data and 3D objects. This paper presents how the tool works and what functionalities have been implemented.

Keywords Scientific visualization · Neuromuscular simulation · OpenSim software

1 Introduction

To study the results of neuromuscular simulations, OpenSim [1] users usually plot datasets (forces, moments, activations *etc.*). However, using such method has some drawbacks. Simulations contain information of different types and in large quantity. If the user wants to display all of them using traditional two dimensional plots, comparing results become very difficult. Here, we present a scientific visualization tool as a solution where numerical results are mapped to visual properties (color, opacity, 3D representation *etc.*). Since these properties can be applied on different geometries, simulation results can be displayed in one single view making comparisons between numerical values easier. The main objective of the tool is so to create relationships between the musculoskeletal data and visual properties. The configuration of these relationships are realized thanks to a setup file that is manually editable. Each visualization task has its own parameters described in the setup file. We describe here how we designed the tool (section 2) and what functionalities have been implemented (section 3).

2 Design

The tool is working around two major components: the tasks and the performers (Fig. 1). The tasks define which data need to be studied, like marker errors or muscle activations. Those values correspond to inputs retrieved from a storage file of a simulation tool or directly referenced in the setup file. A task knows how to use the information provided and how to execute the visualization process. A task can be static, called `Visual Task` where the data do not evolve over time, or time-dependent, called `Animated Task` where the data are updated regarding to the motion state. Each task uses a performer to affect the view by modifying current components (like meshes properties) or adding new elements (like streamlines). The performers create the mapping between the data from the tasks and its visual properties, and then update the 3D scene.

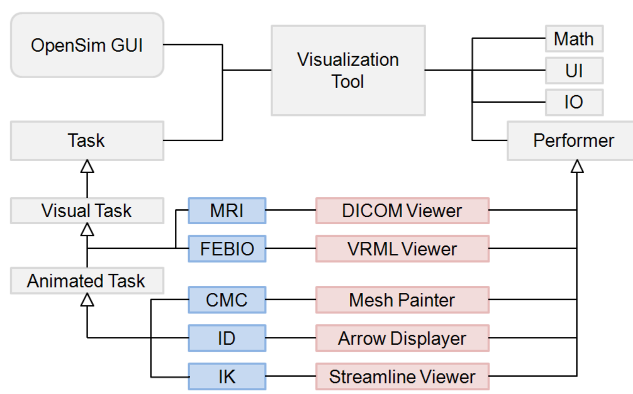

Fig. 1 Simplified overview of the software design of the tool

3 Functionalities

The tool currently provides 5 visualization tasks (Fig. 2).
`MRI`. In this task we propose to visualize and manipulate MRI volumes within the model view thanks to three orthogonal slicing planes. The idea of visualizing MR images in a musculoskeletal platform comes from needs in atlas-based and segmentation-based modeling of action lines. This task can be particularly useful to compare scaled models to actual data. An inverse kinematics process is executed to align up the posture of the model with the posture of the subject during acquisition.
`IK`. The errors between experimental markers and virtual markers during the tracking of a motion are often

N. Pronost, A. Sandholm and D. Thalmann
École Polytechnique Fédérale de Lausanne - Virtual Reality Lab
E-mail: nicolas.pronost@epfl.ch

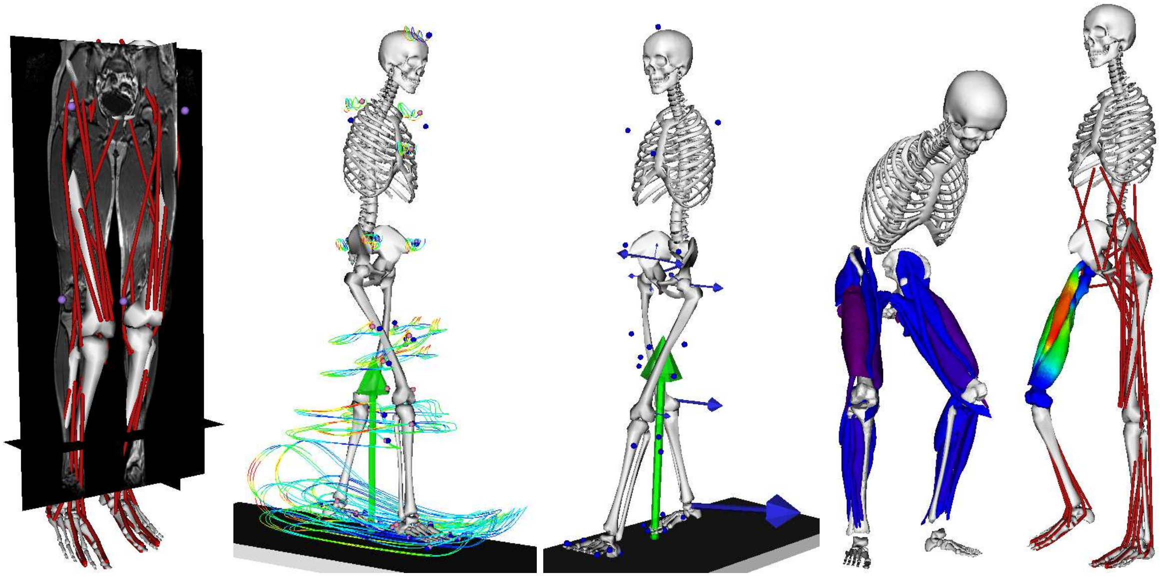

Fig. 2 The five implemented tasks visualizing respectively MR images, inverse kinematics errors, moments from inverse dynamics, muscle activation from computed muscle control, finite element simulations

critical in the evaluation of a simulation. This task enables to point out quickly the best and worth markers, improving the evaluation of the simulation in a more interactive way. Our tool computes the errors between the experimental and virtual markers and displays them using the streamline performer. A streamline indicates the path followed by a virtual marker while its color represents the distance to the experimental marker trajectory.

`ID`. Visualizing forces and moments over time using plots can be tedious when the user wants at the same time to know the corresponding motion phase (typically in a gait cycle), or when the number of data is large. By displaying the moments directly on the model during motion, the user can save time to analyze inverse dynamics simulations. This task reads moments around the rotational degrees of freedom (DoF) and displays them as arrows directly in the model view. The direction of an arrow lies along the rotational axis of the corresponding DoF and its size represents the magnitude of the moment.

`CMC`. This task displays muscular forces and activations from the Computer Muscle Control tool. This task associates muscles to activations or forces by defining assignments between muscle 3D geometries and action lines that exist in the model. When several action lines are specified, the user can define weights defining the contribution of each action line to the final activation. Either color or opacity can be used to visualize the activation or force of each muscle.

`FEBIO`. Nowadays, two main kinds of simulations are used to study the inner functionality of the human body: neuromuscular simulations and finite element simulations (inner displacements, stress on bones *etc.*). This visualization task aims at visualizing FE simulation results directly inside the model view of OpenSim. This task can read files from the post-process application of the FEBio platform which describe the geometries of the objects over time. The task displays the volumetric deformations of those objects and colors their cells according to the deformations.

4 Conclusion

We have presented a tool that creates scientific visualizations of musculoskeletal data. The visualizations are tightly integrated within the OpenSim user interface defining relationships between musculoskeletal data and visual representations. The tool was designed to allow an easy extension of its functionalities just by adding task and performer classes. We so hope future developers will use and extent the tool to create new functionalities.

Acknowledgements This work is supported by the European Marie Curie Program under the 3D ANATOMICAL HUMAN project (MRTN-CT-2006-035763).

References

1. S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, and D.G. Thelen. Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54:1940–1950, 2007.