Making Automatic Movement Features Extraction Suitable for Non-engineer Students

Nicola Corbellini
Gualtiero Volpe
nicola.corbellini93@gmail.com
gualtiero.volpe@unige.it
Casa Paganini – InfoMus, DIBRIS, University of Genova
Genova, Italy

ABSTRACT

Analysis of movement expression is a multidisciplinary research domain, that exploits the contributions from a wide variety of research fields, ranging from biomedical, computer science and robotic engineering, moving through psychology, to dance and performing arts. That is why there is the need to make tools also accessible to students and researchers with a non-technical background to foster their insights and facilitate their contribution. Since corpora are usually multi-modal, when they come as motion capture (MoCap) data, they could be quite difficult to analyze and annotate by people with a non-technical background. Therefore, the present work shows the prototype of a software tool that collects a library of algorithms, to process raw MoCap data. The tool allows the user to extract movement features through an easy workflow, interacting with a user-friendly graphical interface (GUI). The GUI usability has been preliminary user-tested with participants having different expertise in human movement features extraction. During the execution of three tasks, users' attitudes have been collected to assess GUI's ease of use and we found that it is perceived as a useful tool, but it requires basic previous knowledge to be fully understood.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**; **User interface toolkits**.

KEYWORDS

Features extraction tool, human movement analysis, non-verbal communication

ACM Reference Format:
Nicola Corbellini and Gualtiero Volpe. 2021. Making Automatic Movement Features Extraction Suitable for Non-engineer Students. In *Companion Publication of the 2021 International Conference on Multimodal Interaction (ICMI '21 Companion), October 18–22, 2021, Montréal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3461615.3485398

ICMI '21 Companion, October 18–22, 2021, Montréal, QC, Canada

ACM ISBN 978-1-4503-8471-1/21/10…$15.00
https://doi.org/10.1145/3461615.3485398

1 INTRODUCTION

This work presents the prototype of a software tool that collects a library of algorithms to process raw MoCap data. The main purpose of the tool is to provide students and researchers with an easy GUI to support them in the movement features extraction process, through a step-by-step workflow.

Human movement analysis is a wide area of investigation, most of the time addressed from a multidisciplinary point of view. As a result, data transformation and manipulation should be achievable by researchers with any kind of background. This would allow researchers with non-technical skills to be able to fully exploit corpora and to easily annotate and extract features from raw data. For instance, motion capture techniques range from less accurate, but portable, markerless techniques, leveraging computer vision algorithms, to highly accurate marker-based techniques using infrared cameras systems, like Qualisys[1] or Vicon[2] Motion Capture systems. Unfortunately, a technical background is often required to extract information from recorded data, due to the technical nature of these technologies and related tools to manage the obtained files.

Therefore, the software tool described in this work is mainly intended for researchers and students, interested in human body movement analysis, with a non-engineering background. Nevertheless, the tool is also suitable for expert researchers, since the expressive gesture library it embeds allows to speed up the features extraction process. Input data is meant to be marker-based MoCap data, however, the tool accepts raw 3D trajectories data coming from other sources (range imaging devices like Kinect, computer vision techniques and so on), as long as they are provided in `.tsv` (tab-separated values) format.

The tool consists of a library of algorithms for computing motion features organized at different levels, according to the conceptual framework described in [9]; it is based on *EyesWeb XMI* (eXtended Multimodal Interaction) platform[3] [6], an open hardware and software platform to design and develop real-time multimodal systems and interfaces. EyesWeb supports real-time synchronized recordings of multimodal channels and includes a number of software libraries. These consist of many modules specifically developed to extract information from movement data. For this reason, EyesWeb is often used in human movement science. Despite this, it is not easily accessible by everyone and, even for engineers and computer scientists, it could be time-consuming to learn. Indeed a GUI tool would overcome or fasten the filling of this gap. Moreover, the

[1]https://www.qualisys.com/
[2]https://www.vicon.com/
[3]http://www.infomus.org/eyesweb_ita.php

tool will be available under an open-source license as an additional toolbox to EyesWeb XMI [6] on the same website[3] to allow users to contribute to extend the library.

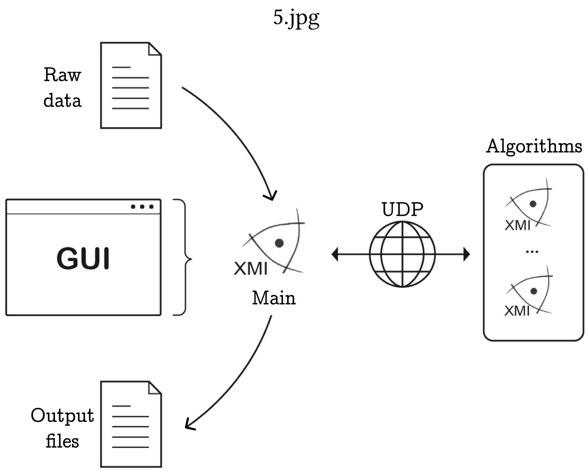


Figure 1: System implementation. The GUI is linked through the local network to the main EyesWeb patch and it allows the users to interact with the algorithms in a user-friendly way. The main patch takes raw data in `.tsv` format; it sends data and user's choices to the library patches, via UDP local connections and it receives back the extracted features, that are then saved as `.txt files`

2 RELATED WORK

Human movement analysis is a multidisciplinary research domain that heavily benefits from the insights coming from social sciences and humanities (i.e. psychology, performing arts and so on). Despite this, technologies and software used to process and analyze data usually require a strong technical background that only researchers with an engineering background own.

For instance, MoCap data can be processed with the MoCap Toolbox for MatLab [5]. This toolbox is an open-source set of functions, written in MatLab, that allows one to analyze and visualize MoCap data. It is mainly intended for music-related movement, but can be used for a wider scope. Another existing library is the Rigid-Body Dynamics Library (RBDL) [10]. It is an open-source library written in C++, that provides some algorithms to model and to compute rigid-body dynamics and physical quantities, like velocities, accelerations, linear momentum and so on. Both the aforementioned libraries require the users to have knowledge about MatLab or C++ environments or programming skills. The same EyesWeb XMI [6], upon which the tool has been developed, offers a library of modules for expressive gesture analysis [8], but it is not easily accessible and even for engineers and computer scientists it requires they learn to use the platform.

For what it concerns humanities, such as performing arts, the Jitter library of Max/MSP [1] and MotionComposer [2] are often used. The first is a library for the Max/MSP software, a visual programming language for music and multimedia. The library provides a set of objects for matrix calculations, hence for data represented as such, like images and videos; the latter is a software leveraging stereo-vision technology to process expressive gestures and movements and turns them into sounds. As stated in [3] "*these tools however are specially conceived for artists rather than for researchers needing scientifically reliable data*". That is why the software presented in this paper meets the needs for a precise and reliable tool, with an educational purpose for students and researchers without a background in engineering or computer science fields.

3 THE TOOL

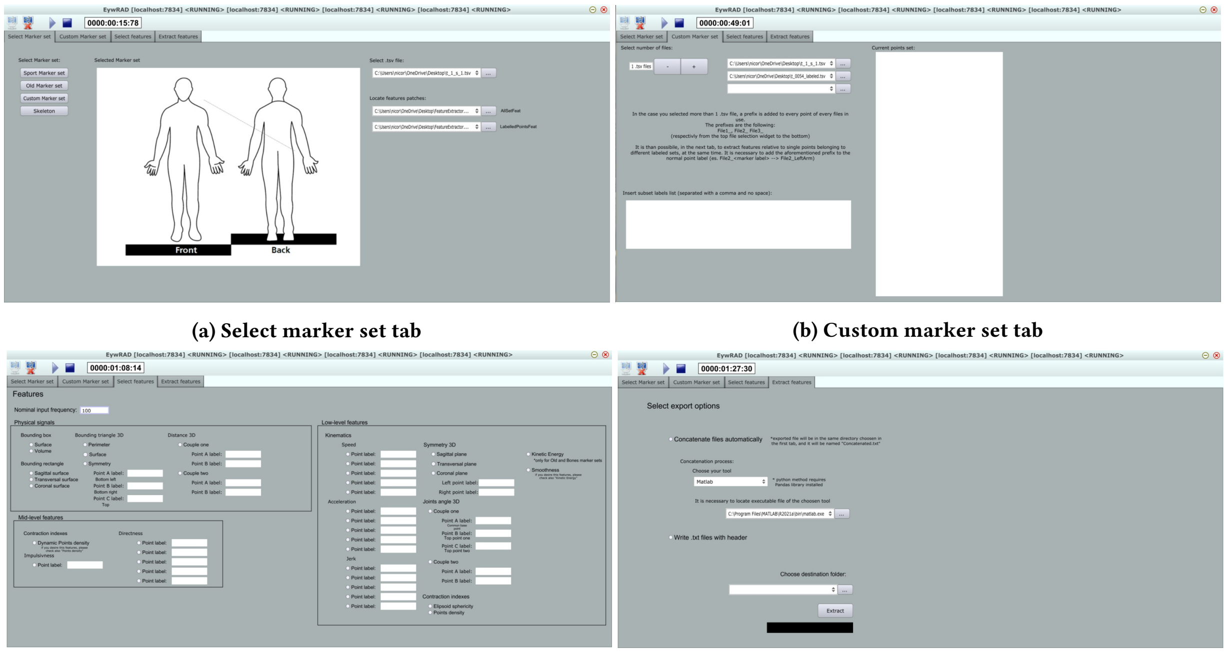
The software is intended to support researchers with a non-technical background during the process of properties and information extraction from raw data. In order to successfully accomplish this, some basic knowledge is required. Hence, users should be aware of how the data recording process happens. Specifically, they should know the concept of *marker set*, that is the configuration of reflective markers on the suit used during recordings with marker-based MoCap techniques. Moreover, they should know the specific marker set used to record the data they are processing and the frequency at which data has been recorded. The input data frequency is set by default to $100Hz$, since this is the typical frequency of marker-based systems like Qualisys[1], but it is a tunable parameter if necessary. Software requirements are, instead, to have EyesWeb [6] installed on the PC and also Python or MatLab, since they are required to merge output files, that are saved separately otherwise.

3.1 The architecture

The system consists of three modules: the first one is the *input module*, it stores and handles all input from the users; the second one is the *computation module* and it deals with features computation; finally, the third one is the *output module* and it returns the output files.
Specifically, the input module takes data files, their related information (i.e. frames number, marker labels and so on) and the user's interaction with the GUI. For instance, user's inputs are: marker set/subset configuration, features to compute and export options. When the user starts the tool, the input module calls the computation module and sends it raw data and the user's input via a UDP (User Datagram Protocol) local connection. The computation module automatically processes data and computes the features selected by the user. The computed values are sent to the output module through a UDP connection. The last module takes care of creating `.txt` files and formatting them according to the options user selected. Output files are a column vector of numbers, that are the computed values of the feature at each frame.

3.2 Implementation & Workflow

The software modules have been implemented by mean of EyesWeb XMI [6], that is a visual programming environment that allows the user to synchronize and process multimodal data, connecting blocks of code. This tool has been chosen since it provides an extensive library of modules for gestures expressiveness analysis [8]. Moreover, EyesWeb comes with a tool, EyesWeb Mobile, that allows developers to design user-friendly interfaces to interact with EyesWeb patches. These are applications that are made by



(a) Select marker set tab **(b) Custom marker set tab**

(c) Select features tab **(d) Extract features tab**

Figure 2: Screenshots of the GUI tabs

interconnecting EyesWeb blocks and that the users can execute. The presented tool consists of three patches - one for the tool execution and two for deploying the library for feature computation - and one meta-patch that runs the GUI.
The tool has been designed to guide the user in a step-by-step process, breaking down the workflow into tabs. There are three mandatory tabs and one optional (see Figure 2).

The first tab (Figure 2a) allows the user to:

- select the marker set used during the recordings. Preset configurations are *Old marker set, Sport marker set* and *Skeleton*
- select a subset of the marker set turning on and off single markers. For instance, only upper body, lower body, limbs, and so on;
- browse the input `.tsv` file;
- browse the library patches.

The second tab (Figure 2b) is an optional step and it allows the user to merge more data together. For instance, it let users to extract features of a group of people (max three) like they would on a single person. Moreover, it is possible to input a marker set with custom labels, that are different from the available preset.

The third tab (Figure 2c) allows the user to:

- set the input data sampling frequency;
- select the features to extract from raw data. Features are organized into three groups, i.e. *physical signals, low-level features* and *mid-level features* [9].

Most of the algorithms have been collected from previous researches on human expressive movements based on EyesWeb. For instance, some features are: the bounding triangle symmetry [16], contraction indexes [7] like ellipsoid sphericity [18], smoothness [11, 12], impulsivness [13] or directness [15]. Some features are computed on the whole set of points selected in the first tab and they only require to check the radio box; other features also require to pass the marker labels as parameters, since they need specific points as vertices of shapes (i.e. the vertices of the bounding triangle) or they are computed on a single point. The latter (that are kinematics and directness [15]) can be computed on up five different points at the same time.

The last tab (Figure 2d) allows the user to:

- merge all extracted files into one, choosing between two programming tools to achieve this (i.e. Matlab or Python) and to browse their executable files;
- export files with or without an EyesWeb header;
- select the destination folder for the exported files.

4 EVALUATION

Since the tool is at its earliest stage, an interactive prototype has been proposed to a group of volunteers, in order to assess the GUI's weaknesses and strengths. We chose to recruit participants with a variety of backgrounds, from researchers in human movement analysis to naive people, in order to investigate qualitative data. We aimed to understand to what extent users could trust the interface and found it easy to use.

4.1 Methods

12 volunteers were recruited, with a mean age of 30 years (SD=9). Among participants, only 4 were engineers or computer scientists. In order to assess users' attitude, a simple user-test was carried out remotely, on the Microsoft Teams platform. Since some participants

were foreign to human movement analysis and research, some slides were provided with images and a glossary of MoCap terminology. Then, the experimenter gave users remote control of his desktop to allow them to freely explore the GUI for 10 minutes, in order to make them familiarize themselves with options and functionalities. If they felt comfortable enough they could stop earlier and start with tasks execution. Users were asked to perform three tasks (Table 1) and data were collected by means of *think-aloud* techniques. At the end of the experiment, we asked for anonymous comments with an online form.

Tasks	
1	You've been provided with a file named "t_1_s_1.tsv", that have been recorded with the Skeleton marker set. You're interested in extracting the whole body Kinetic Energy and the perimeter of the Bounding Triangle of the head and the hands. You want files to be concatenated with Matlab, without the header and to be saved on the desktop.
2	You're interested in extracting features from multiple files at once. Files are named "t_1_s_1.tsv" and "t_1_s_2.tsv" and have been recorded with Skeleton marker set. Features you're interested in are: Distance from the head in file t_1_s_1 and the head in file t_1_s_2, points density of both and the speed of the left hand of the file t_1_s_1. You want exported files not to be concatenated, without a header and to be saved on the desktop.
3	You've been provided with a dancer trajectories file. It is named "t_0018_clean_format.tsv" and it's been recorded with the "Old" marker set. You're interested in extracting some features of the upper body (from head to hips, upper limbs included). You're interested in: bounding box volume, left hand and right-hand distance, directness of the head with a fixed number of points at 25 and left-hand acceleration. You want exported files to be concatenated with Python, with the header and to be saved on the desktop.

Table 1: Texts of the tasks provided to each participant during the user testing. Texts were written in the Microsoft Teams platform chat and available to the user during the task execution

4.2 Results & Discussion

In general, participants could perform tasks without major problems, except for Task 2 (see Table 1), which appeared to be confusing for most of them. In Task 2, participants were asked to manage two files and to use the *Custom marker set* option, some of them reported problems saying that the option implementation "*wasn't immediately comprehensible*". Many complained about the absence of feedback from the system when creating a custom set of points. Moreover, it has been suggested that "*in case of operations on a single file, it would be useful to make text fields unfillable/to make operative «Custom marker set» only if files are more than one*". This is because, as reported by many participants, it was not clear whether the *Custom marker set* options was a mandatory step to take or just an additional one. Despite critical points in Task 2, Task 1 and 3 returned good feedback.

In the online form users stated that during Task 2 they "*[...]started to understand what was required and the third task was already easier.*" or that they "*became more and more familiar with the interface*" along with the tasks execution. This gave us a slight insight about GUI learnability, making us expect to find a good, but improvable, slope of the learnability curve during future work.

Indeed, it emerged that the interface is not immediately intuitive, but a short phase of learning is necessary to fully understand its workflow (Table 2). That is why we aim to make it more intuitive in further design iterations, in order to make it more accessible to people without previous knowledge.

Moreover, useful insight emerged from participants think-aloud. For instance, many of them said they would have appreciated a "select/deselect all" button. It also emerged the need to improve the interlock design [14], which means to prevent the accessibility to

Qualitative data	
Topic	**Answer**
Ease of use/ Learnability	*"I have found the first task quite difficult compared to the next ones[...]. During the second one, I started to understand what was required and the third task was already easier. In the beginning, I felt a bit frustrated and confused, but with a careful analysis I cleared my mind regarding the task to do."*
	"I think that the system can work well for a user with a minimum of previous knowledge, making otherwise complex operations much easier. On the other hand, it may leave the occasional user confused and doubtful at first, but I think the learnability is very good: personally, I became more and more familiar with the interface as I carried out the tasks."
	Regarding the growing "tendency" in the use of software (GUI, UE, UI, ID, etc...) to be self-explanatory, some options "slow down" their use, as long as they are not properly understood. Once "learned"... I would say that it is a great tool to automate/speed up the whole process of feature analysis!

Table 2: Sample answers from participants' online forms

following steps until previous ones are not correctly executed. We plan to make all tabs locked until the users select all options from the current one and they confirm they want to proceed.
Furthermore, we found that it could be necessary to give users more information about their actions. For instance, we saw it would be convenient to provide a list of the available points, selected in the *Marker set tab*, during the features selection step, avoiding the user to go back and forth from the two tabs.

5 CONCLUSION AND FUTURE WORK

Future work will include implementation and correction of elements highlighted as missing or critical. We aim to improve the workflow by mean of an interlock design and to provide more feedback to the users about their actions (i.e. the list of points available or the list of features selected to double-check before starting extraction). We aim to implement a visualisation tool as well, to allow the user to have insights about the extracted features during the extraction process. In addition, further usability investigation will be carried out. We aim to involve more people during the user-test, to start a reliable quantitative analysis of the GUI efficacy and efficiency (i.e. error rate and time-base efficiency) or users' satisfaction. The user experience will be assessed through validated questionnaires, like the System Usability Scale (SUS) [4] and the User Experience Questionnaire (UEQ) [17]. They will give us a measure of participants perceived perspicuity and self-efficacy, dependability and many other scales. Nevertheless, a supportive qualitative analysis will be kept to be able to know and understand users' needs. We aim to conduct focus groups involving both experts and novice researchers, since the former can help to design the tool being more responsive to real research scenarios, while the latter will help to highlight key aspects of the user experience. Moreover, we aim to collect more algorithms for feature computation.

ACKNOWLEDGMENTS

We would like to thank all the Casa Paganini staff for the support during the development and for precious tips during the design process.

REFERENCES

[1] 1988. *Max*. Retrieved 27/06/2021 from https://cycling74.com/products/max/
[2] 2010. *MotionComposer*. Retrieved 27/06/2021 from http://motioncomposer.de/
[3] Paolo Alborno, Kelly Jakubowski, Antonio Camurri, and Gualtiero Volpe. 2018. A system to support non-IT researchers in the automated analysis of human movement. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. 1–3.
[4] John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
[5] Birgitta Burger and Petri Toiviainen. 2013. MoCap Toolbox – A Matlab toolbox for computational analysis of movement data. In *Proceedings of the 10th Sound and Music Computing Conference*, Roberto Bresin (Ed.). KTH Royal Institute of Technology, Stockholm, Sweden, 172–178.
[6] Antonio Camurri, Shuji Hashimoto, Matteo Ricchetti, Andrea Ricci, Kenji Suzuki, Riccardo Trocca, and Gualtiero Volpe. 2000. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal* 24, 1 (2000), 57–69.
[7] Antonio Camurri, Ingrid Lagerlöf, and Gualtiero Volpe. 2003. Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques. *International journal of human-computer studies* 59, 1-2 (2003), 213–225.
[8] Antonio Camurri, Barbara Mazzarino, and Gualtiero Volpe. 2003. Analysis of expressive gesture: The eyesweb expressive gesture processing library. In *International gesture workshop*. Springer, 460–467.
[9] Antonio Camurri, Gualtiero Volpe, Stefano Piana, Maurizio Mancini, Radoslaw Niewiadomski, Nicola Ferrari, and Corrado Canepa. 2016. The dancer in the eye: towards a multi-layered computational framework of qualities in movement. In *Proceedings of the 3rd International Symposium on Movement and Computing*. 1–7.
[10] Martin L. Felis. 2016. RBDL: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots* (2016), 1–17. https://doi.org/10.1007/s10514-016-9574-0
[11] Donald Glowinski, Nele Dael, Antonio Camurri, Gualtiero Volpe, Marcello Mortillaro, and Klaus Scherer. 2011. Toward a minimal representation of affective gestures. *IEEE Transactions on Affective Computing* 2, 2 (2011), 106–118.
[12] B. Mazzarino and M. Mancini. 2009. The need for impulsivity & smoothness: improving HCI by qualitatively measuring new high-level human motion features. In *Proceedings of the International Conference on Signal Processing and Multimedia Applications (IEEE sponsored), SIGMAP is part of ICETE -The International Joint Conference on e-Business and Telecommunications*. INSTICCPress.
[13] Radoslaw Niewiadomski, Maurizio Mancini, Gualtiero Volpe, and Antonio Camurri. 2015. Automated detection of impulsive movements in HCI. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*. 166–169.
[14] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.
[15] Stefano Piana, Alessandra Staglianò, Antonio Camurri, and Francesca Odone. 2013. A set of full-body movement features for emotion recognition to help children affected by autism spectrum condition. In *IDGEI International Workshop*.
[16] Stefano Piana, Alessandra Staglianò, Francesca Odone, and Antonio Camurri. 2016. Adaptive body gesture representation for automatic emotion recognition. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 6, 1 (2016), 1–31.
[17] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2017. Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence* 4, 4 (2017), 40–44.
[18] Michael J Todd and E Alper Yıldırım. 2007. On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics* 155, 13 (2007), 1731–1744.