



ISB 2013
BRAZIL

XXIV CONGRESS OF THE INTERNATIONAL
SOCIETY OF BIOMECHANICS

XV BRAZILIAN CONGRESS
OF BIOMECHANICS

MOKKA – BTK: OPEN-SOURCE FRAMEWORK TO PROCESS AND VISUALIZE BIOMECHANICAL DATA

¹Arnaud BARRE, ²Stéphane ARMAND

¹Laboratory of Movement Analysis and Measurement (LMAM), EPFL, Lausanne, Switzerland

²Geneva University Hospitals and Geneva University, Switzerland; email: stephane.armand@hcuge.ch

SUMMARY

Our objective was to develop and disseminate a free, open-source, cross-platform software able to read, process and visualize biomechanical data based on a standard (C3D) and proprietary file formats (most of motion analysis systems). Mokka (standalone application) and BTK (bindings for high-level languages: Matlab, Octave, Python) were developed to provide a cost-effective and efficient tool to deal with motion analysis data (3D points, analog data, force-plates data, model outputs, events and video). Three examples are presented with Mokka and BTK through Matlab: reading and exporting data, visualizing data and processing data.

INTRODUCTION

Motion capture systems produce large variety of data (3D trajectories, analog signal, etc). To standardize the data storage for the biomechanical community around the world, the company Motion Lab Systems (Baton Rouge, USA) proposed in 1980's to use a unique file format named C3D. This file format can embed raw coordinates of 3D points, raw analog data, system configuration (EMG channels, force plate, etc.), trial information (sample rates, date, filename, etc), subject information (name, age, physical parameters), and computed data (angles, moments, powers, etc).

Whereas C3D file format is widely used by companies and laboratories, only a small number of software can visualize and modify the integrality of the data in the C3D file. To the best of our knowledge, only one standalone application (Visual3D, CMotion, Germantown, USA) offers this possibility. However, this application goes beyond the requirements and the budgets of many people which only want to visualize and/or process raw signals. Moreover, many companies use their own proprietary file format (e.g., Motion Analysis Corp.) making data sharing impossible between laboratories without exporting them in another format.

The need to standardize and visualize data has already been recognized in biomechanics. Indeed, for clinical gait analysis, Tirosh et al. [1] proposed a web-based repository to facilitate sharing of gait data. To be easily adopted by the community, both the format and the tool used to read, write, modify and visualize data from any motion analysis

systems have to be efficient, robust, freely available, open-source, and multi-platform. The software OpenSim (<http://simtk.org/home/opensim>) is a good example of such a tool. Numerous advantages can be identified in this standardized and open-source approach: quality assurance, sharing data, collaborative researches, education, clinical consultation, improvement of data analysis.

Although the standard C3D file already exists, the software solutions associated with this file format and its availability for the biomechanical community needs to be improved. Therefore, our objective was to develop and disseminate a free, open-source, cross-platform software able to read, to process and to visualize biomechanical data based on standard and proprietary file formats.

METHODS

The general picture of the framework is presented in Figure 1. C++ was chosen as the programming language for the development, in combination with the Qt (<http://qt-project.org>) framework to create a cross-platform graphical user interface. The software compilation process is controlled by CMake (<http://www.cmake.org>). Documentation of the code can be obtained using Doxygen (<http://www.doxygen.org>). The software development process relied on hundreds of unit tests implemented with CxxTest (<http://cxxtest.com/>). All the source code was released under New BSD license. Source code, user instructions, and tutorials are available online (<http://code.google.com/p/b-tk/>). A dedicated website summarized the possibilities offered by Mokka (<http://b-tk.googlecode.com/svn/web/mokka/index.html>). Mokka (namely MOtion Kinematic & Kinetic Analyzer) is an open-source and cross-platform standalone software. It has been tested successfully on Windows (XP, 7), MacOS (10.5 - 10.8), and Linux (Ubuntu 10.04 - 12.04).

Separately, BTK (Biomechanical ToolKit), the core of Mokka, proposes bindings for high-level languages: Matlab, Octave and Python. These bindings provide the possibility to read, to modify, or to create an acquisition and to further export all its content in a C3D file. BTK can be used to easily implement custom scripts. Moreover, an emulation of C3Dserver (Motion Lab Systems) under Matlab was also developed and validated to run this tool in any operating system and architecture (32/64-bit).

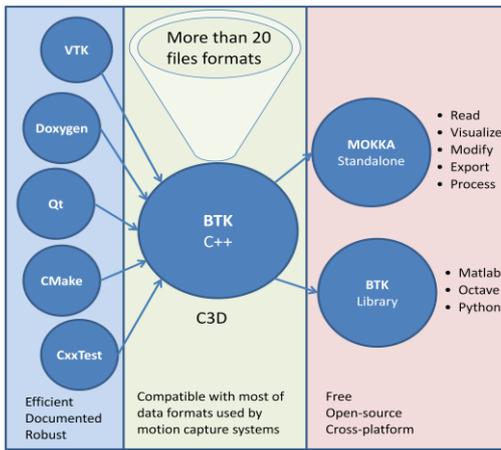


Figure 1: General representation of the framework

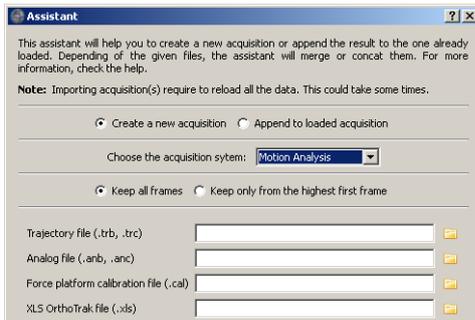
RESULTS AND DISCUSSION

The main possibilities of Mokka and BTK are presented within 3 examples.

1) Read and export data

The platform is able to read more than 20 files format coming from the most used motion analysis systems (AMTI, BTS Bioengineering, Codamotion, Delsys, Motion Analysis Corp., NaturalPoint Inc., Xsens, Vicon, etc.). All the data are stored internally in the same way. Then it is easy to open different files (e.g., trajectory file, analog file, etc) collected from an acquisition, merge them and save them into a unique C3D file. For example, for systems which record acquisition data in several files, Mokka proposes an assistant to merge them (Figure 2a) and BTK through Matlab merges all files with only 6 lines of code (Figure 2b).

a)



b) `% Matlab code`

```
hTRB = btkReadAcquisition('Gait.trb');
hANB = btkReadAcquisition('Gait.anb');
hCAL = btkReadAcquisition('Forceplate.cal');
hXLS = btkReadAcquisition('Gait.xls');
h = btkMergeAcquisitions(hTRB, hANB, hCAL, hXLS);
btkWriteAcquisition(h, 'Gait.c3d');
```

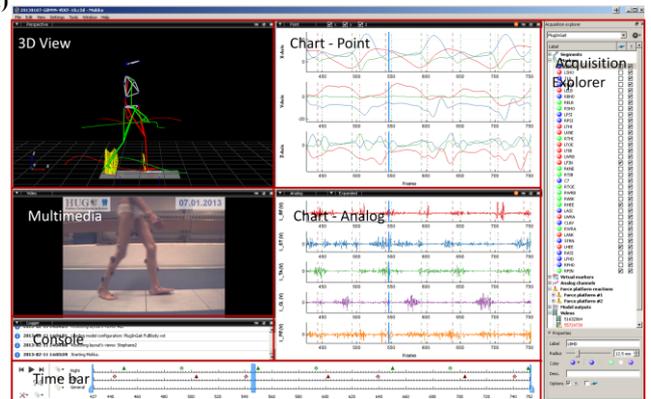
Figure 2: Import and merge Motion Analysis data in BTK; a) Mokka assistant; b) Matlab command

2) Visualize data

Acquisition data can be easily visualized by different views: 3D View, Chart (analog/point), Multimedia (Video) and Console (Figure 3a). The Time bar gives access directly to the visualization of a specific frame. It is also possible to manipulate acquisition events (heel strike, toe off, etc.). Numerous options are available such as multiple selections, expand/collapse charts, modify appearance of

the charts, save the configurations, modify the horizontal axe unit (frame, time, cycle), modify/add events, etc. Using BTK through Matlab, some commands are provided to extract acquisition's data and then, offer the possibility to implement programs to visualize them (Figure 3b)

a)



b)

```
% Matlab code
h = btkReadAcquisition('Gait.c3d');
angles = btkGetAngles(h);
figure; plot(markers.L_KNEE_ANGLE);
analog = btkGetAnalog(h);
figure; plot(analog.L_Rectus);
```

Figure 3: Extraction and visualization of motion analysis data. a) Mokka interface; b) Matlab code to extract data.

3) Process data

Currently, Mokka interface incorporates data processing for points, analog data, and events. It is possible for example to create an average marker or to filter, rectify, scale, remove offset, or compute envelop for analog (EMG) channels. Gait events can be also identified automatically. In the next future, several developments are planned: (1) integration of kinematics and kinetics computations with the most used biomechanical models, (2) Python console inside Mokka, (3) Plug-in integration for custom developments in Mokka.

CONCLUSION

By this work we presented an open-source and cross-platform framework to process and visualize biomechanical data in a cost-effective and efficient manner. The software runs on all major operating systems (Windows, Linux, MacOS X) and with high-level languages: Matlab, Octave, Python. Further instructions for installation and use are available online. As a collaborative and open-source project, all are welcome to contribute.

ACKNOWLEDGEMENTS

We acknowledge all the persons who contributed to this project: Alice Bonnefoy, Raphael Dumas, Fabien Leboeuf, Mathieu Lempereur, Florent Moissonet, Didier Pradon, Vincent Pomeroy, Patrick Salvia, Katia Turcot and the SOFAMEA. We acknowledge also all the companies which send us code or software license to integrate their file format or to compare results: BTS Bioengineering, Codamotion, Delsys Inc and Motion Lab Systems.

REFERENCES

1. Tirosh O, et al., *Comput Biol Med.* **40**:201-207, 2010.