Project Title:	**i-Treasures**: Intangible Treasures – Capturing the Intangible Cultural Heritage and Learning the Rare Know-How of Living Human Treasures
Contract No:	FP7-ICT-2011-9-600676
Instrument:	Large Scale Integrated Project (IP)
Thematic Priority:	ICT for access to cultural resources
Start of project:	1 February 2013
Duration:	48 months

Deliverable No: D4.5

ICH Indexing by Stylistic Factors and Locality Variations

Due date of deliverable:	31 January 2016
Actual submission date:	18 March 2016
Version:	1st version of D4.5
Main Authors:	Mickaël Tits (UMONS), Joëlle Tilmanne (UMONS), Sohaib Laraba (UMONS), Dimitrios Ververidis (CERTH), Spiros Nikolopoulos (CERTH)

Project funded by the European Community under the 7th Framework Programme for Research and Technological Development.

Project ref. number	ICT-600676
Project title	i-Treasures - Intangible Treasures – Capturing the Intangible Cultural Heritage and Learning the Rare Know-How of Living Human Treasures

Deliverable title	ICH Indexing by Stylistic Factors and Locality Variations
Deliverable number	D4.5
Deliverable version	Version 1
Previous version(s)	
Contractual date of delivery	31 January 2016
Actual date of delivery	18 March 2016
Deliverable filename	D4.5-ICH Indexing by Stylistic Factors and Locality Variations.docx
Nature of deliverable	Report
Dissemination level	PU
Number of pages	68
Workpackage	4
Partner responsible	UMONS
Author(s)	Mickaël Tits (UMONS), Joëlle Tilmanne (UMONS), Sohaib Laraba (UMONS), Dimitrios Ververidis (CERTH), Spiros Nikolopoulos (CERTH), Stathis Nikolaidis (CERTH), Anastasios Papazoglou Chalikias (CERTH)
Editor	Joëlle Tilmanne (UMONS)
EC Project Officer	Alina Senn

Abstract	This deliverable entitled “ICH Indexing by Stylistic Factors and Locality Variations” presents different approaches developed for the stylistic analysis of dance performances

	and pottery construction sessions. Motion Machine, a new framework for stylistic analysis and comparison of full body gestures is presented, along with a set of stylistic motion analysis features, and a module for statistical factors and features exploration. Three different dance use cases are approached using this framework. In the second part of the deliverable, a stylistic analysis of pottery gesture is conducted based on electromyography data from the MYO sensor.
Keywords	Motion analysis, mocap, stylistic factor, ICH, dance, pottery

Signatures

Written by	**Responsibility- Company**	**Date**
Mickaël Tits	UMONS	11/3/2016
Sohaib Laraba	UMONS	11/3/2016
Verified by		
Joëlle Tilmanne	Responsible for D4.5 (UMONS)	16/3/2016
Yiannis (Ioannis) Kompatsiaris	WP4 leader (CERTH)	16/3/2016
Approved by		
Nikos Grammalidis	Coordinator (CERTH)	18/3/2016
Kosmas Dimitropoulos	Quality Manager (CERTH)	18/3/2016

Table of contents

Executive summary

The main goal of Task 4.4 is the analysis of ICH performances by stylistic variations. The present report will describe the tools and approaches developed for the stylistic analysis of dance performances and pottery construction sessions. For the case of dance performances, as it was presented during the previous review, a choice has been made to focus this analysis on the analysis of full body gestures, and the approach is tested here more specifically on dance gestures. However, the tools and approaches presented in the present deliverable can be applied to any other discipline involving full body motion.

The first part of this report is articulated along four main sections:

- The framework for stylistic analysis and comparison.
- The motion features implemented within the framework for stylistic analysis independent of the underlying gesture.
- The gesture recognition and comparison approaches for stylistic analysis of similar underlying gesture (i.e. the same dance figure performed by two different dancers).
- The application of these two approaches to three specific use cases:
    - stylistic comparison of contemporary dance improvisations;
    - expertise level comparison in the execution of dance figures from the Walloon Region;
    - locality comparison: Tsamiko vs. dances from the Walloon region.

The second part of this report deals with the analysis of pottery gestures. More specifically, in order to identify style characteristics across pottery sessions, a reverse engineering approach was considered by trying to predict the forearm muscles contributing to each pottery gesture. We have designed a set of Electromyogram (EMG) based features such as muscles total pressure, flexors pressure, tensors pressure, and gesture stiffness, for the purpose of identifying differences in performing the same gesture across three pottery constructions namely bowl, cylindrical vase, and spherical vase. Our work was largely driven by the implementation of a framework that allowed us to visualize in real-time the signals generated from a Myo sensor along with the muscle activation level in 3D space. Using this framework, we were able to perform a number of emulation experiments for estimating the electrical activity of the forearm muscles and identifying the most suitable set of features for our study.

1 Introduction

When considering any performance, in addition to the recognition of high level patterns such as “which gesture has been performed”, or “which note has been played”, the human brain is also very sensitive to what can be referred to as the “style” of any particular low or high-level pattern, i.e. the little variations that can occur between several realizations of the same pattern or on the opposite, to the similarities that occur between different patterns (e.g. playing different notes “softly”, performing different dance figure in the same “energetic” style, etc.).

The aim of the stylistic analysis performed in Task 4.4 is to enable the isolation of the recordings components linked to the individuality of the performer (size, voice timber, gender, etc.), the component or the “style” of the action that identifies the rare knowledge which is studied (emotional state, motion quality, etc.), and the functional part of the action (rising a hand, singing the right melody, etc.). For instance, many dances will use the same low-level gestures, like raising both arms, but according to the dance, the gesture will be brisk or smooth, slow or fast, joyous or sad, etc. The same gesture or the same melody can convey very different additional information to the spectator, according to the style the performer adds to the basic action.

Extracting that stylistic factor will enable us to categorize content and compare the style of performances from different geographical regions or different performers. This analysis will also be used in order to assess the learning of rare knowledge by new users, as separation of fundamental action; inter subject variations and motion style will enable us to differentiate if the learner is singing the right melody but with a different singing style, doing the right dance moves but not in the right way, performing the right gestural emotion with a morphology different than the reference performer that was captured, etc. This comparison will enable the extraction of the stylistic factors, expressing the quality of the motion or sound analyzed. In addition to the individual characteristics influencing the style (gender, age, size, etc.), the influence of the artistic interpretation on the functional action is also encompassed in the stylistic variations. The analysis of the isolated stylistic parameters will hence enable an analysis of the artistic interpretation of the performance.

As it has been presented during the review, in the framework of this project a decision has been made to focus this stylistic analysis to the analysis of full body motions, and more particularly to focus on the dance use case. As it has been presented, two approaches have finally been retained for this purpose:

- Extracting meaningful features representing complete motion sequences (E.g. “Mean bounding box volume”), to be used when underlying motions are different.

- Using statistical HMM models (or other gesture recognition approaches) to compare style of same underlying motion (need sufficient amount of annotated data, with coherent dictionary)

In both cases, the focus is made on performances recorded thanks to motion capture techniques, either very precise (e.g. Qualisys system for optical motion capture), either with low cost sensors (e.g. Microsoft Kinect).

Motion capture (MoCap) is an emerging research and development field, which allows new prospects in many areas, including the intangible cultural heritage (ICH). MoCap raw data, generally consisting of 3D trajectories of a large set of nodes forming a skeleton, are highly dimensional and thus make the interpretation difficult. In order to answer the need of new tools to explore and analyze these complex data, we developed a new framework: MotionMachine. MotionMachine is a C++ library specifically developed for MoCap data manipulation, interactive visualization and analysis. It allows the fast prototyping of features derived from raw data, providing more high-level representations of the movements, and enabling easier interpretation.

In any physical performance, stylistic factors depend on various attributes of the movement. In the exploration of intangible cultural heritage, it is thus essential to emphasize relevant components of the movement that reflect the style of a performer. During the project, a large set of features have been developed in the proposed prototyping environment. These features focus on different aspects of the movement, including kinematics, kinetics, ergonomics, and different concepts of Laban movement analysis. In addition to these high-level features, gesture recognition algorithms have been imported as full part features.

The style of a performance may depend on different inter- and intra- individual variables, such as gender, size, or the type of the performance (for instance, different music attributes in the case of dance performance). To address this issue, a module has been developed in the MotionMachine environment, which allows statistical analysis and comparison of these variables, and their effects on the developed movement features.

As regards *pottery* style characteristics, our goal is to identify differences between the pottery gestures that may phenomenally perceived as identical and are difficult to isolate and transmit when safeguarding intangible cultural heritage. Towards this direction, for the purpose of capturing arts involving hand movements, a gesture style identification technique is needed that will be able to cope with the problem of fingers occlusion and non- physical availability of sensors (i.e. it is impossible to place accelerometers at finger tips). Hand gesture recognition is already researched in the past 20 years [33], and it is not in our scope to replicate such experiments but to extend gesture recognition with the identification of gesture styles within a class of gestures. It is interesting to find high level characteristics that make the master's gestures better than the learner's gestures. In the following some interesting gesture tracking technologies are reviewed and it is discussed how they can be extended to capture high level characteristics of handcrafts.

2 Stylistic variations analysis based on motion capture data, application to the dance use case

2.1 Motion feature prototyping framework (*MotionMachine*)

As presented in the introduction, motion capture is an emerging research field and the high dimensionality of mocap data calls for new tools to better take advantage of the data which is collected in many domains of application. In order to answer the need of such tools to explore and analyze the complex motion capture data, we developed a new framework called MotionMachine. MotionMachine is an open source C++ library that enables the rapid prototyping of motion features, their extraction on standardized motion capture data structures coming from typical motion capture file formats and live OSC streams and their selection so as to represent motion in the considered use case [6]. In this section, we first give an overview of the main structure of MotionMachine, and then focus on features and time management, enabling for both offline batch and real-time processing.

2.1.1 General structure

The overall data flow used in MotionMachine is presented in Figure 1. The library is built from two independent modules: one for data representation and feature extraction (built on the top of the *Armadillo* C++ library[1]), the other to take care of 2D and 3D scenes visualization and general user interaction aspects (built on the top of the *openFrameworks* C++ library[2]).

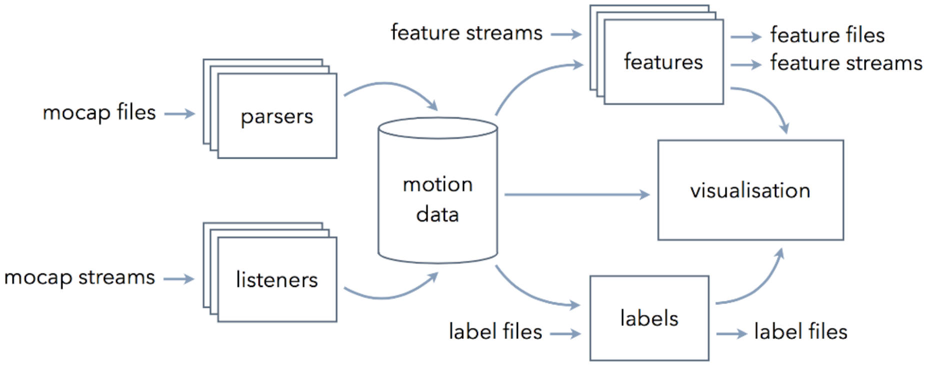


Figure 1 - Overall data flow used in MotionMachine: modular structure to process mocap data files and/or streams into features files and streams, labels files and visualization.

Four important core features are available in the MotionMachine framework:

1. *Skeletal Model Independent Motion Data*: Nowadays the number of available motion capture sensors is significantly growing. Most of these devices provide skeletal data, i.e. changes in the position and/or orientation of 3D joints and/or segments. For the moment, most of the achieved motion processing is model-specific and the corresponding know-how is not directly transferable to another model, though the underlying morphology is similar. In MotionMachine, we have developed a model-independent formalism for storing and accessing such skeletal data through APIs. The back-end is based on timed containers, which are further

[1] *Armadillo*: http://arma.sourceforge.net

[2] *openFrameworks*: http://www.openFrameworks.cc

explained in the next Section. The front-end gives an easy-to-understand access to a few nested data structures [6].

2. *Collections of Motion Feature Extractors*: MotionMachine is built around the idea that developers can write custom code to be inserted in the motion capture data processing pipeline, while still preserving the intuitiveness and efficiency of the overall environment. The design principle underlying the available collection of feature extractors is essentially container-driven and based on the idea that offline batch and windowed real-time processing should both be available by default for any built-in or third-party feature. We give an in-depth description of this idea in Subsection Features and time management 2.1.2.

3. *Interactive 2D/3D Scene View*: In MotionMachine, we wanted to improve the affordance of motion capture data processing by solving several visualization issues and bring the user faster to his/her valuable work. Such improvements were achieved by balancing the apparent complexity of the environment. As a result, the library comes with an integrated 2D/3D scene viewer for displaying mocap data on screen and interacting with the contents [6]. On the one hand, many visualization aspects are automatically determined from data types, included the so-called "degraded" modes where not all the information is available. On the other hand, 3D and 2D time lines are automatically synchronized (embedded app time handlers), helping to observe available motion capture data from different viewpoints.

4. *Annotation Layer*: In MotionMachine, we have integrated a lightweight annotation scheme. It allows the programmatic and UI-based insertion of *Labels* alongside the motion capture data. It means that the time tag of these *Labels* can be automatically derived from signal properties in the feature extraction code or added manually by the user. Moreover, these *Labels* get properly rendered in the 2D view and can then be rearranged manually. *Labels* can also be imported from and exported to label text files ("*.lab"* extension).

2.1.2 Features and time management

Time management can be a serious issue in motion capture based interactive applications: the application has its own life cycle and time line, input files and streams may have their own frame rate, time stamps or nothing, and any piece of data might reveal opportunities to be processed as one batch (offline), frame by frame or within a sliding window. Moreover, as MotionMachine is intended to be open-ended, i.e. with third-party feature extractors, we aim at optimizing the modularity of the code base, e.g. not duplicate code for offline vs. real-time scenarios.

In this project, we have decided to foster the role of underlying data containers in the process of framing the way time and skeletal structures are handled in MotionMachine, as well as how the feature extractors should be developed. Our *timed containers* aim at providing all the logic that enables the processing of motion capture data files and streams. The two main behaviors are:

1. *Hierarchical structure of data is transparent*: Motion capture data and more generally 3D animation data are described by using a hierarchical structure. On one

hand, the position and the orientation of a 6 degrees of freedom (DoF) limb can be encoded with absolute values in a global “world” coordinate system. On the other hand, these data can be computed in local coordinate system linked to a parent object, e.g. foot orientation is given relatively to the data of the leg. Following the aimed application, local or global values are the most appropriate description. The timed containers implement this kinematic chain and allow the user to choose between global or local motion descriptors.

2. *Time management within data is transparent*: In order to enable the processing of data extracted from different sources, MotionMachine provides a time modeling that respects two main constraints. Firstly, the timed container accepts data with a fixed frame rate or frames with time stamps, if they have not been recorded with a unique temporally stable device. Secondly the containers are able to store data from a real-time streams and data from offline prerecorded files. To do that, the container can switch between a ring buffer mode and a regular buffer mode, i.e. with a directly indexed-timed relation as showed in Figure 2. Series of APIs are provided to the developer so as to access the data at a given time, no matter the underlying time metric.

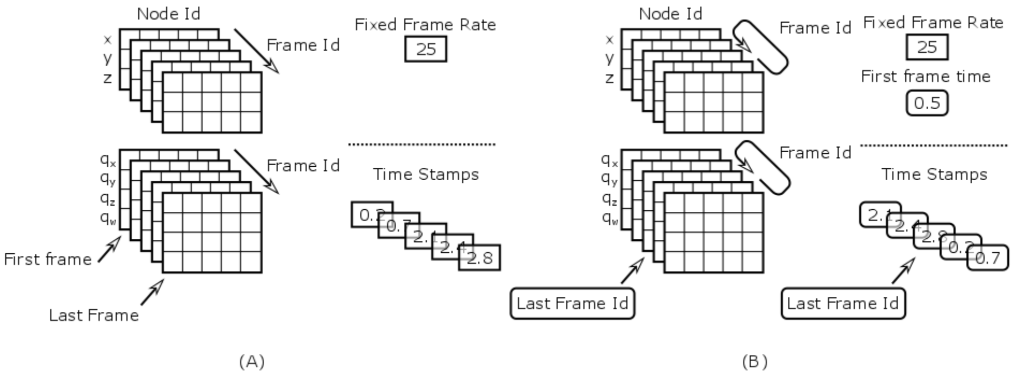


Figure 2 - Representation of the four modes to store the motion capture data in memory buffer. Position of each node is encoded by 3 Cartesian coordinates; rotation of each bone is encoded with 4 components of a unit quaternion. (A) Example of a track working in an offline mode. (B) Example of a track in real-time mode. In each case, the time can be modeled with a fixed frame rate or a time stamp vector. All time-related parameters are updated to insure a transparent management for the developer.

2.2 Stylistic features

One very transversal aspect of this project was to grow an amount of motion feature extractors that can be applied on various kinds of input data (files or streams). In this Section, we give an overview of what we aimed at keeping common among all the implemented features and then we give a list of feature categories and implementations that have been achieved.

We acknowledged this idea that feature extractor developers should not be “forced” to follow a DSP chain format. Indeed, the variability within motion capture based use cases is quite significant and we do not want to load MotionMachine with a complex and cumbersome patching environment, nor prevent developers to design their software following their scenario-driven constrains.

The one constraint that we put onto the development of motion feature extractors within MotionMachine is to take timed containers as inputs and outputs arguments and work with their APIs for accessing structural and time-handling information. This design choice also has another implication: it makes a lot of sense to work with Armadillo for the vector and matrix operations vs. transferring data into another format then back. This choice is motivated by the ability of Armadillo to do parallel computation on arrays using CPU optimization.

Nine different feature categories have been implemented and are presented here under. Six are high level motion feature categories, for each of which several feature extraction functions have been implemented inside the MotionMachine framework. Three are gesture recognition algorithms, either fully implemented within MotionMachine (relation features and DTW), based on external libraries (HMM-based gestures recognition based on the HTK library), or communicating with external modules (e.g. Wekinator).

2.2.1 Balance

The balance/weight category represents the idea of stability, balance and equilibrium, which can be perceived when looking at a skeleton with a given posture. It encapsulates concepts such as center of mass (CoM), floor contact points and base of support, in both continuous and binary ways: continuous features give an evolving metrics about the studied property, binary features represent a state. To compute these high level balance features, some intermediate features and geometrical operations are required and presented further in this subsection.

2.2.1.1 Center of Mass (CoM)

Description The center of mass (CoM) of a body is the point at which the whole mass may be considered as concentrated [15]. The center of mass of the whole body can be computed as the weighted average of the center of mass of each body segment in 3D space. The ``center of mass" feature is tridimensional since it is computed as the 3D-coordinates (position) of the center of mass of either one of the body segments or of the whole body.

Computation The position of the center of mass of a body depends on the distribution of the mass in the considered body. For the human body, several tables can be found in the literature and give an estimation of the position of the center of mass for each of the human body segments under the form of a ratio between the extremities of the considered segment. We have based our implementation of the center of mass computation on de Leva's work [16].

For each body segment, the COM is computed as the ratio between the position of both extremities of the considered segment, using the ratios found in [16]. The global center of mass is computed as a weighted sum of the position of the center of mass of each one of the body segments, the weight being defined by the relative weight of each segment of the body as presented in de Leva's paper.

$$CoM_{global} = \sum_{i=1}^{N} w_i * CoM_i \qquad \textbf{(1)}$$

with N = number of body segment considered, w_i the weighted mass of body segment i and CoM_i the center of mass of segment i.

Visualization A visualization of the center of mass of each segment as well as of the global center of mass has been implemented. The center of mass of each body segment is represented by a green sphere whose center is located at the xyz coordinates of the center of mass of the considered segment and whose diameter is proportional to the weighted mass of the segment. The global center of mass of the body is represented by a red sphere, as illustrated in Figure 3.

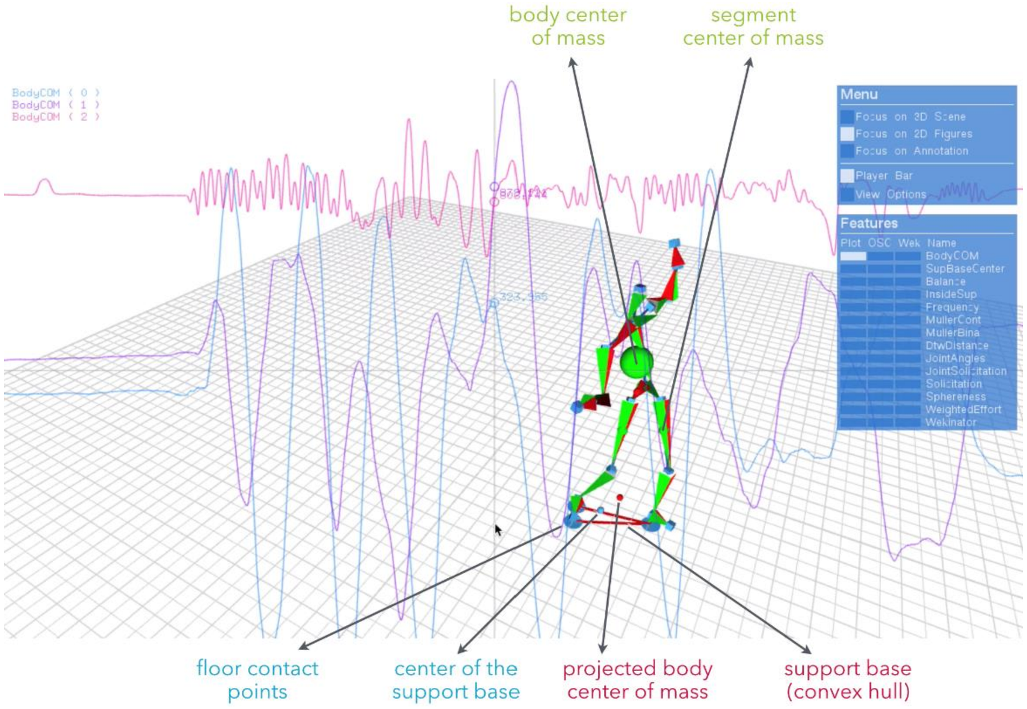


Figure 3 - Illustration of the Balance feature elements: body center of mass (with *xyz* position in 2D), segment center of mass, floor contact points, support base (convex hull), center of support base and projected body center of mass.

2.2.1.2 Floor Contact Points

Description When considering motion sequences performed on a standard planar ground, the equilibrium or balance of the body is highly dependent on the contact points of the body with the ground. This feature calculates which joints of the body can be considered as in contact with the ground.

Computation For each joint, this feature calculates if it is touching the ground or not. The binary decision is based on the height of each joint, using a threshold which can be adjusted for each joint, since the physical position of the mocap markers on the actor's body will change the height of the marker when the considered segment is actually touching the ground. By default, the threshold is set to 5cm, except for foot and ankle, for which it is computed as the mean height of the first ten frames of the motion sequence + 1.5cm).

Visualization Each floor contact point is displayed as a blue circle on the ground, using for its center position the x and y position of the corresponding joint.

2.2.1.3 Convex Hull of Support Base

Description The convex hull of a set of points is the smallest convex envelope that contains all of the points in the set. This geometrical feature is used in our case for instance to compute the contact base area based on the set of contact points with the ground. It can also be used to compute the area corresponding to the space occupation of the body on the ground at any given time, or to compute the area covered by a subject during a given motion sequence by computing the convex hull of the positions of the center of the body over time.

Computation The computation of the convex hull is and iterative process. From a set of points in a plane (defined by xy positions), it computes the set of points defining the convex hull containing all the points. Our implementation of the convex hull computation is based on [17]. From the defined convex hull, typical metrics can be determined such as contact base perimeter or area.

Visualization The implemented convex hull visualization consists in drawing the lines between the points defining the convex envelope, so as to show the area.

2.2.1.4 Binary Balance Value

Description In MotionMachine, the discrete value of the balance feature is named *isInsideCOMSupport*. This feature calculates if the projection of the body center of mass on the ground is inside the support base or not. The feature can take one of three values at each frame: 1 if the CoM_{global} is inside the support base, 0 if it is outside, and -2 if there is no support base (for instance when the subject is jumping). This feature can be seen as a ternary balance feature since it represents the equilibrium of the posture at each frame.

Computation This discrete balance feature is computed by testing if the projection of the CoM_{global} on the ground is inside the convex hull defined by the support base. This test of being inside is achieved by testing if the projected point is on the right of each of the successive segments of the hull.

Visualization The projection of the global center of mass on the ground is represented as a green sphere if the CoM_{global} is inside the support base, a red sphere if it is outside, and an orange one if there is no support base.

2.2.1.5 Continuous Balance Value

Description This continuous representation of the balance of the body is computed as the distance between the center of the support base and the projection of the body center of mass on the ground. When used in combination with the discrete value of the balance (*isInsideCoMSupport*), it illustrates how far from the equilibrium the current body posture is. The higher the feature is, the poorer the balance is. Another implementation that should be considered in the future would be to compute the distance from the border of the support base area (inside or outside) rather than the distance from the center of that area.

Computation The center of the support base is computed as the mean of the support joints positions. The continuous balance feature is then computed as the distance between the center of support base and the projection of the CoM_{global} on the ground.

Small distance means that the CoM_{global} and the center of the support based are more aligned vertically, large distance the opposite.

Visualization The visualization of the continuous balance value has been done in the 2D view. Indeed, peaks and valleys in the graph would inform quite clearly about moments where the body was particularly (un)balanced.

2.2.2 Rhythm and Periodicity

This Section describes our work in implementing a first set of periodicity-related features. It goes from low-level analysis to fundamental frequency estimation.

2.2.2.1 Auto-correlation

Description For a given signal, its auto-correlation is the cross-correlation of the signal with itself. The cross-correlation between two signals of length N measures their correlation for all possible delays 0, …, N-1. In the case of a pseudo-periodic signal, the auto-correlation would present a peak for each delay that is a multiple of the fundamental period, thus providing a way to detect whether a signal is periodic and to measure its period, if any. This feature takes as input a 1D signal whose values correspond to a temporal sequence of either node positions, angles or any feature described in Section 2.2.

Computation The input signal of length L is split into blocks, or *analysis windows*, of length N with a shift of H samples between the first samples of two successive blocks. For each analysis window $x_h(n)$ – with $h = 0, 1, \ldots, \frac{L-N}{H}$ – its auto-correlation $R_h(m)$ is computed as:

$$R_h(m) = \mathcal{F}^{-1}\left(\left|\mathcal{F}\left(\mathrm{x_h(n)w(n)}\right)\right|^2\right) \qquad \textbf{(2)}$$

where $w(n)$ is a *weighting window* and F and F^{-1} represent the *Discrete Fourier Transform* (DFT) and its *Inverse* transform (IDFT) respectively. The DFT and IDFT are computed over $2N$ points but only the first N +1 points are stored in the final result as $R_H(m)$ is symmetrical. The reason why DFT has been used instead of the straightforward correlation is because of the reduced computational complexity of the DFT. Optionally, the average value of $\mathrm{x_h(n)}$ can be removed before computing the DFT. The equation becomes:

$$R_h(m) = \mathcal{F}^{-1}\left(\left|\mathcal{F}\left(\left(\mathrm{x_h(n)} - \bar{x}_\mathrm{h}\mathrm{(n)}\right)\mathrm{w(n)}\right)\right|^2\right) \qquad \textbf{(3)}$$

This latter approach gives better results, especially when applied to node positions with a non-zero long-term average (typically, the vertical position of the head). Finally, the auto-correlation vectors Rh(m) are stored in an array R with $\frac{L-N}{H}$ columns and N + 1 lines.

Visualization The auto-correlation array R can be displayed as a 2D curve changing over time. Shortly, the h^{th} column $R_h(m)$ can be shown on screen for all time stamps corresponding to frame indices f such that: $h \times H \leq f < (h + 1) \times H$.

2.2.2.2 Power Spectral Density

Description The Power Spectral Density (PSD) is a measure of how the power of a given signal is distributed across the frequency domain. Random signals tend to have a continuous PSD and periodic signals exhibit some peaks.

Computation The computation is similar to that of the auto-correlation. An input signal is split into analysis windows $\mathrm{x_h}(\mathrm{n})$ and a PSD $S_h(m)$ is computed for each $\mathrm{x_h}(\mathrm{n})$ and stored in an array *S*. The computation is made by means of the squared absolute value of the DFT of $\mathrm{x_h}(\mathrm{n})$ over *N* points:

$$S_h(m) = \left|\mathcal{F}\big(x_h(n)w(n)\big)\right|^2 \tag{4}$$

Note that a weighting window $w(n)$ is applied before computing the DFT. Since the result is symmetrical, we store only $N/2 + 1$ points in *S*. Also note that the average value $\bar{x}_\mathrm{h}(\mathrm{n})$ can be subtracted before computation:

$$S_h(m) = \left|\mathcal{F}\big((\mathrm{x_h}(\mathrm{n}) - \bar{x}_\mathrm{h}(\mathrm{n}))w(n)\big)\right|^2 \tag{5}$$

Visualization Likewise, the visualization for *S* is similar to *R*, except that values in *S* are always positive and better represented on a logarithmic scale (dB).

2.2.2.3 Periodicity from Peak

Computation This feature takes as input the array resulting from one of the two periodicity-related features here above (Auto-correlation or PSD). For each frame of the input, it looks for a peak in a region that corresponds to acceptable values of periodicity (e.g. a human being cannot jump repeatedly at 100Hz but 2Hz would be reasonable). A peak is defined as a value in a frame that is larger than its 2*K* direct neighbors and above a given threshold. Results using the auto-correlation function were much better and smoother than with the PSD.

Visualization As presented in Figure 4, the periodicity can be visualized as a dot superimposed on the peak detected in the graph of the auto-correlation/PSD function, accompanied by an overlay text with the value of the periodicity (or the indication “no period” when no peak is found in the look-up region).

2.2.3 Ergonomics

Ergonomic features are related to bio-mechanical aspects of the movement. They are based on musculo-skeletal modeling of the body, aiming to describe the quality of movement in terms of comfort, robustness, or load.

2.2.3.1 Postural Load

Description The postural load of a body joint is an indication of the perceived stressfulness of a joint, according to its orientation, taking into account each degree of freedom (DoF) of the joint. The postural load of the whole body provides information on the comfort of a posture, and on the risk of injury. The postural load allows assessment of the comfort of a posture, and is widely used in industrial ergonomics, but it can also be used in performance assessment.

Computation The computation of the postural load requires mocap data containing joints orientations. The algorithm defined below is inspired from Andreoni et al. (2009) [22]. For each DoF of a joint, a relation exists between the perceived discomfort and the orientation of the joint. Tables of average perceived discomforts according to the orientation of a joint on a DoF have been set in the ergonomics literature to report these relations in [20] and [21].

The load, or stress of a joint S_j can be defined as the sum of independent perceived discomforts associated with each DOF of the joint S_{ij}. In our case, the perceived discomforts S_{ij} were spline-interpolated from the tables, as proposed in [22], in order to have continued values:

$$S_{ij} = f_{ij}(\theta_{ij}) \tag{6}$$

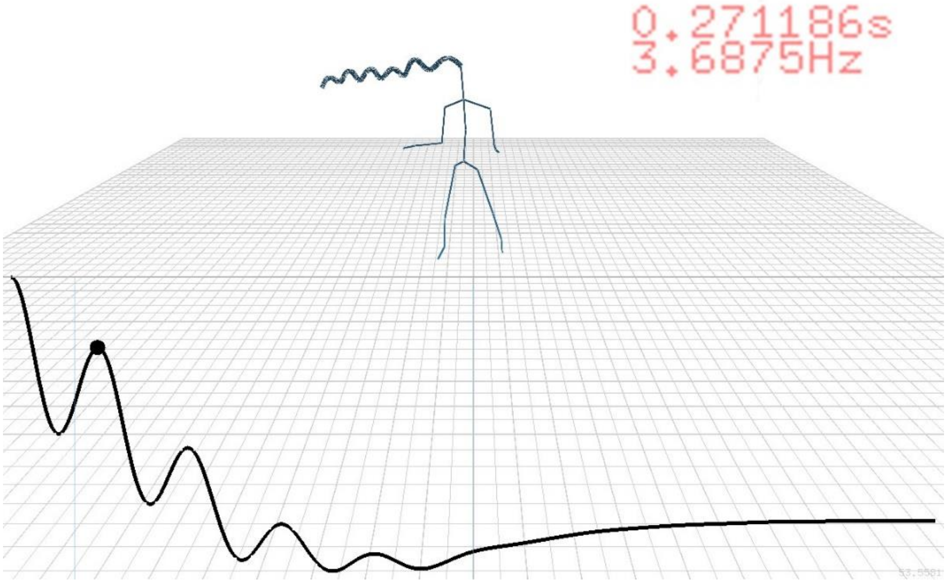


Figure 4 - The recorded movement is that of a body hopping repeatedly, as we can see in the trajectory of the head (shown attached to the head on graph). The auto-correlation function (black curve) of the Z position of the head presents a peak (black dot) at a position corresponding to the periodicity of the hops (0.27 s or 3.7Hz in this case).

where θ_{ij} is the orientation of the joint j on the axis i, and the function f_{ij} represents its relation based on spline functions and the aforementioned tables:

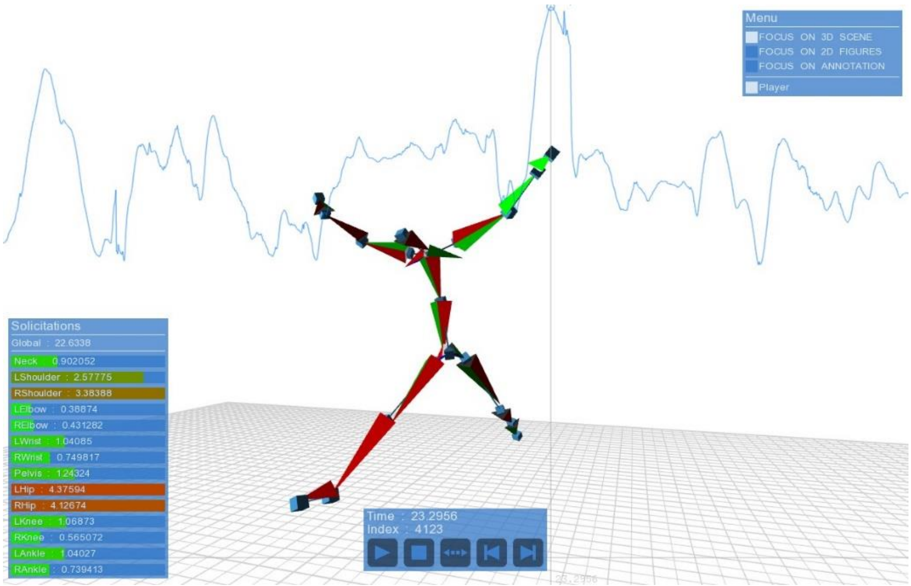
$$S_j = \sum_{i=1}^{m_j} S_{ij} \tag{7}$$

where m_j is the number of DoF of the joint j. The postural load results from the sum of each joint stress, as described in the previous equation:

$$PosturalLoad = \sum_{j=1}^{n} S_j \tag{8}$$

Andreoni et al. (2009) [22] proposed to define the postural load as a weighted sum of each joint stress, where the weight for each joint is proportional to the mass of the corresponding distal body district. However, as the perceived discomforts, i.e. feelings of pain, are supposed to inherently depend on the mass of the distal body district, we omitted these weights in the current study.

Visualization Figure 5 shows an example of the postural load analysis on a contemporary dance performance. In the 3D scene, the dancer is rendered at the most constraining pose. The 2D curve shows the computed postural load, and the peak corresponding to the maximal postural load. The colored sliders in the canvas view on the left display the stresses of each joint. In the case of this specific pose, the most stressed joints are the hips and the shoulders.



Solicitations
Global : 22.6338
Neck : 0.902052
LShoulder : 2.57775
RShoulder : 3.38388
LElbow : 0.38874
RElbow : 0.431282
LWrist : 1.04085
RWrist : 0.749817
Pelvis : 1.24324
LHip : 4.37594
RHip : 4.12674
LKnee : 1.06873
RKnee : 0.565072
LAnkle : 1.04027
RAnkle : 0.739413

Time : 23.2956
Index : 4123

Figure 5 - An illustration of the postural load visualization in MotionMachine.

2.2.3.2 Sphereness

Description The sphereness is a novel exploratory feature inspired from a concept of Taijiquan. In the philosophy of Taijiquan, the body and the movements are represented through mental images, to enhance the transmission of the sensations of the art of Taijiquan from the teacher to the pupil. The sphere of the body is one of these images. The body is surrounded by an imaginary sphere, and controls the sphere with its movements. This image fosters the idea of moving the body as a whole rather than individuated segments. Empirical principles of Taijiquan rule the way the body interacts with the sphere: the body must fit the shape of the sphere, and the sphere must not be too wide nor too small. These empirical concepts can be linked to more concrete physical principles. The spherical shape of the body can be related to robustness, or to notions of balance, and the restrictions on the sphere expansion may be linked to Hill's muscle model (the force-length relationship), explaining that the muscles have less power when they are too expanded or too crouched. The sphereness feature tends to describe these relations between the body and the sphere.

Computation The sphereness is computed through three steps:

1. the body center of mass (CoM) is computed and is defined as the center of the sphere. See Equation 1 of Section 2.1) for its computation details;

2. the Euclidean distance (d_J is calculated between each one of the five end-effectors of the body (head, hands and feet) and the body CoM:

$$d_J = \left|\overrightarrow{p_J} - \overrightarrow{p_{COM}}\right| \qquad \textbf{(9)}$$

For $J = \{head, left\ hand, right\ hand, left\ foot, right\ foot\}$;

3. the mean and the standard deviation of these distances are processed:

$$r = \frac{\sum_{J=1}^{5} d_J}{5} \tag{10}$$

$$\sigma = \sqrt{\frac{\sum_{J=1}^{5} (d_J - r)^2}{5}} \tag{11}$$

The radius *r* of the sphere is defined as the mean of these distances, and indicates if the body is expanded or crouched. The standard deviation *σ* indicates the deviation of the body from the spherical shape, i.e. its *sphereness*.

Visualization Figure 6 shows an example of sphereness visualization on a Taijiquan exercise. The pose shown on the left is a rest pose, and the pose shown on the right is the basic Taijiquan posture, also called the "Tree posture". The 3D scene shows two spheres. The big one is the aforementioned sphere, whose radius is the mean of the five distances of the five end-effectors to the CoM. The radius of the small sphere is the radius of the first one minus the deviation. In the 2D scene, the upper curve displays the radius of the sphere, and the lower curve displays the deviation. We can observe that the Taijiquan posture is as expanded as the rest pose, but is much closer to a spherical shape.

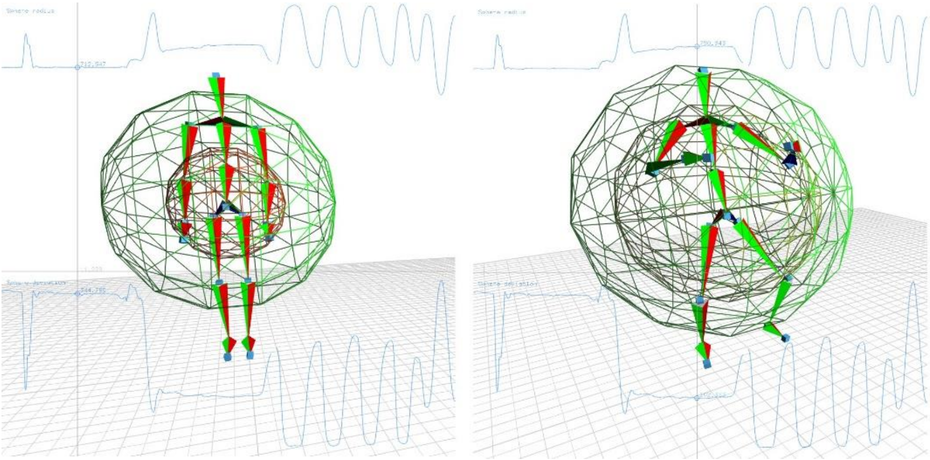


Figure 6 - An illustration of the sphereness visualization in MotionMachine.

2.2.4 Effort (Laban)

Effort features, derived from the Laban Movement Analysis (LMA) [4], describe the quality of the motion in various dimensions: dynamics, shape, space, expression, etc. Different categorizations can be found in literature but we focused on three of them for the implementation: weight, time and space. Several effort features take the kinetic energy feature as their starting point.

2.2.4.1 Kinetic Energy

Computation We compute an estimation of the kinetic energy E^k of a body part *k* (we can have one value per skeletal joint) as the half of the square of its velocity:

$$E^k(t_i) = \frac{1}{2} v^k(t_i)^2 \tag{12}$$

with $v^k(t_i)$, the velocity of the body part k at time t_i. The kinetic energy of the whole body is determined as a weighted sum for all the body parts:

$$E(t) = \sum_{k \in K} \alpha_k E^k(t_i) \tag{13}$$

with α_k, the normalized weight associated to body part k [16]. Figure 7 shows the whole kinetic energy of the body during one performance, and kinetic energies corresponding to all the joints of a given skeletal model.

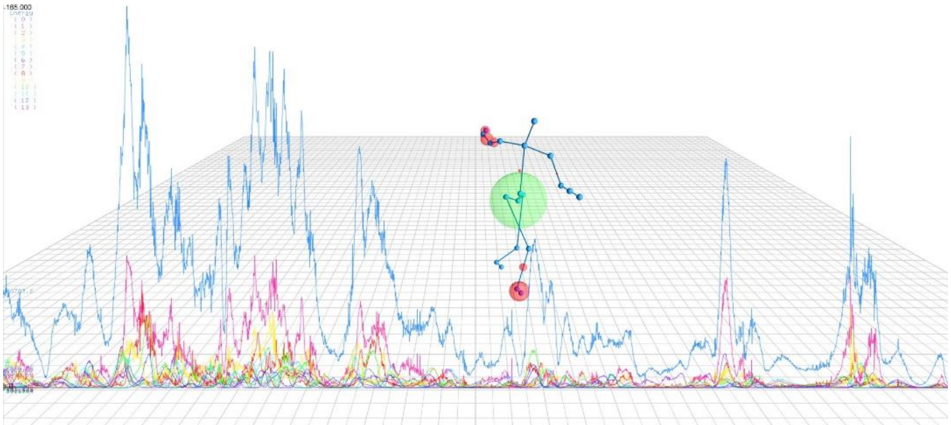


Figure 7 - Kinetic energy: Blue: Whole body. Other colors: different body parts.

2.2.4.2 *Weight Effort*

Description The weight effort refers to physical properties of the motion. In LMA, the two opposite weights for a motion can be *strong* or *light*.

Computation The weight effort is estimated by computing the maximum of kinetic energy over a defined time interval [24], as shown in the equation below:

$$WeightEffort(T) = \max_{\text{i=1:T}}(E(t_i)) \tag{14}$$

with T, the size of the analysis window. Figure 8 shows the weight effort values computed for different window sizes: 0.1*s*, 0.5*s*, 1*s* and 5*s*.

2.2.4.3 *Time Effort*

Description The time effort represents the sense of abruptness and change over time. A motion having a high time effort is defined as *sudden* or *urgent*. A motion having less time effort is *sustained* or *steady*.

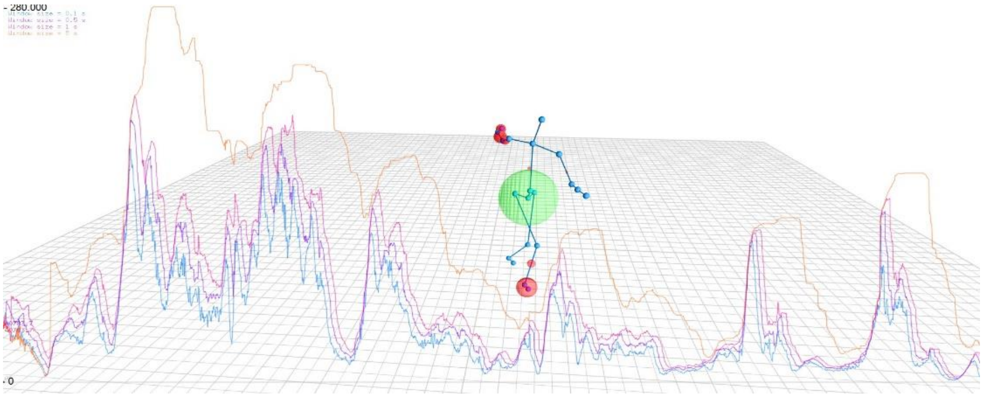


Figure 8 - Weight Effort: blue: T=0.1s, purple: T=0.5s, pink: T=1s, orange: T=5s.

Computation The time effort is estimated by computing the weighted sum of the accelerations over time for a given body part (can be skeletal joint):

$$TimeEffort^k(T) = 1/T \sum_{i=1}^{T} a^k(t_i) \quad \textbf{(15)}$$

where $a^k(t_i)$ is the acceleration of the body part k at time t_i. The time effort for the whole body is computed as a weighted sum of time efforts of body parts:

$$TimeEffort(T) = \sum_{k \in K} \alpha_k \,.\, TimeEffort^k(T) \quad \textbf{(16)}$$

Figure 9 shows the time effort values computed, for the different body parts and the whole body, for different window sizes: 0.1*s*, 0.5*s*, 1*s* and 5*s*.

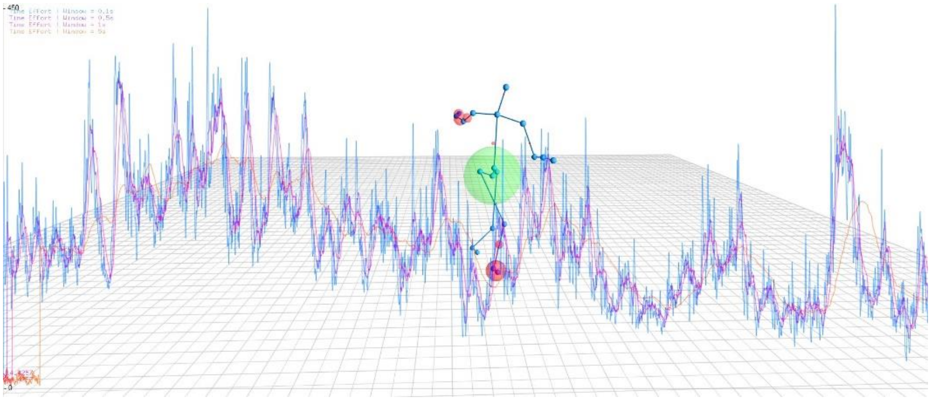


Figure 9 - Time Effort: blue: T=0.1s, purple: T=0.5s, pink: T=1s, orange: T=5s.

2.2.4.4 Space Effort

Description The space effort defines the directness of the movement. By computing the space effort, we can say that a movement is considered as *direct* (focused and toward a particular spot) or *indirect* (multi-focused and flexible).

Computation For one body part *k*, we compute the space effort as follows:

$$SpaceEffort^k(T) = \sum_{i=2}^{T} \frac{\|x^k(t_i) - x^k(t_i - 1)\|}{\|x^k(T) - x^k(1)\|} \quad \textbf{(17)}$$

where x^k are the relative 3D positions associated to the k^{th} joint with $1 \leq k \leq m$. m is the number of joints.

Similarly to other metrics, the space effort for the whole body is computed as a weighted sum of space efforts on the different body parts:

$$SpaceEffort(T) = \sum_{k \in K} \alpha_k \, . SpaceEffort^k(T) \tag{18}$$

Figure 10 shows the space effort computed, for the different body parts and the whole body, for different window sizes: 0.1*s*, 0.5*s*, 1*s* and 5*s*.

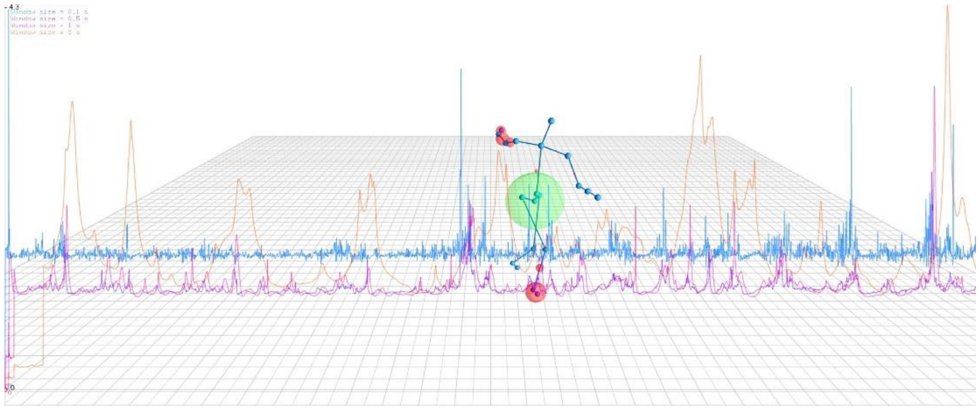


Figure 10 - Space Effort: blue: T=0.1s, purple: T=0.5s, pink: T=1s, orange: T=5s.

2.2.5 Space (Laban)

Space features of Laban Movement Analysis [4] describe the relation between the body and the environment. In the scope of this research, two features were developed: the covered distance, and the covered area.

2.2.5.1 Covered Distance

Description This feature refers to the accumulated distance that is covered on the ground over a period of time by a given body part or the whole body.

Computation The covered distance can be processed on any node of the body, though the pelvis is used by default, in order to analyze global displacement of the body on the scene. The body CoM could also be used. The node is first projected on the ground. The cumulative sum is then processed on the lengths of the segments linking two successive positions of the projected node:

$$\overrightarrow{p_\iota} = (x_i, y_i) \tag{19}$$

$$d_i = |\overrightarrow{p_\iota} - \overrightarrow{p_{\iota-1}}\,| \tag{20}$$

$$D_i = \sum_{j=1}^{i} d_i \tag{21}$$

where $\overrightarrow{p_\iota}$ is the projection on the ground of the node, d_i the distance between two successive frames, and D_i the accumulated distance at frame i.

Visualization Figure 11 shows a visualization of the Laban space component in MotionMachine. In the 3D scene, the trace of the pelvis on the ground is displayed in gray, and its convex hull (covered area), in red. In the 2D scene, the upper curve is the covered distance, the lower curve the covered area.

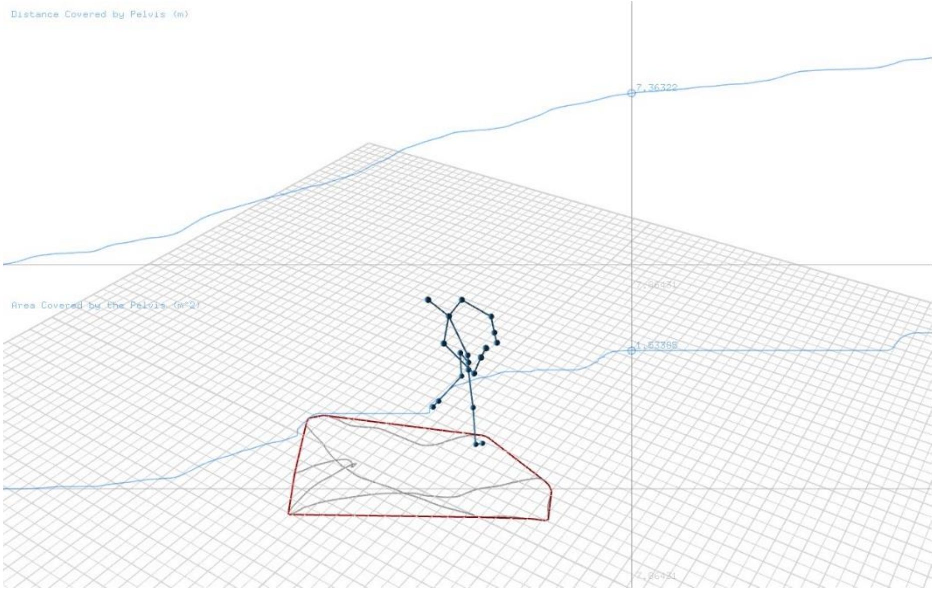

Distance Covered by Pelvis (m)
7.36322
Area Covered by the Pelvis (m^2)
1.63388

Figure 11 - Laban space component visualization in MotionMachine.

2.2.5.2 Covered Area

Description This feature refers to the area that is covered on the ground over a period of time. It describes how widely the scene is explored by the body.

Computation The covered area is based on the convex hull, presented in Section 2.1. The convex hull is calculated on the trace of the pelvis projected on the ground. The area is then computed on the points of the resulting convex hull, with an algorithm derived from Findley’s polygon area algorithm [18].

Visualization The covered area is represented in Figure 11, by the lower curve in the 2D scene, and the red convex hull on the ground in the 3D scene.

2.2.6 Smoothness/Jerk

The jerk is a kinematic descriptor corresponding to the rate of change of the movement acceleration. Low values of jerk correspond to smooth movements, while high values appear for complex, discontinuous movements.

2.2.6.1 Jerk Vector

Description For each frame, a jerk vector can be determined. It corresponds to the evaluation of the rate of change in each of the *xyz* coordinates of each joint.

Computation For one joint *k*, it can be computed as:

$$j^k(t_i) = \frac{x^k(t_{i+2}) - 2.x^k(t_{i+1}) + 2.x^k(t_{i-1}) - x^k(t_{i-2})}{2.\Delta t^3} \qquad \textbf{(22)}$$

where x^k are the relative 3D positions associated to the k^{th} joint with $1 \leq k \leq m$. m is the number of joints.

2.2.6.2 Jerk Magnitude

Description From the jerk computed on each joint and each dimension – therefore a jerk vector – we will preferably compute a scalar value for each joint. This value represents the overall jerkiness of a given joint.

Computation The jerk magnitude is the length of the jerk vector:

$$j^k t_i = \sqrt{j_x^k(t_i)^2 + j_y^k(t_i)^2 + j_z^k(t_i)^2} \tag{23}$$

2.2.6.3 Flow Effort

Description The flow effort describes the continuity of the movement over time. Low values describe fluid movement, high values for restrained movement [19].

Computation For one joint *k* and a time window of length *T*, it is computed as:

$$Flow^k(T) = \frac{1}{T}\sum_{i=1}^{T} j^k(t_i) \tag{24}$$

For a set of joints K, we just aggregate the flow efforts of all joints *k* ∈ K:

$$Flow(T) = \sum_{k \in K} \alpha_k . Flow^k(T) \tag{25}$$

2.2.7 HMM Decoding with HTK

Description In previous works, we detailed how it is possible to implement a gesture decoding system based on Hidden Markov Models (HMMs) [23]. The HMMs represent the gesture with a succession of states. At each state, local statistics of the observations apply and both local statistics and state transition probabilities are determined by training. Hence it is possible to identify the most likely gesture corresponding to new observations and compute an approximation of the current state, which reflects the progression in the executed gesture.

Computation The HMMs used for gesture recognition are trained using HTK (Hidden Markov Model Toolkit [14]). Each input data frame contains the body skeleton pose data. We can choose to describe this pose with the position of skeleton nodes or the orientation of the bones. The gesture recognition is performed by using a Viterbi algorithm. We used an implementation proposed by the *MLPack* library [12]. The Viterbi algorithm is applied on the whole temporal window of data stored in a MotionMachine Track data type. Figure 12 shows gesture decoding of a dance step model. The purple line represents the actual state and the blue line represents the decoded gesture.

2.2.8 DTW/Muller Gesture Recognition

This method of gesture recognition has been initially proposed by Müller [5] who decomposes a movement into a set of relational characteristics between joints of a skeleton, reflecting simple human body motions. Binary features are then converted from one value per frame to one value per segment. Forty features are extracted and can be interpreted as a summary of the mocap data under the form of characteristics understandable from an anatomical point of view.

For each gesture to recognize, we hence know which characteristics are active/inactive and their evolution over time. A motion template is defined by computing the mean of the relational features based on several executions of the same gesture. This motion template hence highlights the zones where the relational features differ between several occurrences of the same gesture (in black in the Figure 13) and the zones where they are the same (in white in the Figure 13), which can be considered as a definition of the gesture to be recognized. Zones in orange are considered are zones of uncertainty.

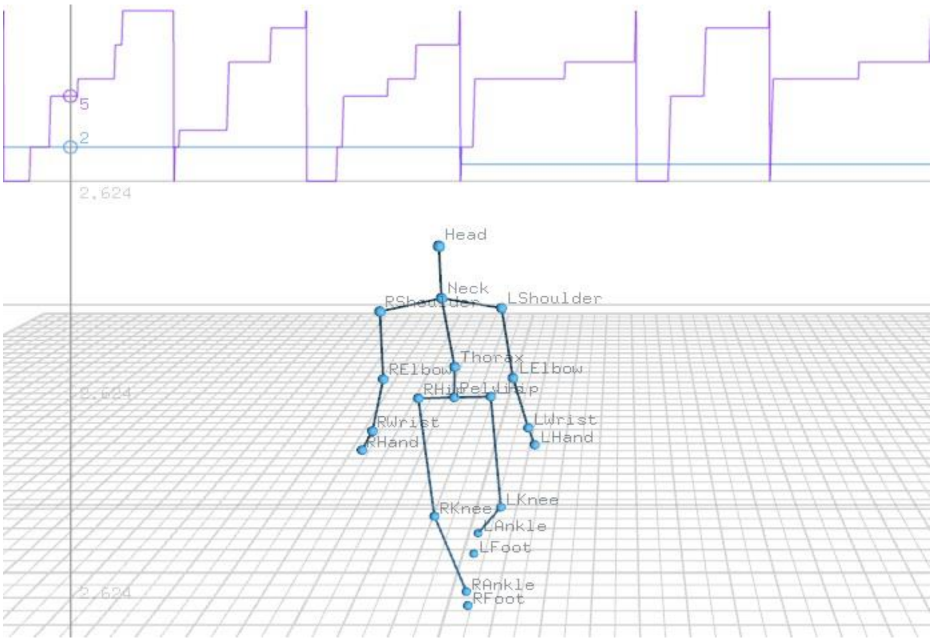


Figure 12 - HMM-based gesture decoding of a dance step sample. Four HMMs for four different dance steps were trained with HTK. Each gesture was modeled with 10 states. On the top of the Figure, we can see the decoded gesture and the most likely current state. The cursor is pointing at a frame decoded as gesture 2 and state 5.

The last step consists in measuring the similarity between the template X of size K and the motion to be compared, Y of size L (with $L > K$) or a continuous movement sequence. A distance between the motions is measured based on features outside of the uncertainty zone of the motion template.

$$\begin{cases} D(k,l) = c(k,l) + \min\left(D_{k-1,l-1}, D_{k-2,l-2}, D_{k-1,l-2}\right), k \in [3:K], l \in [3:L] \\ D(k,l) = c(k,l) + \min\left(D_{k-1,l-1}, D_{k-1,l-2}\right), k \in [1:2], l \in [3:L] \\ D(k,l) = c(k,l) + \min\left(D_{k-1,l-1}, D_{k-2,l-1}\right), k \in [3:K], l \in [1:2] \end{cases} \quad \textbf{(26)}$$

where $c(k,l)$ is the normalized Manhattan distance

$$c(k,l) = \frac{1}{I(k)} \sum_{i \in I(k)} |X(k)_i - Y(l)_i| \quad \textbf{(27)}$$

$I(k)$ is a vector containing the indices i of the features outside the uncertainty zone. Figure 14 shows the distance between the motion template and a full dance motion sequence (in blue) and intervals recognized as corresponding to the motion template (highlighted in green).

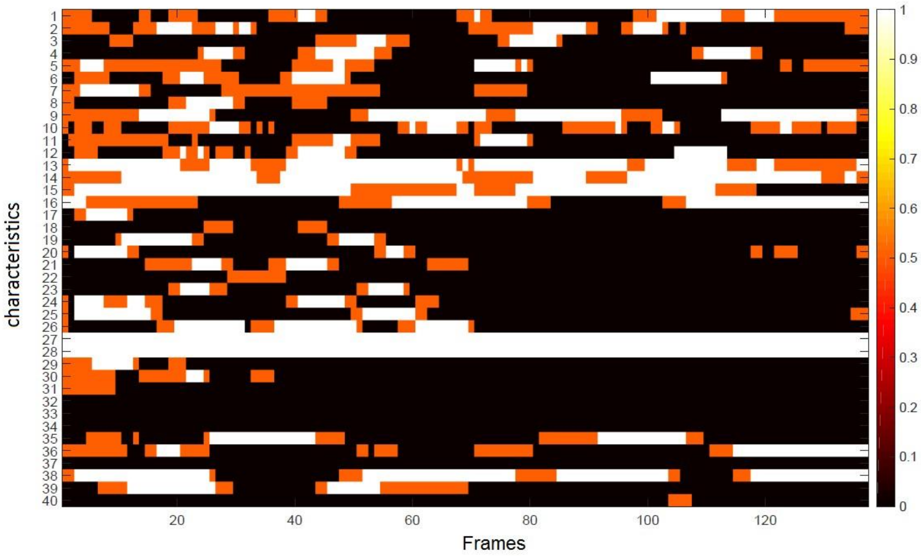


Figure 13 - Example of the motion template of a dance sequence.

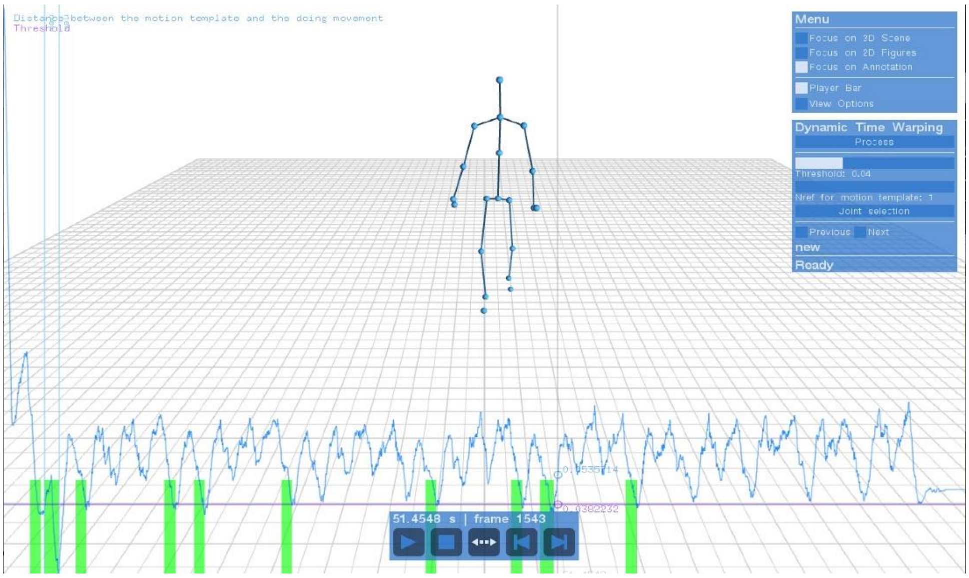


Figure 14 - DTW-based gesture recognition in MotionMachine. The similarity score between the considered motion template (forward dance step) and a dance motion sequence is represented as a blue line, the threshold is represented as a purple line and the time intervals recognized as similar to the motion template are shown in green.

2.2.9 Wekinator

Additionally to these motion features internally implemented in the *MotionMachine* framework, we established a link to another tool, *Wekinator*. *Wekinator* is a standalone software for applying supervised machine learning algorithms interactively to real-time applications. Its integration into *MotionMachine* allows interactive machine learning capabilities, including classification, gesture following and recognition. *MotionMachine* and *Wekinator* communicate with the OSC protocol. *Wekinator* algorithms are trained by the input data sent by *MotionMachine*. These data may be either motion raw data, or motion features. The results of the machine learning process are then sent back to *MotionMachine*, and can be considered as new motion features.

2.3 Statistical factors and features exploration

The analysis of a motion database is a complex task. Firstly, movements generate highly dimensional data, and their processing is a necessary step in order to extract relevant features of fewer dimensionality, easier to analyze and understand. Secondly, according to the size of the dataset, a large number of inter-samples or inter-individuals factors can arise, which have different effects on the studied motion features. To address that issue, and reduce the complexity of motion data analysis, we developed a statistical analysis module in the *MotionMachine* framework, allowing an exploration of the effects between stylistic factors and motion features. In the following, we first explain the general structure of the module in Section 2.3.1. We then focus on the factors classification in Section 2.3.2. Features selection, and statistical processing are then respectively explained in Sections 2.3.3 and 2.3.4. Finally, we briefly present the rendering and interaction functionalities we included in the module, additionally to the basic functionalities inherent to the *MotionMachine*.

2.3.1 General structure

Figure 15 displays the general structure of the module. The input data required by the program are the MoCap data, and a factors table. From the MoCap data, motion features are extracted, and are then processed by the statistical module. Firstly, the data are classified into groups according to the factor to analyze (see Section 2.3.2). From these groups, different statistics are then processed, and histograms and box plots are generated within groups, and between groups.

The layered visualization module of the *MotionMachine* framework allows superimposed view and interaction with different types of data. Firstly, the *3D scene* layer allows 3D rendering of the motion data. Secondly, the *2D scene* layer displays graphs corresponding to the features extracted from the 3D motion data. Finally, the statistical results are rendered in the *Annotation* layer.

The control and interaction with the program is made easy and intuitive, thanks to the *User Interface* layer, consisting of graphical commands, allowing features and factors selection, statistical process, and navigation between the different results

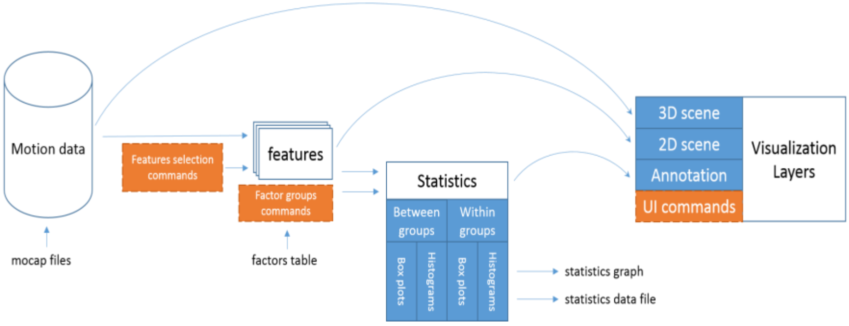


Figure 15 - Statistical exploration module general block diagram

2.3.2 Factors classification

Many variables, or factors, may have an influence on the way a performer moves. These factors can be intra- or interindividual. Interindividual factors may be of various types, including among others:

- Social factors: culture, education, etc.
- Psychological factors: mood, state of concentration, etc.
- Physiological factors: gender, age, size, mass, force, suppleness, etc.
- Psychophysiological factors: handedness, motor control, etc.

Intra-individual factors, on the other hand, are related to the performance, such as the type of dance, or exercise.

All these factors may affect, independently or not, motion features. By means of statistical analysis, the module proposed in this project allows to investigate which features are influenced by each factor.

The information about factors of each sample of the dataset is provided to the program thanks to a specific file type, readable by the module: the factors table. This table, stored in a standard file (*.csv* extension), can be read and edited by any standard spreadsheet editor (Microsoft Excel, OpenOffice Calc, etc.). A dummy factor table is shown as an example in Table 1.

Table 1 – Dummy factors table

	Factor1 (gender)	**Factor2 (dance)**	**Factor3 (experience)**	**Factor4 (culture)**	**…**
File1	0 (male)	0 (rock’n roll)	3 (years of practice)	0 (Occidental)	…
File2	1 (female)	0 (rock’n roll)	3	0 (Occidental)	…
File3	0 (male)	1 (classic)	10	1 (Oriental)	…
File4	1 (female)	1 (classic)	10	1 (Oriental)	…
…	…	…	…	…	…

Each line corresponds to a sample of the dataset, and each column to a factor. For each sample, an index corresponding to each factor is given, allowing automatic classification by the program, for each factor. For instance, to analyze the first factor (gender), the program will create two groups, one containing performances rendered by a woman, and the other containing performances rendered by a man.

2.3.3 Features selection

The module allows analysis of any motion feature implemented in the *MotionMachine* framework. All these features are presented in details in Section 2.2. According to the type of motion data, different features may be more relevant than others. For instance, Laban movement analysis was originally developed for dance gestures, and more broadly, expressive gestures. On the other hand, ergonomics features are more meaningful to analyze athletic gestures. In order for the user to choose which features to analyze, graphical user commands were embodied in the module (see Section 2.3.5.2).

2.3.4 Statistical process

From the groups of data classified according to the selected factor, statistics are processed on the selected motion features. The results are presented under two different graphical representations: the box plot and the histogram (see Section 2.3.4.1). In order to have comparable results between features, the results are mapped between dimensionless limits (see Section 2.3.4.2), and a ratio between statistical attributes is proposed to quantify factors effects on motion features (see Section 2.3.4.3).

2.3.4.1 Representation

Basic statistics were extracted from the data to estimate and represent their statistical distribution: the box plot, and the histogram.

An example of box plot as rendered by the module is shown in Figure 16. The box plot is a convenient graphical representation of the profile of statistics data through quartile values. The box, between quartiles 25 and 75, indicates the interval containing the central 50% of the distributed data, and allows straightforward comparison with other distributions.

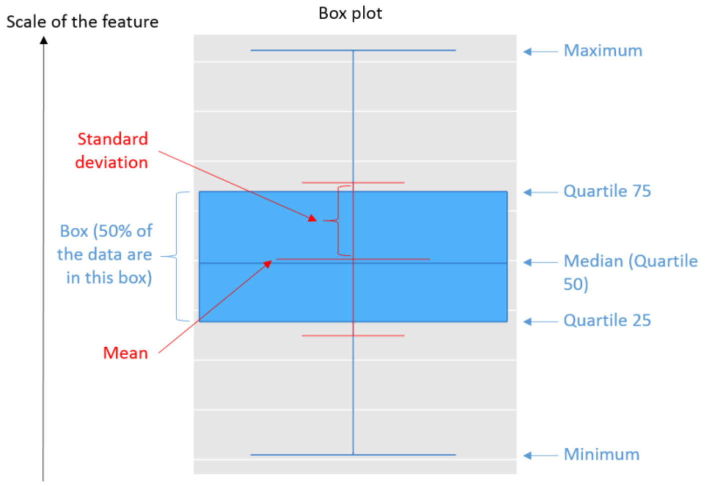


Figure 16 - Box plot example

An example of histogram is presented in Figure 17. The histogram provides an approximation of the profile of the data distribution. For both representations, the mean and standard deviation are represented as additional information.

The process of these basic statistics within groups and between groups allows a first exploration of the effects between features and factors. For group comparison, the motion features of each sample of a group are concatenated, and global box plots and histograms are generated for each group. On the other hand, for comparison of samples within a group, a box plot and a histogram is generated for each sample.

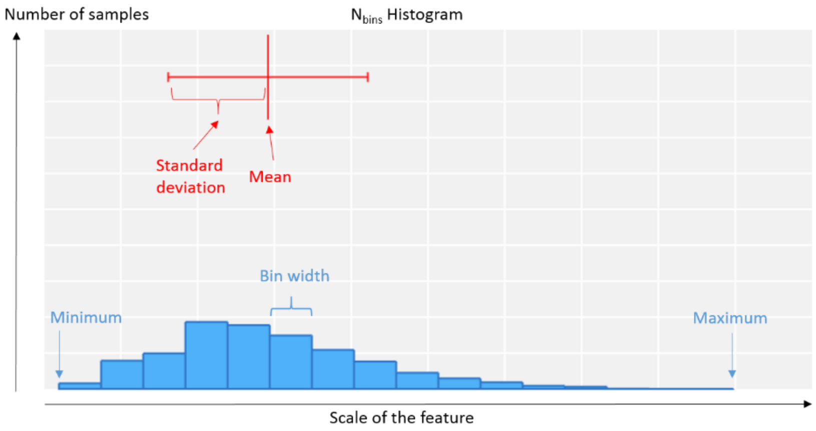


Figure 17 - Histogram example

2.3.4.2 *Standardization*

Motion features are defined in various measurement units and scales, and their direct comparison is irrelevant. Nonetheless, it is of great interest to know which feature allows better discrimination of different factors. The data to compare being the statistical results (and not the motion features themselves), we standardized these results as new dimensionless values. For each feature, we processed the global mean (μ_{db}) and standard deviation (σ_{db}) on the database, and the standardized results (X_{std}) were computed according to the following equation:

$$X_{std} = \frac{X - \mu_{db}}{\sigma_{db}} \quad \textbf{(28)}$$

For each statistical attribute (X), i.e. the moments, quartiles and extrema.

2.3.4.3 *Distance Index*

The box plots and histograms visualization in the *MotionMachine* framework enables observational comparisons between groups and samples, for different factors and different motion features, allowing an exploration of the effects between them. However, it can be valuable to extract from these box plots an index, in order to quantify these effects. To that end, a distance between statistical results for different states of the factor is proposed:

$$\Delta_{X,f_a f_b} = X_{std,f_a} - X_{std,f_b} \quad \textbf{(29)}$$

where f is the factor analysed (e.g., the culture), f_a and f_b are two different states of the factor (e.g., Occidental and Oriental), X is a statistical attribute (mean, median, quartile 75, etc.), and $\Delta_{X,f_a f_b}$ is the distance between states f_a and f_b, on statistical attribute X.

The maximal distance for a factor corresponds to the states providing the maximal and minimal statistical attributes. This maximal distance will be simply written as Δ_X in the rest of the document for notation simplicity:

$$\Delta_X = \max_{a,b}(\Delta_{X,f_a f_b}) \quad \textbf{(30)}$$

One maximal distance can be computed for each statistical attribute. For this analysis, the moments and quartiles were retained, and extrema were omitted. Indeed, in the case of the extrema, maxima and minima can be fully determined by any unusual peak in the feature, and hence do not globally represent the feature. As a final index to quantify factor effects, a mean of these distances was chosen:

$$\Delta = \frac{\Delta_{mean} + \Delta_{std} + \Delta_{median} + \Delta_{Q75} + \Delta_{Q25}}{5} \tag{31}$$

This index can be considered as a global distance between two box plots, taking moments and quartiles into account. As it is computed with standardized data, it is expressed as a ratio of the global standard deviation of the database (σ_{db}). For instance, $\Delta = 0.5$ means that the distance between box plots is $0.5 * \sigma_{\mathrm{db}}$, i.e. half of the global standard deviation of the database. This distance index is an indication of the importance of the factor influence on a motion feature, and will be expressed as the “effect size” in the rest of the report.

2.3.5 Interface

2.3.5.1 Display

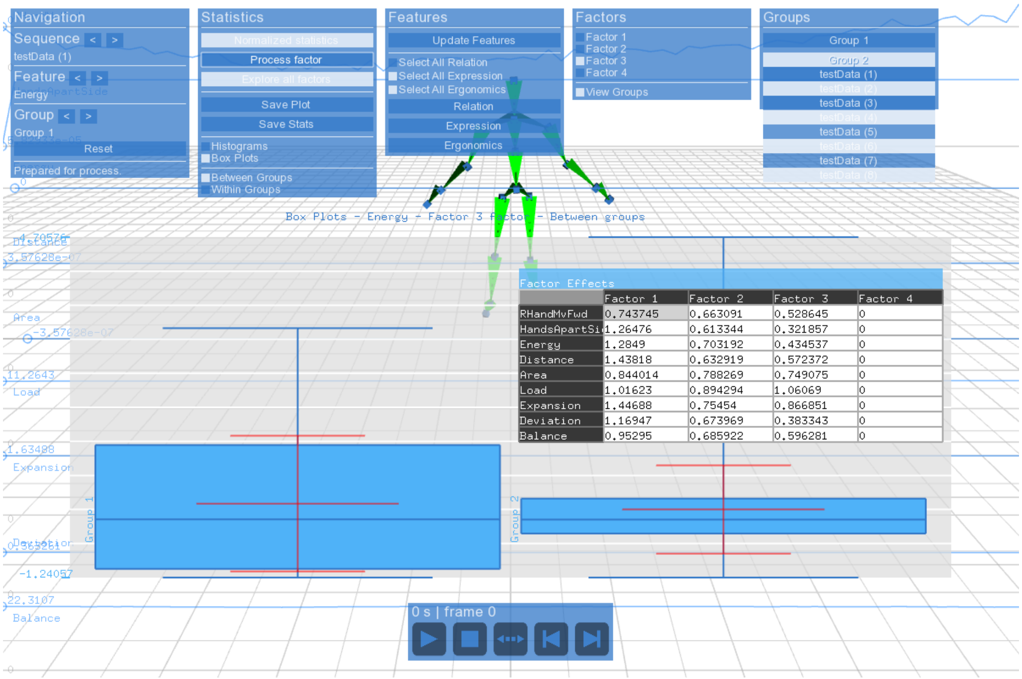
The application generates a large amount of information that is complicated to visualize simultaneously. The superimposed layered visualization of the *MotionMachine* framework is used to solve that issue. As explained in Section 2.3.1, motion data, motion features and statistics are displayed in three different layers, and the graphical commands are rendered in a fourth one. The basic controls of *MotionMachine* allow displaying of certain layers only. Figure 18 shows an example of the application graphical interface, where each layer is displayed.

2.3.5.2 Interaction

The interface makes the use of the program easy and intuitive. First of all, in addition to a basic loading functionality at the start of the program, the data (motion data and factors table) can be loaded with a simple drag and drop action. Furthermore, different command tabs allow the control of the application workflow. The *Feature* tab consists of checklists for each feature category, which can be used to select the motion features to analyze. The *Factors* tab is used to choose the factor to consider for the sample classification. When a factor is selected, the *Groups* tab automatically classifies the data into groups, according to the factors table. Additionally, the dropdown checklists of the *Groups* tab make possible the edition of custom groups.

As soon as the aforementioned tabs are configured, the *Statistics* tab is used to start the process, and allows navigation between the different results (box plots, histograms, within and between groups). Finally, the *Navigation* tab allows exploration between these results for the different motion features and the different groups.

Additionally, the button *Explore all factors* allows the process for all factors and all selected motion features. An overview of all results is then displayed in a floating table, where each value stands for the effect of a factor on a motion feature (see Figure 18).



Factor Effects

	Factor 1	Factor 2	Factor 3	Factor 4
RHandMvFwd	0.743745	0.663091	0.528645	0
HandsApartSi	1.26476	0.613344	0.321857	0
Energy	1.2849	0.703192	0.434537	0
Distance	1.43818	0.632919	0.572372	0
Area	0.844014	0.788269	0.749075	0
Load	1.01623	0.894294	1.06069	0
Expansion	1.44688	0.75454	0.866851	0
Deviation	1.16947	0.673969	0.383343	0
Balance	0.95295	0.685922	0.596281	0

Figure 18 - Interface of the statistical exploration module

2.4 Results

2.4.1 Use case 1: Factorial analysis of improvised contemporary dance

During this project, a database of contemporary dance was recorded, in order to explore relationships between music style and a dancer's way of moving. To that end, six professional dancers from the dance school P.A.R.T.S[3] were recorded while dancing on six different music pieces.

First, each dancer rendered five improvisations, each with a duration of 45s[4], on five music pieces picked in an emotionally tagged music database. The tags of these music were defined on two scales: arousal (calm to powerful), and valence (sad to happy) [12]. The five picked music pieces were chosen so that each one carries an extreme emotion on the valence-arousal diagram (see Figure 19). In addition to these improvisations, the dancers also performed a choreography proposed by a professor of P.A.R.T.S.

All the data were recorded with the *Qualisys*[5] motion capture system, at a framerate of 175 FPS. For each session, a total of 68 reflective markers were placed on the entire body, and were captured by 11 infrared cameras. From these markers trajectories, 20 anatomical landmarks trajectories were extracted with *Visual3D*[6].

[3] P.A.R.T.S : www.parts.be

[4] Actually, the analysed pieces are 35s long, as the first 10 seconds were omitted to take into account the fact that some dancers listened without moving the first seconds to appropriate the music.

[5] Qualisys: http://www.qualisys.com/

[6] Visual3D (C-Motion): http://www.c-motion.com/products/visual3d/

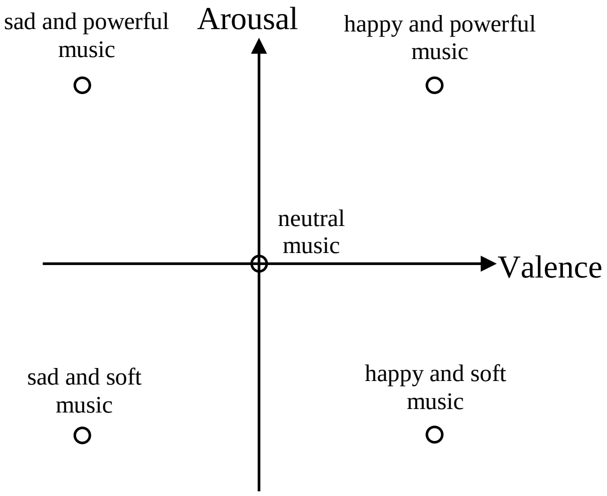


Figure 19 – Valence-Arousal emotional diagram

From the database of 36 samples (6 dances rendered by 6 dancers), we established a factors table including six factors:

- Gender (women – men)
- Dancer (1 – 2 – 3 – 4 – 5 - 6)
- Music (1 – 2 – 3 – 4 – 5 – 6)
- Arousal[7] (positive – negative – neutral)
- Valence (positive – negative – neutral)
- Improvisation (yes – no)

An exploratory analysis was then conducted on these data through the module presented in Section 2.3.

Among all the possible motion features, a subset of features was selected for the analysis:

- Effort Laban features: kinetic energy
- Space Laban features: covered distance and area
- Ergonomics features: postural load, and sphere radius and deviation

The graphical results of the analysis are provided in Appendix 5. The following results are based on observation of these graphs and on the extracted ratios, and serve as an exploration of the database. They express possibilities of factors effects on motion features, without any statistical confidence test. For more significant results, a more complex analysis with confidence tests must be conducted. Nonetheless, an arbitrary criterion on the effect size (i.e. the global distance between extreme box plots) can be set, to validate or not an effect of a factor on a feature. In the following, we choose two arbitrary thresholds in order to classify the different results. For effect sizes lower than $0.25 * \sigma_{db}$ $(\Delta < 0.25)$, we consider that there is no significant factor effect on the considered motion feature. For indices higher than $0.25 * \sigma_{db}$ but lower than $0.5 * \sigma_{db}$,

[7] For arousal and valence factors, we must note that only 30 samples are considered, as the choreography was not emotionally tagged.

we consider that there is a small effect. Finally, for $\Delta > 0.5$, we consider that there is a big effect.

2.4.1.1 Kinetic energy

Graphical results concerning the kinetic energy can be found in Appendix 5. First, Figure 40 shows each sample box plot. We can observe that there are a lot of differences between the samples. The factorial analysis allows us to investigate which factors may be partly responsible of these differences.

Gender does not seem to have an effect on kinetic energy, as shown in Figure 41. The two displayed box plots are similar according to the criterion ($\Delta = 0.13$).

On the other hand, the dancers seem to have differences concerning kinetic energy ($\Delta = 0.81$) (see Figure 42). For instance, the mean kinetic energy of Dancer 2 (0.7 J/kg[8]) is about twice higher than that of Dancer 4 (0.29 J/kg).

Concerning the music factor, the extreme music pieces are Music 4 and Music 3, with a global distance of $\Delta = 1.07$. This value indicates high differences between the music pieces concerning kinetic energy. It is interesting to note that Music 3 is the calm and sad music, while Music 1, which is on the opposite of the arousal-valence diagram (powerful and happy) has high results, almost as high as Music 4 (see Figure 43). Music 4 is actually the choreographed music.

The independent effects of both arousal and valence factors can be observed in Figure 44 and Figure 45 respectively. Both factors seem to have an effect on kinetic energy, independently of the dancer (for arousal: $\Delta = 0.35$, for valence: $\Delta = 0.66$). More kinetic energy seems to be spent on powerful music, and on happy music. Note that there is no clear difference between neutral and negative values for arousal.

Finally, concerning the improvisational factor, we can observe an effect: $\Delta = 0.64$. However, this result clearly depends on the only one choreography and associated music that was proposed. If we look back to the music factor, we can note that the energy spent in Music 1 (the powerful and happy music) is similar to the energy spent on the choreography.

2.4.1.2 Covered distance

Covered distance and area are cumulative positive features, and are inherently monotonically increasing functions. The interpretation of their statistics can hence be different. For instance, the maximum of covered distance in a sample is obviously obtained at the end of the sequence, as this value gives the actual total distance travelled. The feature itself could be considered as one value only for each sample (the total distance), instead of a continuous feature. The box between quartiles 25 and 75 only indicates if the dancer travelled more at the beginning or the ending of the performance. Hence, besides standard box plots, a more relevant statistical value concerning these specific cumulative features is the mean of each sample maximum in a group (i.e. the mean total distance for the group). This attribute will be used in the following for distance computing (Δ_{dist}). The global mean and standard deviation in the whole database on this attribute are respectively: $\mu_{db} = 18.37m$ and $\sigma_{db} = 7.71m$.

[8] The used formula for kinetic energy estimation is E(t) = ½.v(t)² (see Section Kinetic Energy 2.2.4.1) and is independent of the mass.

Table 2 shows the mean total covered distance per gender, which are close to each other. The distance between them is thus:

$$19.04m - 17.56m = 1.48m$$

This distance is small compared to the global standard deviation of the database, resulting in a normalized distance of:

$$\Delta_{dist} = \frac{1.48}{7.71} = 0.19$$

We cannot validate an effect of the gender on covered distance, according to our criterion.

There are noticeable differences between dancers according to the covered distance, showing a possible effect between that factor and that feature ($\Delta_{dist} = 1.3$). Dancer 6 and 1 are the most travelling dancer, with a mean total distance almost twice higher than that of Dancer 4 and 5.

Table 2 - Mean total distance per gender

Women	Men
17.56m	19.04m

Table 3 - Mean total distance per dancer

Dancer 1	Dancer 2	Dancer 3	Dancer 4	Dancer 5	Dancer 6
21.51m	20.72m	18.15m	13.39m	13.03m	23.03m

Concerning the music factor, results (see Table 4) are similar to those for kinetic energy ($\Delta_{dist} = 1.74$). Musics 1 and 4 give again the highest results, about twice higher than those of Music 3. More generally, the arousal and valence levels (see Table 5 and Table 6) show the effects of musical emotion on the covered distance. The biggest effect is due to valence ($\Delta_{dist} = 0.96$), meaning that dancers travel more on a happy music than on a sad music. They also seem to travel a bit more on a powerful music than on a neutral or quiet music ($\Delta_{dist,\pm-} = 0.25$). Moreover, we can note that the results are similar for negative and neutral arousal levels ($\Delta_{dist,n-} = -0.08$).

Table 4 - Mean total distance per music

Music 1	Music 2	Music 3	Music 4	Music 5	Music 6
25.96m	23.79m	13.42m	26.86m	17.98m	17.75m

Table 5 - Mean total distance per arousal level

Positive	Negative	Neutral
18.15m	16.21m	15.63m

Table 6 - Mean total distance per valence level

Positive	Negative	Neutral
20.89m	13.46m	15.63m

Table 7 indicates the mean total distance for improvisation and choreography. The ratio $\Delta_{dist} = 1.12$ points out an effect of this factor on the covered distance. An obvious interpretation is that the displacement in the proposed choreography was well pre-designed to occupy the available space, where the travel during an improvisation is generally spontaneous. Again, the results for the choreography are completely dependent to the only one proposed choreography in the database.

Table 7 - Mean total distance for improvisation and choreography

Improvisation	Choreography
16.87m	25.47m

In general, we can notice that the results for the covered distance are similar to those concerning kinetic energy, which is logical as a higher covered distance (for the same duration) means a higher mean velocity, and thus a higher kinetic energy.

2.4.1.3 Covered area ($\mu_{db} = 2.98m^2, \sigma_{db} = 1.97m^2$)

Mean total covered areas per gender, as shown in Table 8, indicates a possible small effect on that factor ($\Delta_{area} = 0.32$). This result is not consistent with kinetic energy and covered distance. Details on individual dancers' contributions to these results are shown in Table 9. Women are dancers 1, 3 and 5, and men are dancers 2, 4 and 6. We can see that Dancer 6 seems to be an exception, with a particularly high result compared to others, causing alone the difference between genders. Consequently, we cannot confirm a global effect of the gender on the covered area. Although, an individual effect of the dancer clearly appears ($\Delta_{area} = 1.74$). Indeed, Dancers 1 and 6 both covered an area more than twice higher than that covered by Dancer 5.

Table 8 - Mean total area per gender

Women	Men
2.57m²	3.21m²

Table 9 - Mean total area per dancer

Dancer 1	Dancer 2	Dancer 3	Dancer 4	Dancer 5	Dancer 6
3.56m²	2.34m²	2.38m²	2.14m²	1.73m²	5.15m²

Table 10 shows an effect of the music on the covered area ($\Delta_{area} = 1.87$). Music 4 has the highest result, probably because it corresponds to the choreography. Table 13 shows indeed the effect of the improvisational factor on the covered area, with a distance of $\Delta_{area} = 1.44$. Within the five improvisations, the maximum ratio is between Musics 2 and 6: $\Delta_{area,2-6} = 1.04$. Concerning emotion tags (see Table 11 and Table 12), the arousal level seems to have an opposite effect ($\Delta_{area,+-} = -0.43$) on covered area, compared to covered distance and kinetic energy. As for the valence level, the effect ($\Delta_{area} = 0.61$) is in agreement with both previous features.

Table 10 - Mean total area per music

Music 1	Music 2	Music 3	Music 4	Music 5	Music 6
2.4m²	3.6m²	2.04m²	5.25m²	2.45m²	1.56m²

Table 11 - Mean total area per arousal level

Positive	Negative	Neutral
1.98m²	2.82m²	2.45m²

Table 12 - Mean total area per valence level

Positive	Negative	Neutral
3m²	1.79m²	2.45m²

Table 13 - Mean total area for improvisation and choreography

Improvisation	Choreography
2.41m²	5.25m²

2.4.1.4 Ergonomics features

Ergonomics features are uncommon for dance style analysis, compared to Laban features and kinetics. Though, as they allow to analyse a different aspect of the movement, it is possible to explore new factors effects on the way dancers move.

For instance, Table 14 shows global distances of each factor, indicating their effects on postural load. We can see that there is a big effect of the dancer on the postural load, and a small effect of the gender ($\Delta = 0.73$). Figure 21 shows the postural load box plots for each dancer. We can observe results for the women (Dancers 1, 3 and 5) are generally higher (in terms of mean, median, quartiles 25 and 75) than men, justifying a small effect of the gender on postural load ($\Delta = 0.32$). Box plots for each gender can be seen in Figure 20. An interpretation of this effect may be that women are generally more flexible that men, allowing them to reach larger amplitudes for each joint, resulting in a higher postural load.

Table 14 - Postural load – Effect sizes for each factor

	Δ
Gender	0.32
Dancer	**0.73**
Music	0.29
Arousal	0.07
Valence	0.26
Improvisation	0.12

Table 15 and Table 16 respectively show the effect sizes for expansion and deviation features. Again, we can observe individual effects of the dancer on these features, but they are clearly independent of emotional music tags.

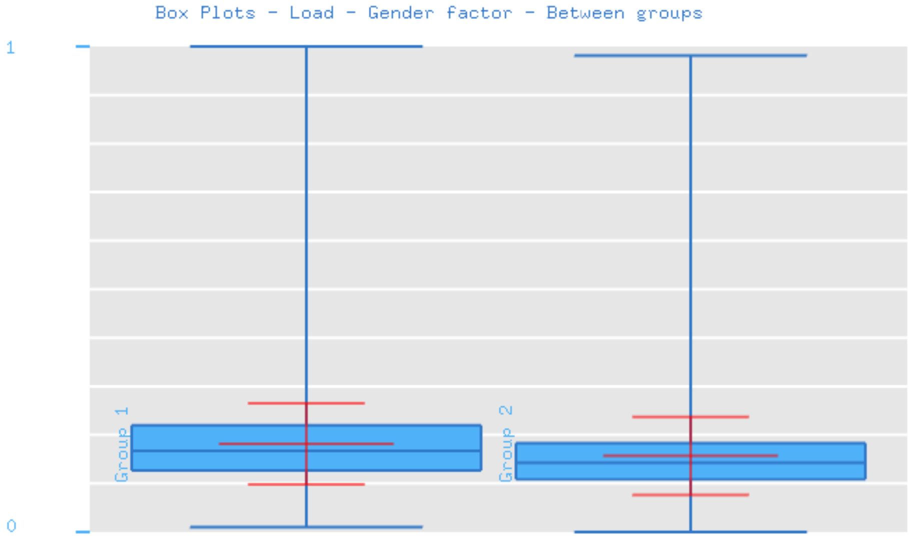


Figure 20 – Postural load box plot for each gender (women left, men right)

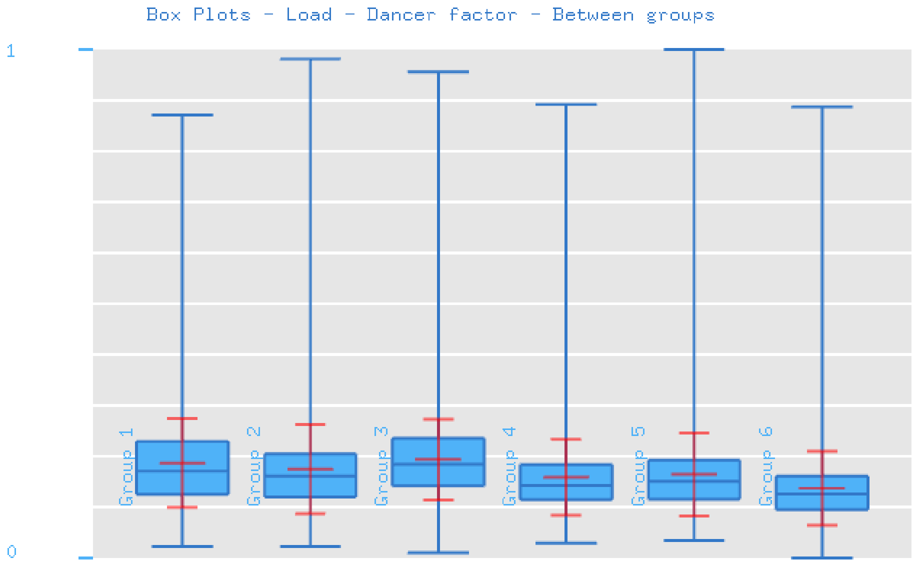


Figure 21 – Postural load box plot for each dancer

Table 15 - Expansion – Effect sizes for each factor

	Δ
Gender	0.27
Dancer	**1.62**
Music	0.36
Arousal	0.23
Valence	0.2
Improvisation	0.19

Table 16 - Deviation – Effect sizes for each factor

	Δ
Gender	0.2
Dancer	**0.9**
Music	0.36
Arousal	0.14
Valence	0.13
Improvisation	0.24

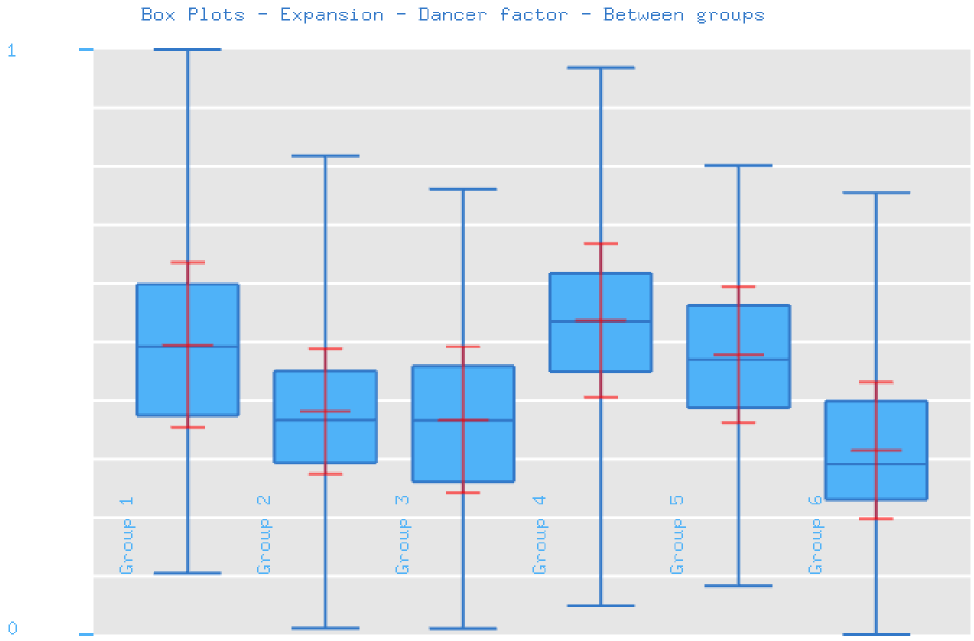


Figure 22 - Expansion box plot for each dancer

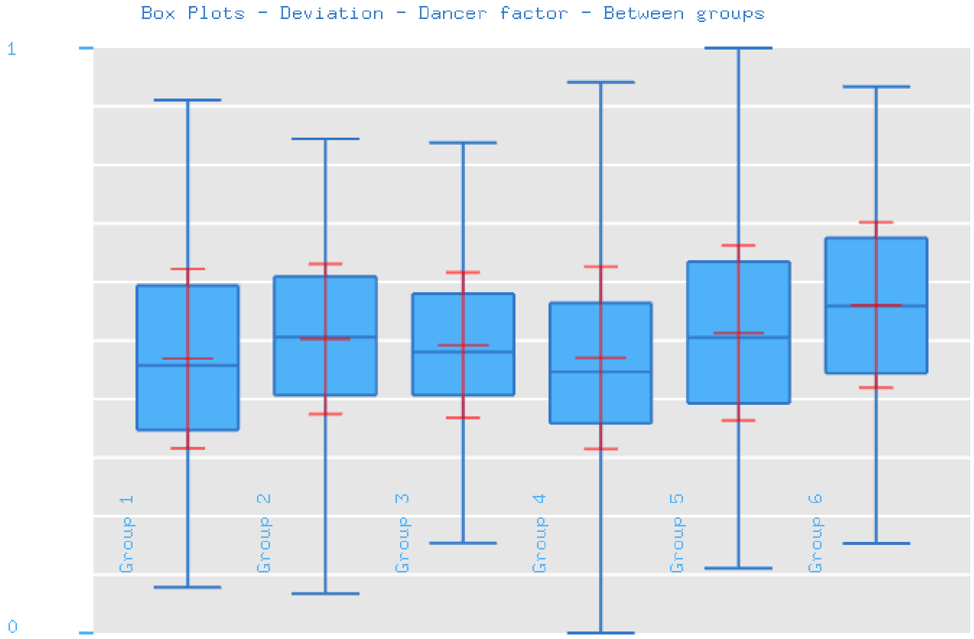


Figure 23 - Deviation box plot for each dancer

2.4.1.5 Discussion and conclusion

On overview of the ratios indicating effects between factors and features can be found in Table 17. We can see that the first three features, associated to Laban movement analysis, are affected by several factors. The three other features, associated to ergonomics, are affected only by the dancer, and a bit by the gender for the postural load. It is interesting to note that Laban features were originally developed for analysis of dance gesture, and more generally expressive gestures. It is thus logical that these features are affected by music and its perceived emotion.

Table 17 - Factors and features effects – Effect sizes

	Energy	Distance	Area	Load	Expansion	Deviation
Gender	0.13	0.19	0.32	0.32	0.27	0.2
Dancer	**0.81**	**1.3**	**1.74**	**0.73**	**1.62**	**0.9**
Music	**1.07**	**1.74**	**1.87**	0.29	0.36	0.36
Arousal	0.35	0.25	0.43	0.07	0.23	0.14
Valence	**0.66**	**0.96**	**0.61**	0.26	0.2	0.13
Improvisation	**0.64**	**1.12**	**1.44**	0.12	0.19	0.24

Concerning emotional tags, effects between valence and all features are more pronounced than those with arousal.

However, all these results serve as a first exploration of these effects, and their validity is limited to a small database. Moreover, significance tests could be done to provide an index of confidence of these results. Nonetheless, these results also serve as a proof of concept of the capabilities of the developed module for stylistic motion analysis.

Various perspectives can be considered as next steps of this project, including integration of more complex and robust statistics in the module.

2.4.2 Use case 2: Stylistic comparison (expertise level assessment) of Walloon dancers through Hidden Markov Models (HMM)

In this use case, we evaluate a Walloon dancer's performance using the Hidden Markov Models module presented in Section 2.2.7. For this purpose, a database has been constructed by capturing the motions of two dancers performing basic steps of chosen dances from the Walloon region. The first dancer is an expert of this dance and the second one is a student.

2.4.2.1 Database

Each dancer was supposed to perform three basic steps that are *Maclotte Base*, *Passepied Base* and *Passepied Fleuret*. The student did only record Maclotte Base and Passepied Base because of lack of time. The duration of each performance was 80 to 90 seconds.

In each performance, dancers repeat a series of forward steps (that are the three mentioned basic steps) and backward steps that are the same in each performance. We have considered the backward step as a forth independent step. Each step is repeated 20 to 30 times.

Data was recorded using two motion capture systems: Qualisys and Kinect V2. Qualisys captured 68 reflective markers that were placed on the body of the dancers at a framerate of 177fps. 20 anatomical landmarks trajectories were extracted from these markers trajectories using Visual3D[9]. The Kinect V2 captured 25 joints locations for each dancer at a framerate of 30fps. 20 landmarks trajectories were also extracted from this data in order to have the same format.

2.4.2.2 Modelling

The objective of the study is to estimate a score of the performance of the student, relative to the performance of the expert. We have trained HMM models for five classes (4 steps and pauses between performances). The topology selected for our models is an 11-states left-to-right HMMs with no skip transitions as shown in Figure 24.

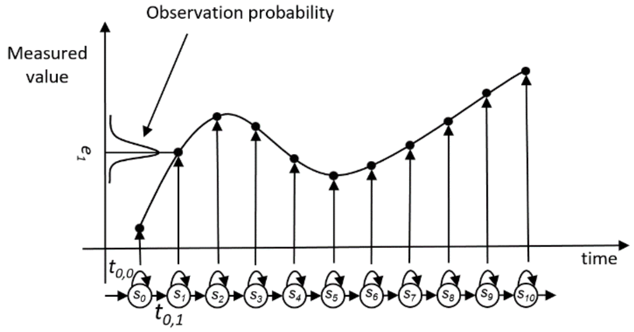


Figure 24: 11-states left-2-right Hidden Markov Model for learning procedure.

In this work we studied also the case of decoding and evaluation of gestures captured by a Kinect where HMMs are trained with Qualisys data. This is a realistic situation where, in many cases, we want to capture gestures using high precision motion capture system for precise analysis, but for daily use, the kind of systems is too expensive to be acquired, and thus, it is not suited for this kind of applications. A user prefers rather using a cheap markerless sensor, like the Kinect.

To reach this objective, we apply an adaptation procedure based on the algorithm of Maximum Likelihood Linear Regression (MLLR), in which, parameters (means and variances) of previously trained HMMs are shifted by a linear transformation in order to fit the most the adaptation data. In our case, a *mean-only* MLLR is used, being already implemented in the HTK toolkit, where only the mean is transformed (Figure 25). More details about modelling and adaptation procedures can be found in the Section 3.1.2.3 of the deliverable D3.3 "*Final Report on ICH Capture and Analysis*"

[9] http://www.c-motion.com/products/visual3d/

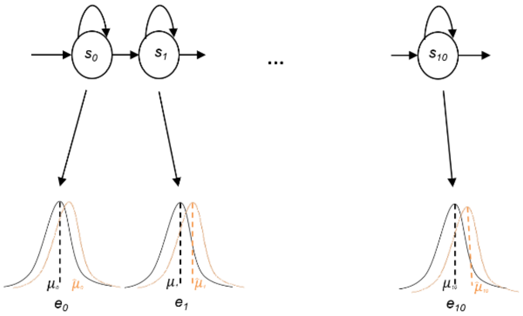


Figure 25: Illustration of the *mean-only* MLLR procdedure.

2.4.2.3 Evaluation

The decoding is performed by a standard dynamic programming algorithm called Viterbi. The system outputs a cumulated log-likelihood value which decreases when the length (number of frames) of the sequence increases. We perform time normalization of log-likelihood by dividing by the length of the sequence. The resulting value is not in a limited range and hence it cannot be interpreted by the user.

Our goal is to give the user a percentage score comparing his/her performance to a reference represented by the step models. This percentage score can be interpreted by the user as an evaluation of his/her performance whether it is good (higher than 75%), medium (between 50% and 70%) or bad (less than 50%). In order to obtain this score, we map the resulting normalized log-likelihood (L) on the following function:

$$score = \begin{cases} 0, & L < a \\ 1, & L > b \\ \dfrac{L-a}{b-a}, & otherwise \end{cases} \tag{32}$$

where a and b are determined empirically by outputting the values of the log-likelihood when decoding reference gestures. This function is illustrated in the Figure 26.

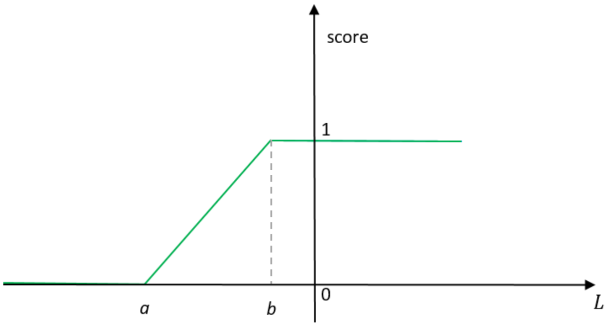


Figure 26: Score function used for mapping the normalized log-likelihood.

In order to evaluate the system, we compare the results to the expert evaluation and comments about the performance of the student, resumed in the Table 18.

Table 18: Evaluation of the student performance by the expert.

	Maclotte Base	Passepied Base	Backward step

Expert Eval.	Good	Bad	Medium

The average scores of 10 repetitions of each step given by the system are shown in Table 19. The steps are captured by the Kinect V2. We notice that the values of expert score are always high (>75%) and this is obvious because the models are trained on his data. The Maclotte Base score of the student was 87.63% (good), Passepied Base was 48.48% (bad) and Backward step was 56.40 (Medium). These results match expert evaluation.

Table 19: Comparison between expert and non-expert of the Walloon dance performances.

	Maclotte Base	Passepied Base	Backward step
Expert	96.06	99.32	87,63
Student	78.63	48.48	56.40

The main errors that students make, according to the expert, are in the fact that they don’t usually cross their feet and bend their knees enough. The Passepied step is even more complicated because the dancer needs, at the same time as crossing the feet, to make alternately a rotation to the left then to the right. For a student, this combination of gestures may be more complicated.

This can be verified by using the statistical exploration module and analyzing the effect of some of the relational features [5], related to crossing the feet and rotations, on the experience of the dancer and also on the style of the step. The data used for this analysis is the one captured by the Qualisys.

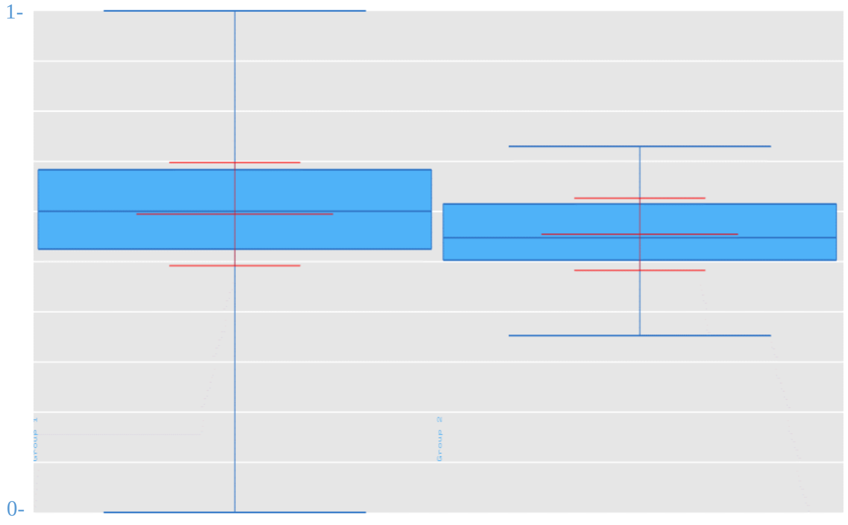
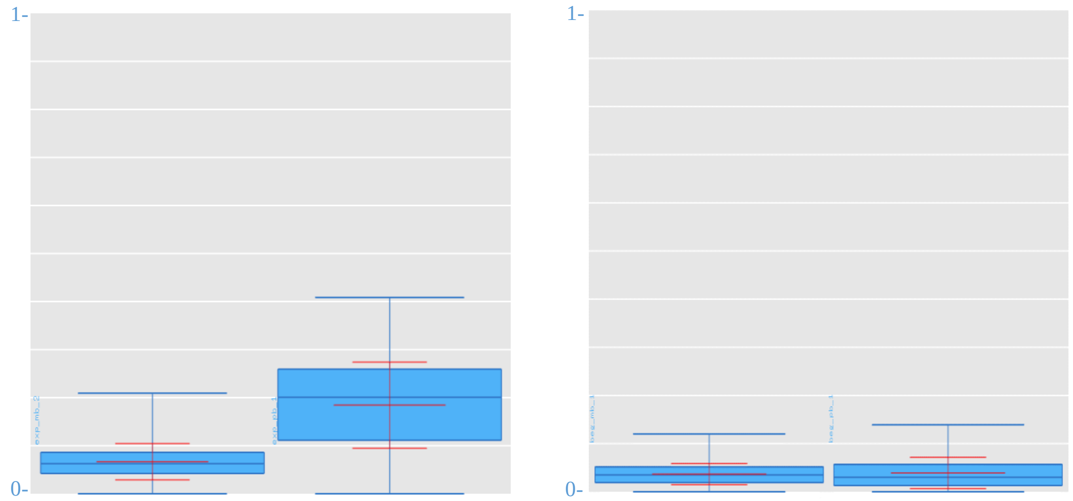


Figure 27: Effect of the feature "feet crossed" on the experience of the dancer. Left: the expert. Right: the student.

Figure 27 shows the effect of the experience of the dancer on the feature “feet crossed”. We can see clearly that the expert (left) is crossing his feet more than the student (right).

Figure 28 shows the effect of the style of the step and the feature “rotation around the spine”. We can notice first, by analyzing the expert moves (a), that the Passepied step (right) contains more rotations than the Maclotte base step (left). Second, we notice that the student is making less rotations than the expert.



a. (left: Maclotte base, right: Passepied base)

b. (left: Maclotte base, right: Passepied base)

Figure 28: Effect of feature "rotation around the spine" on the experience of the dancer and the style of the dance. a. expert’s performance, b. student’s performance

These preliminary results show that Hidden Markov Models can be used not only to model expert gestures, but also to estimate a global score of one performance relative to these models. This module has been used in the Walloon dance game to evaluate students’ performances. The given score represents an overall evaluation of the whole student’s performance. In order to have more details on the performance, for example, if the student is moving faster, or if he is not crossing his feet enough, etc. we can use the statistical factors and feature exploration module, presented in Section 2.3.

2.4.3 Use case 3: Stylistic comparison (locality variations) of Walloon dance vs Tsamiko dance based on factorial analysis

This third use case presents a preliminary analysis and comparison between two different traditional dances studied in the i-Treasures project: the dances from the Walloon region and the Tsamiko dance. The objective of the analysis is to extract differences and similarities between these dances using the factorial analysis module described in Section 2.3.

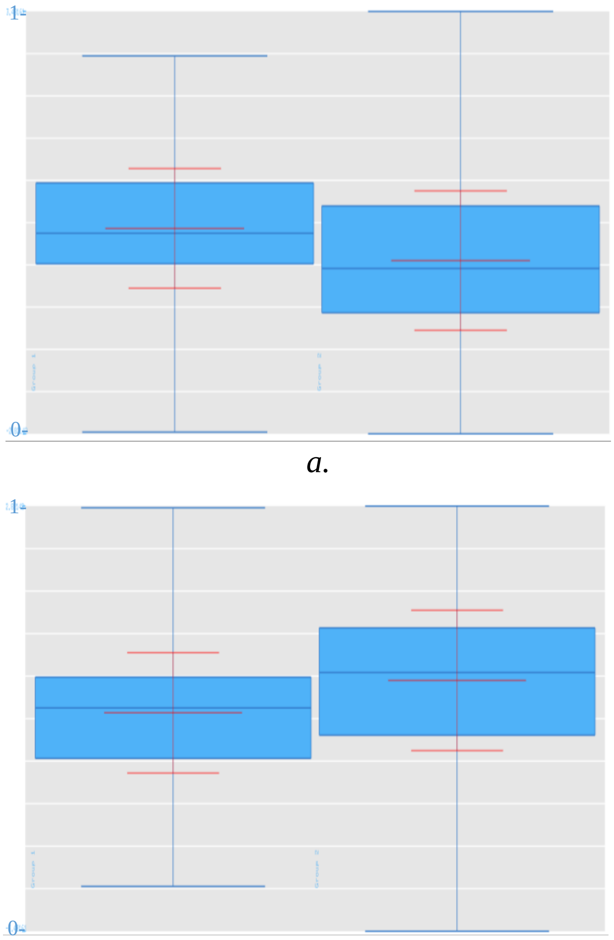
We have used the first databases recorded during the i-Treasures project. The Tsamiko dance database was recorded using the Microsoft Kinect V1 whereas the Walloon dance database was recorded using the Microsoft Kinect V2. Unfortunately, the Kinect V1 data presents a very high level of noise, which prevented us from any in-depth analysis, the results presented here are hence more of a proof-of-concept study and the work should be continued. We have selected sequences that are the less noisy in the Tsamiko database, in order to have a more meaningful analysis. The total lengths of sequences were 6565 frames for the Tsamiko database and 6355 frames for the Walloon database with a framerate of 30fps.

Among the different stylistic features available in our module, we have selected the ones that we have found to be useful in this preliminary analysis, which are:

- Relational features: feet crossed, feet apart sideways
- Effort features: weight effort
- Ergonomics: expansion, deviation and balance

2.4.3.1 Relational features

We have selected among different relational features, those that are the most significant and the less noisy related to feet movements, since the important movements of both Tsamiko and Walloon dances are concentrated on the feet. These features measure distances related to crossing the feet (a) and moving feet aside (b).



a.

b.

Figure 29: Effect of the dance on two relational features. a. feet crossed. b. feet apart sideways. (in each columun: left: Walloon performances, right: Tsamiko performances).

We notice that, in general, dancers from both dances cross their feet a lot when performing (Figure 29.a); but the Walloon dancer crosses the feet more than the Tsamiko dancer. Because the Tsamiko dance is performed laterally, we see in Figure 29.b that the feature “feet apart sideways” is higher for this dance.

2.4.3.2 Effort features: Weight effort

A motion characterized by a high weight effort is strong (powerful), whereas a motion with a low weight effort is light (gentle). The weight effort during performances in both dances is not high, as shown in Figure 30 and is almost equal in both dances. This shows that the movements in both dances are not very energetic.

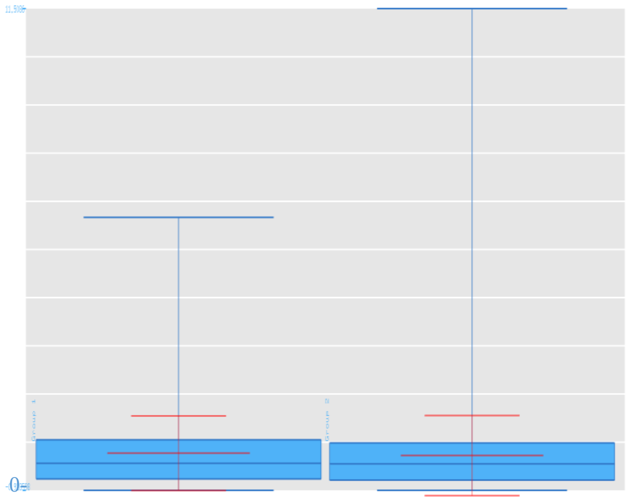
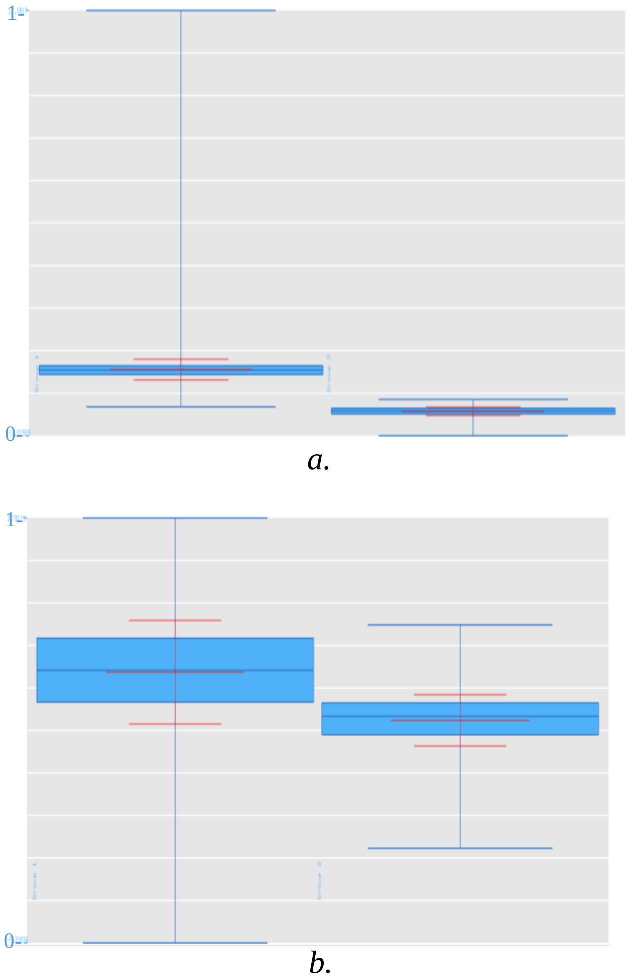


Figure 30: Effect of the dance on the weight effort feature (left box plot: Walloon performances, right box plot: Tsamiko performances).

2.4.3.3 Ergonomic features: Expansion and deviation

Analysing Figure 31, we notice that for both dances, the radius of the covering sphere (a) is small, which means that the muscles are crouched and dancers do not adopt an expanded pose. We also notice that this feature is higher in the Walloon dance case. The deviation (b), indicates the deviation of the body from the spherical shape (in other words, the sphereness of the body). We notice a high deviation for both dances. This can be explained by a low balance of posture during the dance. We can also say that the Walloon dance is less balanced than the Tsamiko dance.



a.

b.

Figure 31: Effect of the dance on the Expansion (a) and the deviation (b) features. (left box plot: Walloon performances. right box plot: Tsamiko performances)

2.4.3.4 Conclusion

This preliminary analysis of the Walloon and Tsamiko dances showed few characteristics of each dance and what are the factors that can distinguish between them. Unfortunately, the quality of data did not allow for a complete and meaningful analysis of these dances. In a future work, we can have an in-depth analysis by having a data of a better quality, especially for the Tsamiko dance, having already a high precision data available for the Walloon dance.

3 Pottery gesture style comparison using the MYO sensor

This section presents the methodology proposed for the style comparison of pottery gestures across several types of vases. Our work was inspired from previous investigations that are overviewed in the following 5 paragraphs.

Today, the hardware for gesture recognition is based on infrared (IR) sensors, Electromyograms (EMGs), and Inertial Motion Sensors (IMS) or combinations of them. IR sensors are based on projecting an infrared pattern in the room and capturing the pattern with an IR camera. A common pattern is a grid of dots. If the dots captured on a surface are dense, the surface is near the camera, and vice versa. Such devices are LEAP motion sensor [34] and MS Kinect [35]. The methods based on EMGs classify movement according to the electricity on the surface of the skin, which reflects the activity of the muscles. Myo sensor is such a device [36]. Methods based on IMS are exploiting several wearable sensors worn by the user in several places in his body. The device we have employed is Myo that combines EMG and IMS sensors in the forearm, and therefore, it can be used unobtrusively for pottery sessions.

Most of the research based on EMG is performed with a sensor made by a certain lab that it is not available for the community and therefore experiments cannot be reproduced accurately. Myo sensor is a relatively new device as it was released to the public at 2013 and offers a benchmark device at low cost ($200). It is a wireless device connected to desktops or mobiles offering a good user experience. Myo is ideal for art performances such as pottery, since the other options, IR and IMS based sensors, cannot be used due to physical limitations. In this work, we focus on gesture recognition at the level of fingers from EMG(s). In particular, we wish to extract information about the finger-gestures, which is a particularly valuable piece of information for the learner in order to improve his performance in pottery arts. In the literature, the gesture recognition methods that rely on EMG typically refer to hand-gestures. Although we do not treat the same task, we have reviewed some of these methods so as to highlight the most interesting components.

The process of gesture recognition is split into two parts, namely feature extraction and classification. A feature proposed in [37] is the entropy of splines fitted on the EMG signal and the entropy of splines fit on the residuals of every fit. This process leads to a dimensionality reduction of the original signal where entropy can be estimated to characterize the low frequencies of the signal. The classifier employed was based on linear discriminant analysis (LDA) using the Mahalanobis distance in order to classify signs for the deaf people with an industrial research EMG sensor. In [38], the Hidden Markov Models (HMMs) and decision trees were used to separate signs language from EMG and acceleration values by using as features the duration of the sign, the autoregressive coefficients of fourth order applied to short-term frames of 250ms, and the mean values of the signals in the short-term frames. For this purpose, a custom EMG sensor was used. In [39], the mean value, the variance, the fourth order coefficient, and the sample entropy on short-term frames of 250ms were used to classify gestures to seven primitive classes, e.g. grasp, point, tip, rest and so on by using two industrial EMG sensors. LDA was used as a classifier to achieve more than 97% correct classification rate. The resulted decision was graphically depicted by using a VRML model of the palm, where each finger was represented by the concatenation of cylinders. Motivated from this work, we have foreseen the necessity of visualizing the arm in 3D space, so that experiments are performed in emulated conditions. However, compared to [39], instead of using VRML and Matlab we have preferred WebGL and javascript which are more accessible to the community and they can lead to a ready to use solution.

In [40], a custom armband sensor is presented that comprises of 8 transmitters and 8 receivers of electrical signals at 40Hz, which is able to measure electrical impedance of a forearm cross-section area. Thus, a tomography of the forearm is achieved that is able to predict the inner muscles activation level. The combination of the 8 sensors to 2 pairs achieves a feature vector of 28 coefficients that are updated every 100ms. The Support Vector Machine (SVM) based classifier achieved about 95% correct classification rate for 11 gestures. Following a similar approach, our work was also based on the notion of cross-sections of the forearm, and was further augmented by projecting the virtual models of the muscles.

Concluding hand gestures recognition review, it is important to mention that most of the aforementioned research is performed with a toolkit for experiments, i.e. Matlab, SPSS, WEKA, LabView, etc. In our work for pottery style recognition, we have relied on HTML-5, Javascript and WebGL so as to develop a cross-platform and ready to use eco-system for myoelectric analysis that can be easily used for every application. An additional advantage of the employed architecture is that the computational burden of handling 3D graphics lies on the client side, where WebGL is taking advantage of the available GPU. Similarly, most of the Myo signal recording processes are performed by webworkers that run on a parallel thread to avoid overloading the GUI thread [41].

The outline of the rest of this section is as follows. Section 3.1 presents the data recordings and the procedure that has been employed to annotate them. In Section 3.2, the proposed 3D visualization tool and the type of features that have been introduced for our analysis are described. Details of the implementation are provided in Section 3.3. The experimental results are shown in Section 3.4, where the proposed features indicate differences across the three types of vases for the same gesture.

3.1 Database and Annotation tool

For the purposes of our experimental study, we have assembled a dataset of Myo recordings during 9 pottery art performances from a master performer, namely 3 bowls, 3 cylindrical vases, and 3 spherical vases where each one lasts about 100 secs. Each vase type is shown in Figure 32.

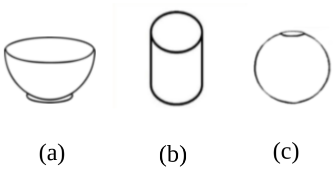
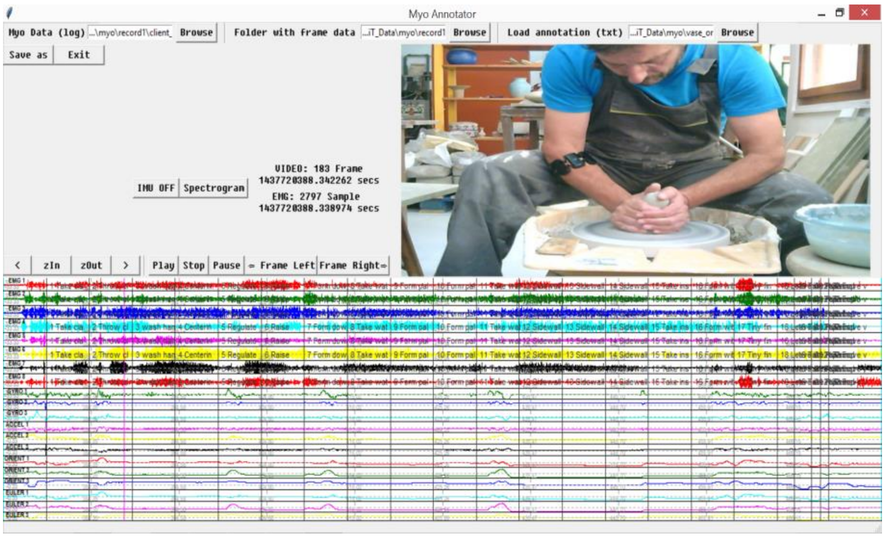


Figure 32: Data consists of Myo recordings of the right hand during three vase classes, namely (a) bowl, (b) cylindrical, and (c) spherical vases.

The video of each performance was captured to help annotating the signals. The video frames were recorded with a custom C++ program synchronized to start with the Myo recorder. Each video frame is named according to the current unix timestamp, and each Myo sample is also associate with its unix timestamp. The sampling rate for EMG is 200Hz and for IMU is 50Hz. Subsequently, the Myo recordings were annotated into basic pottery gestures using a custom made annotation tool with a graphic user interface. The tool was written in Python 3 using tkinter as a GUI constructor. The tool as shown in Figure 33, visualizes, in synchronization with the video frames, the 8 EMG signals from each Myo pod and the 9 IMU signals, namely acceleration, gyroscope, orientation for each of the x, y, z axis. The GUI support several actions such as the playline in the timeline can be dragged with the mouse and the video frame is updated on-the-fly. Each of the three types of vases requires about 25 gestures and the total number of gesture classes is 33. The gestures for making a bowl, a cylindrical vase, and a spherical vase as annotated with our tool, are shown in Table 20: Gestures required to make a bowl, a cylindrical vase, and a spherical vase with timestamps for three annotated sessions.


Myo Data (log) ...\myo\record1\client_

Folder with frame data ...iT_Data\myo\record1

Load annotation (txt) ...iT_Data\myo\vase_or

VIDEO: 183 Frame
1437720388.342262 secs

EMG: 2797 Sample
1437720388.338974 secs

Figure 33: Annotation tool for Myo signals synced with video signal.

Table 20: Gestures required to make a bowl, a cylindrical vase, and a spherical vase with timestamps for three annotated sessions.

No.	Bowl	Cylindrical	Spherical
1	2.2, Take clay	0.9, Take clay	0.5, Take clay
2	4.4, Throw clay	2.7, Throw clay	3.3, Throw clay
3	5.2, Wash hands	3.5, Take water	4.3, Take water
4	8.1, Centering clay	4.8, Centering clay	6.0, Centering clay
5	10.5, Regulate speed	6.5, Raise clay	8.8, Raise clay
6	12.3, Raise clay	9.8, Form down cyclic	10.9, Take water
7	15.6, Form down cyclic	20.5, Take water	12.5, Raise clay
8	26.3,Take water	22.0, Form palms facing	16.6, Form down cyclic
9	27.8, Form palms facing	25.7, Form palms open	28.0, Take water
10	34.8, Form palms open	27.5, Take water	30.5, Form palms facing sphere
11	36.7, Take water	29.1, Sidewalls cyl1	33.6, Form palms open sphere
12	39.3, Sidewalls 1	32.2, Take water	35.4, Take water
13	50.7, Sidewalls 2	34.1, Sidewalls cyl2	37.6, Sidewalls sph1
14	54.2, Pointer fix	51.3, Pointer finger	41.7, Take water
15	56.3, Take instrument	55.2, Sidewalls cyl2	44.5, Sidewalls sph2
16	59.3, Form with instrument	57.3, Pointer finger	53.1, Take water
17	74.4, Tiny finger	58.0, Take instrument	55.8, Sidewalls sph3
18	82.5, Leave instrument	60.4, Form with instrument	60.6, Sidewalls sph4

19	84.5, Take rope	77.0, Tiny finger	64.7, Take instrument
20	86.2, Pass rope under	78.5, Pointer finger	67.5, Form with instrument
21	89.7, Remove vase	81.4, Form with instrument	78.7, Leave instrument
22	96.7, End	89.0, Leave instrument	85.1, Sidewalls sph5
23	-	90.9, Take rope	93.6, Take instrument
24	-	92.7, Pass rope under	97.7, Form with instrument
25	-	96.2, Remove vase	104.9, Tiny finger 1
26	-	103.1, End	110.1, Tiny finger 2
27	-	-	113.5, Form with instrument
28	-	-	119.3, Leave instrument
29	-	-	121.9, Take rope
30	-	-	123.8, Pass rope under
31	-	-	126.2, Remove vase
32	-	-	130.1, End

3.2 Methodology

In order to analyze the correspondence between the EMG-based muscle activation and the pottery gestures, we have visualized the forearm in 3D space and interactively observed if the proposed theory is reflecting the actual muscle-driven process.

3.2.1 Virtual forearm model resources and study

The muscles located at the level of Myo pods were investigated by the use of an anatomy atlas that also offers for free the 3D models of the bones and the muscles [42]. The directly involved anatomical items are the 2 forearm bones, namely radius and ulna as well as 15 muscles which are located at the upper part of the forearm as shown in Table 21: Muscles contributing to Myo signals. Abbreviation: “cs. vol.” is the cross-sectional volume found by the intersection of a 0.5cm thick plane with the muscle at Myo level.. Some muscles have two heads that are connected to different bones, e.g. humerus and ulnar heads. For convenience the names are shortened from now on to three letters, e.g., extensor digiti minimi becomes “ext. dig. min”. Indirectly involved anatomical items are the rest bones of the arm such as the humerus bone, the eight wrist bones, and the finger phalanxes that are also visualized in the 3D model. The anatomic functionality of the muscles is found in [43]. The result is the forearm scene that is depicted in Figure 34. It is seen that Myo can capture the flexors and the extensors of the distal, middle, and proximal phalanxes, the carpal ulnar and radial deviation, the pronation and suspination, but not the thumb movements.

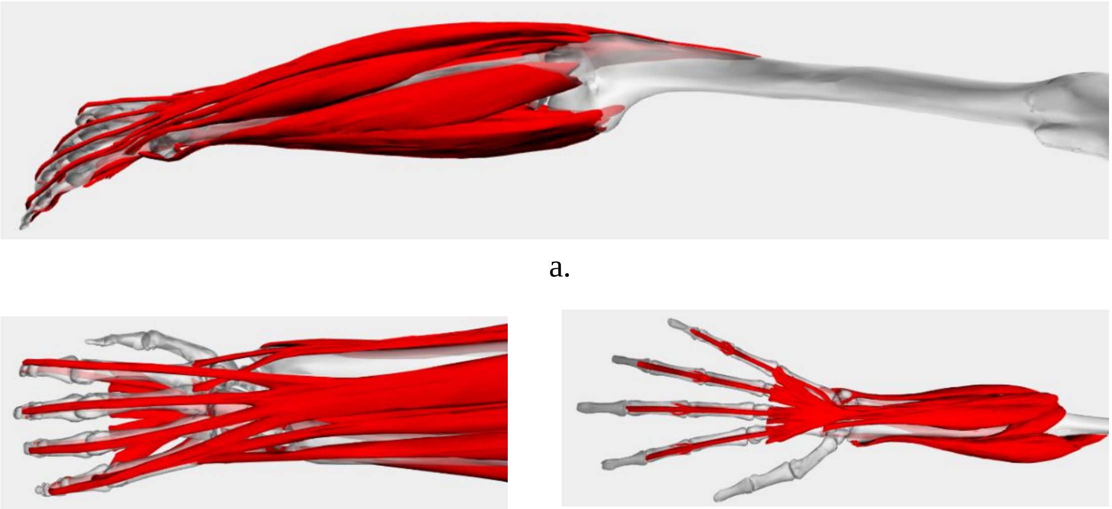
Table 21: Muscles contributing to Myo signals. Abbreviation: “cs. vol.” is the cross-sectional volume found by the intersection of a 0.5cm thick plane with the muscle at Myo level.

#	Name	Main functionality	cs vol. cm^3
1	Brachioradialis	Flex elbow joint	2.26
2	Extensor carpi radialis brevis	Extends and radial deviates hand at wrist	0.89
3	Extensor carpi radialis longus	>>	0.84
4	Extensor carpi ulnaris	Extends and ulnar deviates hand at wrist	0.53
5	Extensor digiti minimi	Extends the little finger	0.17
6	Extensor digitorum	Extends wrist and 4 fingers	0.91
7	Flexor digitorum profundus	Flexes wrist and 4 fingers (connected at tips)	1.83
8	Flexor digitorum superficialis	Flexes wrist and 4 fingers (connected at 2nd phalanx)	1.77
9	Flexor carpi radialis	Flexes and radial deviates hand at wrist	0.80
10	Flexor carpi ulnaris (Ulnar head)	Flexes and ulnar deviates hand at wrist	0.88

11	Flexor carpi ulnaris (Humeral head)	>>	1.01
12	Palmaris longus	Flexes hand at wrist and tenses palmar aponeurosis (like grasping a ball)	0.53
13	Pronator teres (Ulnar head)	Pronates (turn door key to your body) and flexes forearm at elbow	0.40
14	Pronator teres (Humeral head)	>>	0.40
15	Supinator	Rotates radius to turn palm anteriorly (turn door key away from your body)	0.44

The cross-sectional volume is the volume of the mesh formed by intersecting a 0.5 cm thick plane with each muscle at Myo cross-sectional level. The result is the forearm scene that is depicted in Figure 35. More specifically, the muscles at Myo level as indicated by the grey plane intersecting the forearm are the electrical sources contributing to the EMG signals. The intersection of the plane with each muscle produces a muscle tomography which we will call as mesh. The mesh is found with a known algorithm in 3D graphics called as binary space partitioning, and the implementation followed is as in [44]. The center and the volume of the mesh are also calculated. Volume is calculated with summing the signed volume of tetrahedrons formed by the triangles (faces) topped at the origin [45]. Let us name as m_i the mesh of the i-th muscle. The center of the mesh, $\underline{c}_{m_i} = [\, x_{m_i}, y_{m_i}, z_{m_i}]$, is calculated by dividing the centers of all of its faces, $\underline{c}_f$, multiplied by their face area, a_f, and afterwards divide each vector with the total area of its surfaces, .e.g.

$$\underline{\mathrm{c}}_{\mathrm{m_i}} = \frac{\sum_{\forall \mathrm{f} \in \mathrm{m_i}} \mathrm{a_f} \underline{\mathrm{c}}_{\mathrm{f}}}{\sum_{\forall \mathrm{f} \in \mathrm{m_i}} \mathrm{a_f}} \tag{33}$$


a.

b.

c.

Figure 34: Muscles contributing to Myo signals visualized with the developed tool: (a): overview; (b): posterior view; (c): anterior view.

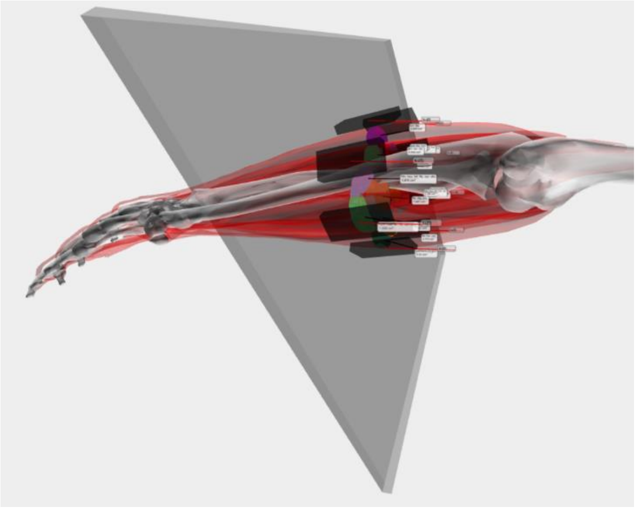


Figure 35: Intersecting the muscles of the forearm 3D object with a 0.5cm thick plane.

3.2.2 Muscle activation level

Let us define as muscle activation level the amount of electricity that the muscle produces which is reflected in the Myo signals. The aim is to predict the muscle activation level by exploiting the Myo signals and the forearm anatomy. The algorithms are as follows.

The negative values of the EMG signals were clipped as muscle activation is denoted by the positive spike values, i.e. the discharges of the muscles. Negatives are the charges of the muscles that occur as a physical re-action to the discharge. Second, an average mean filtering of the past 25 values is applied to remove high frequency artifacts. Third, weights for each of the 8 Myo sensors per each of the 15 Muscles are calculated that reflect amount of electricity that each Myo sensor (pod) measures per muscle. The weights depend on the distance of the muscle from the sensor as well as the muscle volume. It is assumed that the weight is proportional to the volume of the muscle, i.e. the bigger the muscle is, the more electricity it produces. However, the weight is inversely proportional to the distance that is the further the muscle is from the sensor, the lowest is its contribution to the sensor's signal. Let e_{m_i} be the electrical activity signal of the muscle m_i, where i =1, 2… 15, and t is the time, then:

$$e_{m_i} = V_{m_i} \sum_{j=1}^{8} \frac{s_j(t)}{d_{ij}} \tag{34}$$

where V_{m_i} is the volume of the muscle m_i, $s_j(t)$ is the signal of the j-th Myo pod and d_{ij} is the distance of each pod to each muscle, i.e.

$$d_{ij} = \sqrt{(x_{s_j} - x_{m_i})^2 + (y_{s_j} - y_{m_i})^2 + (z_{s_j} - z_{m_i})^2} \tag{35}$$

The numbering of Myo pods starts from -45 degrees as shown in Figure 36 and continues counter-clockwise until 270 degrees. The weight coefficients 1/d_{ij} normalized to their sum towards j are displayed in Table 22. Each row corresponds to a muscle and each column corresponds to a Myo pod. Thus, the weight is the relation of the Myo signal to each muscle signal.

Table 22: The contribution of each Myo signal to each muscle signal. Values in %.

	Pod 1	Pod 2	Pod 3	Pod 4	Pod 5	Pod 6	Pod 7	Pod 8
Brachioradialis	2	2	2	3	69	11	4	8
Hum. hea. lef. pro. ter.	7	3	3	5	37	12	8	24
Uln. hea. lef. pro. ter.	8	4	6	8	28	13	9	25
Ext. car. rad. lon.	3	3	3	8	13	31	35	4
Supinator	6	6	9	17	15	15	21	11
Ext. car. rad. bre.	4	4	5	2	11	15	35	7
Ext. dig.	3	3	6	44	4	7	29	3
Ext. dig. min.	6	6	9	37	7	9	2	6
Ext. car. uln.	5	7	14	45	5	5	13	5
Uln. hea. lef. fle. car. uln.	1	18	42	15	3	3	5	5
Hum. hea. lef. fle. car. uln.	11	72	9	2	1	1	1	2
Fle. dig. pro.	21	15	15	11	9	6	6	17
Palmaris longus	88	4	2	1	1	0	0	4
Fle. dig. sup.	45	17	9	5	5	3	3	14
Fle. car. rad.	29	7	4	3	6	3	3	46

Features extraction

The proposed features are based on summing the muscle activations in certain muscle groups, which in the general case is found by:

$$f = \frac{1}{N}\sum_{t=1}^{N}\sum_{M} e_{m_i} \tag{36}$$

where M denotes the muscle group and N is the total duration of the gesture. Depending on the muscle group M we define the following features.

The f_{total} found when M is the set of all muscles of Table 21 denotes the *total muscle pressure* of the gesture. The f_{flex} is the flexion tension of the gesture which is calculated by the flex muscles M = {"brachioradialis", "palmaris longus", "fle. car. rad.", "Uln. hea. lef. fle. car. uln.", "Hum. hea. lef. fle. car. uln.", "fle. dig. sup.", "fle. dig. pro."}, denoting the force put towards the anterior of the forearm. In the same manner extension tension f_{ext} of the gesture is found by the 5 extensors, i.e. M = {"ext. dig.", "ext. car. uln.", "ext. car. rad. lon.", "ext. dig. min.", "ext. car. rad. bre."}, denoting the force put towards the posterior of the forearm. The suspination feature f_{susp} is defined by using only the suspinator muscle, and the pronation feature f_{pron} is defined by using the {"Uln. hea. lef. pro. ter.", "Hum. hea. lef. pro. ter."} set of pronation muscles.

Finally, we denote as stiffness f_{stif} the steadiness of the palm towards forearm that is caused by balanced force of the extensors/flexors and suspinators/pronators, i.e. the opposite muscle groups that put opposed forces to make the hand stiff to be defined as

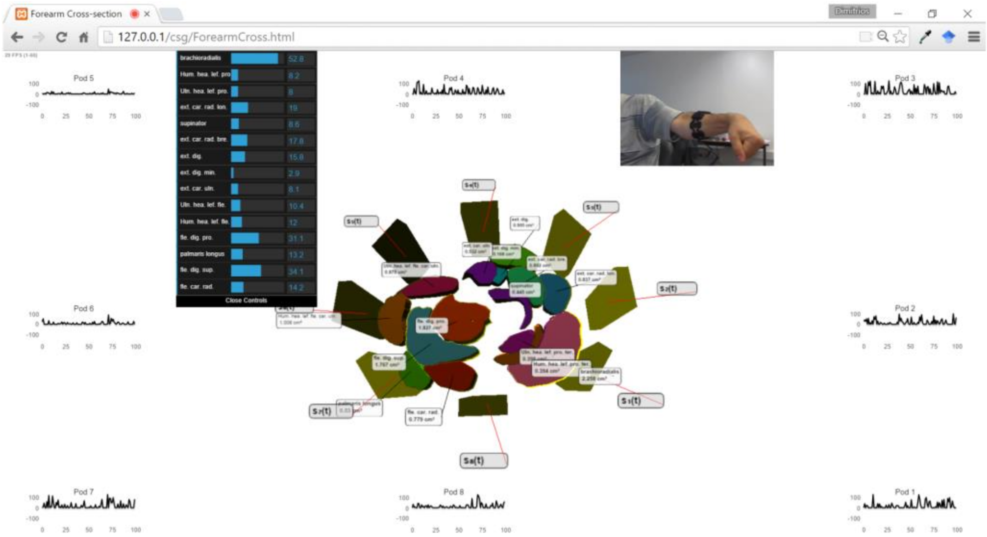
$$f_{stif} = f_{flex}f_{ext} + f_{susp}f_{pron} \tag{37}$$

3.2.3 Implementation context for myoelectric analysis

A 3D engine was necessary to render the muscles for visualizing the whole forearm. The reason for favouring a solution based on web-browsers was due to the following advantages: a) they are accessible from everywhere without any posing any software requirements; b) they offer a rich development environment as they include a console that can be used to interactively experiment with the loaded web page; and c) they support the WebGL of HTML5 that renders 3D graphics with the client's Graphic Processing Unit (GPU). However, in order to cope with the problem that WebGL has low level commands that are not easy to use for experiments, we have employed the Three.js framework [46] which is a javascript framework for WebGL 3D graphics that has high level commands for rendering scenes.

On top of the aforementioned infrastructure, several javascript libraries were used in order to transform the web page into a full Myo observation console. In order to port Myo signals to a web page the Myo.js library was used [41]. The library ports values from the Myo desktop drivers to the web page through websocket technology. The Myo signals were plotted into the web page through a jquery library named as flot [47]. The signal values were also shown in an interactive user interface with the help of dat.gui library [48].

In this way, using the web browser, the muscles' electrical activity can be observed in real-time as shown in Figure 36. A video demonstrating the functionality of the tool can be found in [49]. In this video, the muscles electrical activity and the Myo pods signal values are represented with a yellow overlay on the respective objects.



brachioradialis 52.8
Hum. hea. lef. pro 8.2
Uln. hea. lef. pro. 8
ext. car. rad. lon. 19
supinator 8.6
ext. car. rad. bre. 17.8
ext. dig. 15.8
ext. dig. min. 2.9
ext. car. uln. 8.1
Uln. hea. lef. fle. 10.4
Hum. hea. lef. fle. 12
fle. dig. pro. 31.1
palmaris longus 13.2
fle. dig. sup. 34.1
fle. car. rad. 14.2

Figure 36: Live view of Myo signals and muscle electrical activity in the cross-section model of the muscles.

3.3 Experimental results

Pottery gestures such as "Raise clay", "Form down cyclic", "Form with little finger", "Form with palms open", "Form with palms closed" are in common among the 3 types of vases, and therefore, constitute an interesting case to be analysed using the proposed features. Other gestures in common such as "Take water", "Take instrument", "Leave instrument", "Take rope", "Pass rope under" do not exhibit style differences as they are very simple to perform. The features used in our experiments are the pressure, stiffness, flexors pressure (flexion), and extensors pressure (extension). Statistics such as the mean, median, the variance, and the 25/75 percentiles were estimated on these features across all instances per class (whereas by class we refer to the type vase being constructed, i.e. bowl, cylinder, sphere). The results are plotted in the so called box plots [Beyer81].

In Figure 37, “Raise clay”, i.e. the gesture to pull clay upwards while it is rotating on the table, is compared across the 3 vase types with respect to the total muscle pressure feature. In the bowl vases the user seems to put high pressure across all of his attempts. In cylinder vases the pressure is much less, whereas in spherical vases the pressure varies across the attempts with a tendency to the high values. Based on these results, we can infer that cylindrical vases do not need much pressure when raising up the clay.

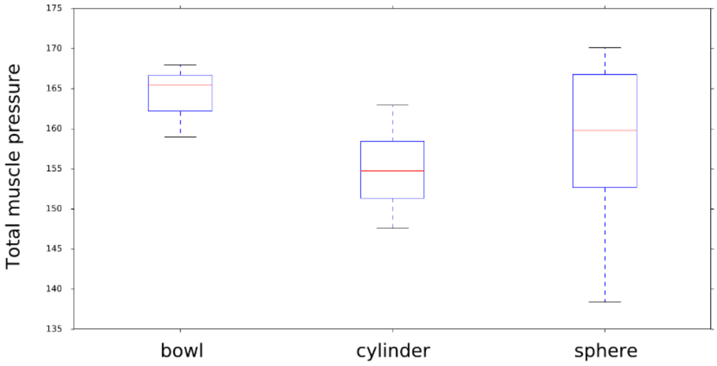


Figure 37: Total muscle pressure during “Raise clay” gestures across three types of vases.

In Figure 38, the stiffness of the forearm when performing the “Tiny finger” gesture, i.e. the formulation of the upper rim of the pot with the little finger, is depicted. It is seen that when making a spherical vase the stiffness of the little finger should be high, which might be explained by the sensitivity of the narrow rim of the vase which requires more precise formulation with a stiff tool.

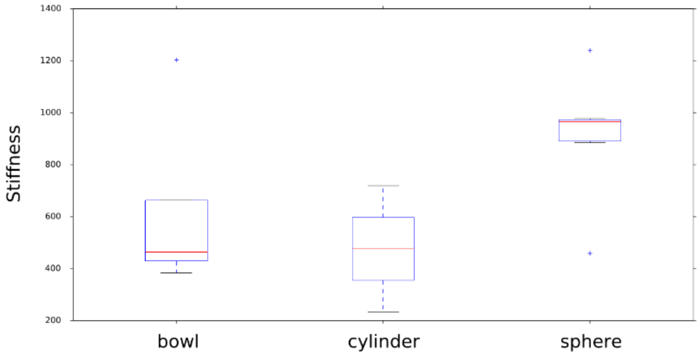


Figure 38: Stiffness during “Tiny finger” gestures across three types of vases.

In Figure 39, the flexion of the forearm when performing the “Centering clay” gesture, i.e. the transformation of the raw clay into a rounded mass on the spinning table, is depicted. The flexion of the forearm is an indication of press down and centering force put on the clay which is higher in the cylindrical and the spherical vases. This might be due to the higher height of cylinder and sphere that require a more condensed clay that can stay still when it is pulled up.

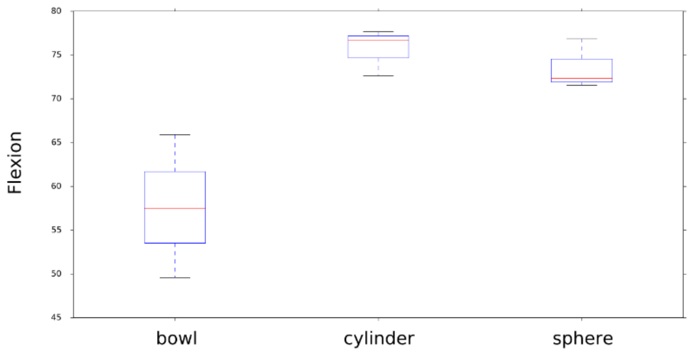


Figure 39: Flexion feature during "Centering clay" gesture across three types of vases.

In Table 23, we present the mean value of 4 features for 4 gestures across all attempts per each of the 3 vase types. It is seen that significant differences exist across vase types within each gesture. For example, for "Raise clay" the stiffness and the total muscle pressure are higher in bowl than in the other vases. Over-viewing the values across gestures, it is validated that "Tiny finger" gesture has smaller muscle activation level than the "Raise clay", "Form down cyclic clay", and "Centering clay" as it requires a smaller group of muscles to activate the little finger.

Table 23: Mean value of features per vase class for gestures "Raise clay", "Form down cyclic clay", and "Tiny finger".

	Total Muscle Pressure	**Stiffness**	**Flexion**	**Extension**
Vase Type	"Raise clay" gesture			
Bowl	164	5675	67	65
Cylinder	154	5022	67	58
Sphere	157	5201	70	58
	"Form down cyclic clay" gesture			
Bowl	165	5756	67	64
Cylinder	130	4054	55	50
Sphere	154	5031	65	58
	"Tiny finger" gesture			
Bowl	52	628	21	22
Cylinder	45	475	18	18
Sphere	65	913	30	24
	"Centering clay" gesture			
Bowl	145	4505	57	58
Cylinder	181	6802	75	68
Sphere	176	6448	73	67

This preliminary experimental results verify the validity of the proposed methodology in identifying differences across the pottery gestures that may phenomenally look identical, constituting a powerful framework for capturing and communicating the intangible aspects of a pottery performance that makes it unique.

3.4 Discussion and future work

The developed framework, is not just limited to forearm but it can be extended to any part of the body and it could be potentially connected to any EMG device that has the proper middle-ware (websocket interface). In the future, the accelerometer and the gyroscope of Myo will be connected to our framework, so that the 3D forearm can move in the virtual environment and offer a more insightful visualization. In addition, the muscles will be animated so that the electrical activity will drive the visualization of the finger movements and the forearm. Furthermore, we look forward in incorporating a 3D physics library into our visualization tool so that interaction with the 3D environment is possible.

The resulted work can be exploited through an open source channel and a commercial channel. As regards open source channel, the database, the MyoAnnotation tool, and our middleware for muscle activity emulation will be released in Github. The commercial approach is to develop an application for learning pottery that can be sold through Myo market portal [50] or a handcraft/manufacturing company. Through our website for pottery it will be possible to collect more data that can be used for training the algorithms for “high level information” extraction or for gestures recognition as well. Although pottery is the main target, the business model should be generic to cover other handcrafts and functionalities where the gestures to speech ability is a fundamental one. This can be achieved with the interconnection with open APIs such as text to speech API.

4 Conclusion

In this report on ICH Indexing by Stylistic Factors and Locality Variations, we addressed the problem of complexity arising in the analysis of full body and hand gestures. For full body gestures analysis, we presented a new tool for statistical exploration of relations between motion features and factors. We first introduced the environment created and used for its development: MotionMachine, a programming framework allowing the development of interactive applications for motion processing and analysis. Several stylistic features have been prototyped using MotionMachine including rhythm and periodicity, ergonomics, Laban and machine learning representations. These features have been used by the factor statistical exploration module in order to investigate how the different factors (gender, arousal, valence…) can influence the way a performer moves. Three use cases have been studied. The first was conducted on database of improvised contemporary dance, the second had the aim of using Hidden Markov Models for performance evaluation of Walloon dances, and the third one evaluated locality variations between Tsamiko and Walloon traditional dances. The analysis of the Tsamiko and Walloon dance are preliminary results, and should be further investigated in the future, using better quality data.

As regards pottery style recognition, we have proposed a methodology and a set of features for identifying fine-grained differences across pottery gestures that may phenomenally look identical. Our work was largely driven by the implementation of a framework that allowed us to visualize in real-time the signals generated from a Myo sensor along with the muscle activation level in 3D space. Using this framework, we were able to perform a number of emulation experiments, for estimating the cross-sections of the arm and identifying the most suitable set of features for our study.

5 Appendix: Factorial analysis of improvised contemporary dance – Kinetic energy graphs

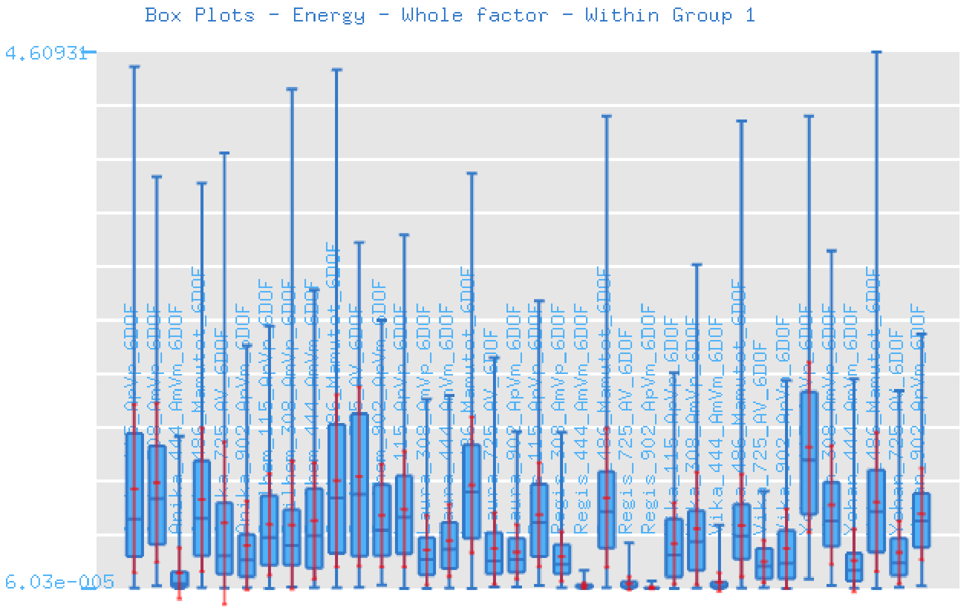


Figure 40 - Kinetic energy boxplot for each sample

Note: the samples are ordered by dancer, and then by music.

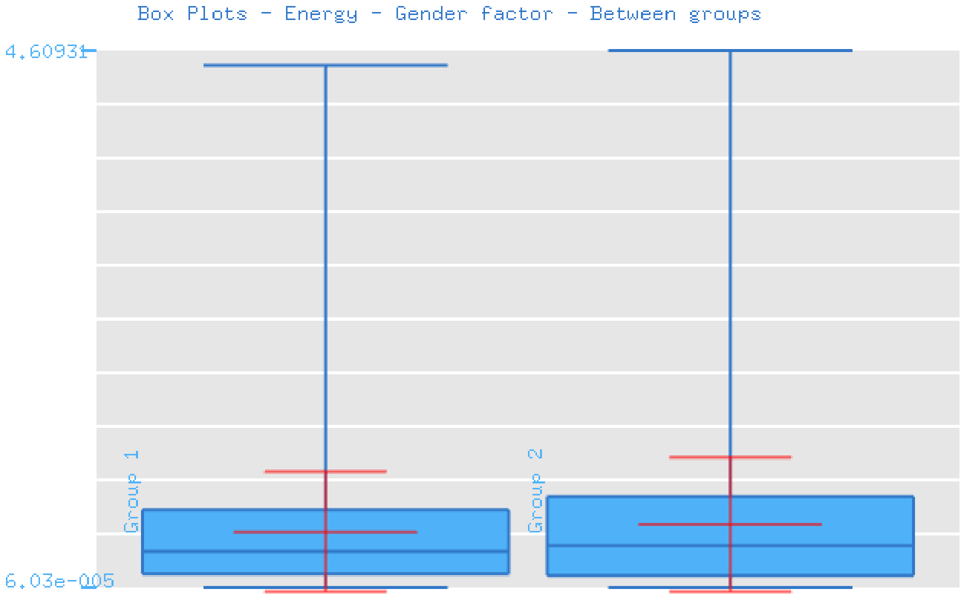


Figure 41 – Kinetic energy for each gender (women left, men right)

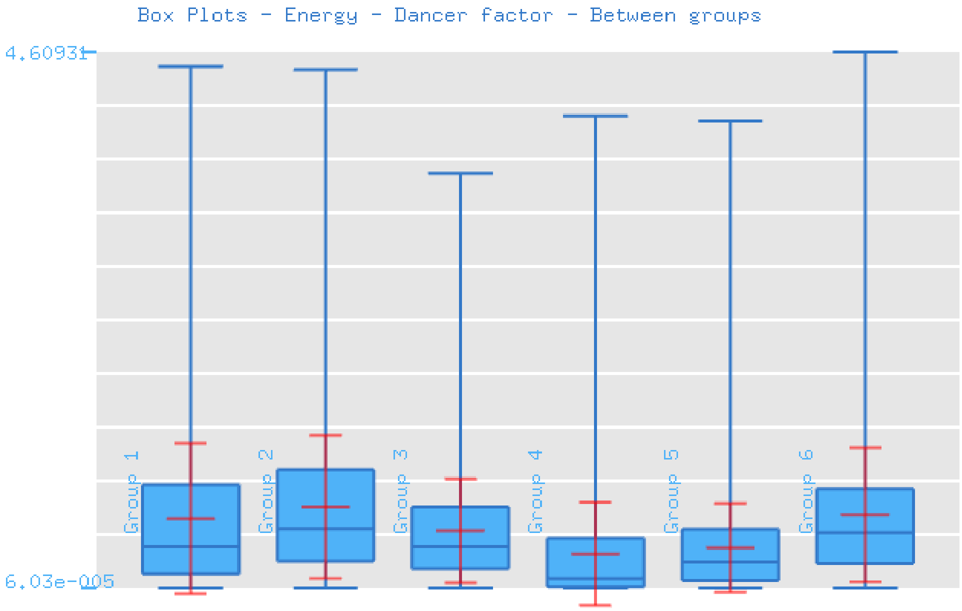


Figure 42 - Kinetic energy for each dancer

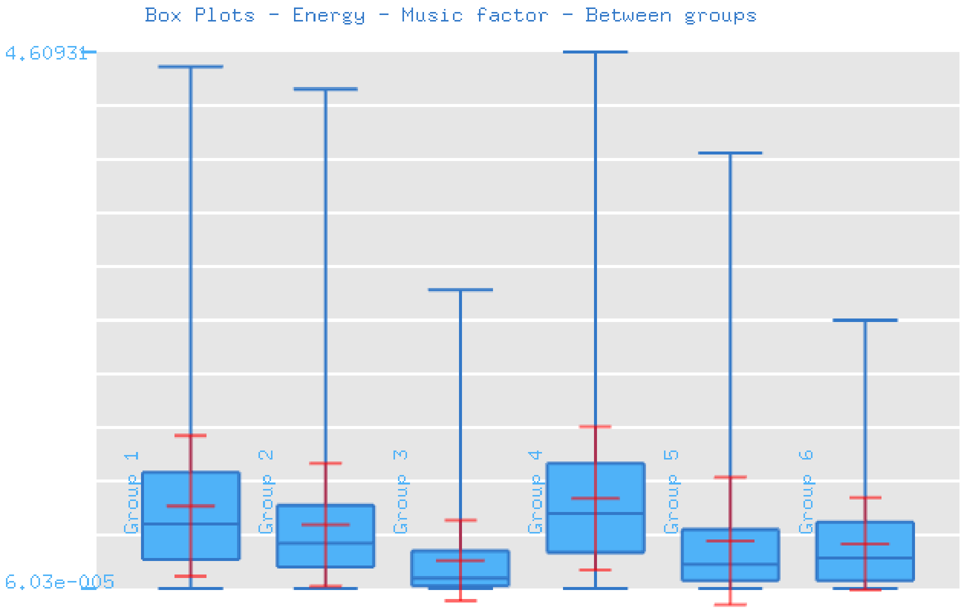


Figure 43 -Kinetic energy for each music

Music 1: 115, arousal +, valence +

Music 2: 308, arousal -, valence +

Music 3: 444, arousal -, valence -

Music 4: 486 Mamutot choreography

Music 5: 725, arousal 0, valence 0

Music 6: 902, arousal +, valence -

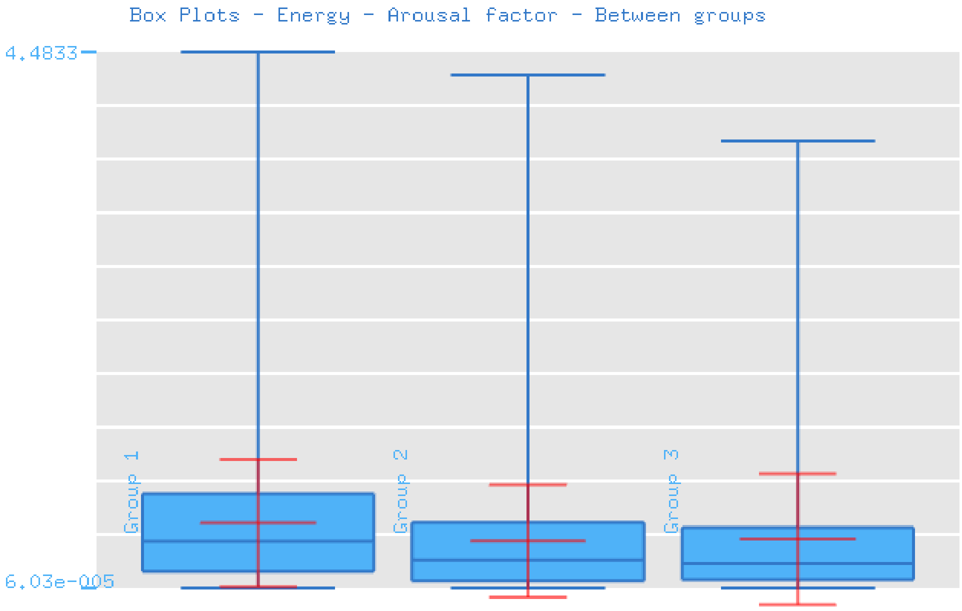


Figure 44 - Kinetic energy for each arousal level (+, -, neutral)

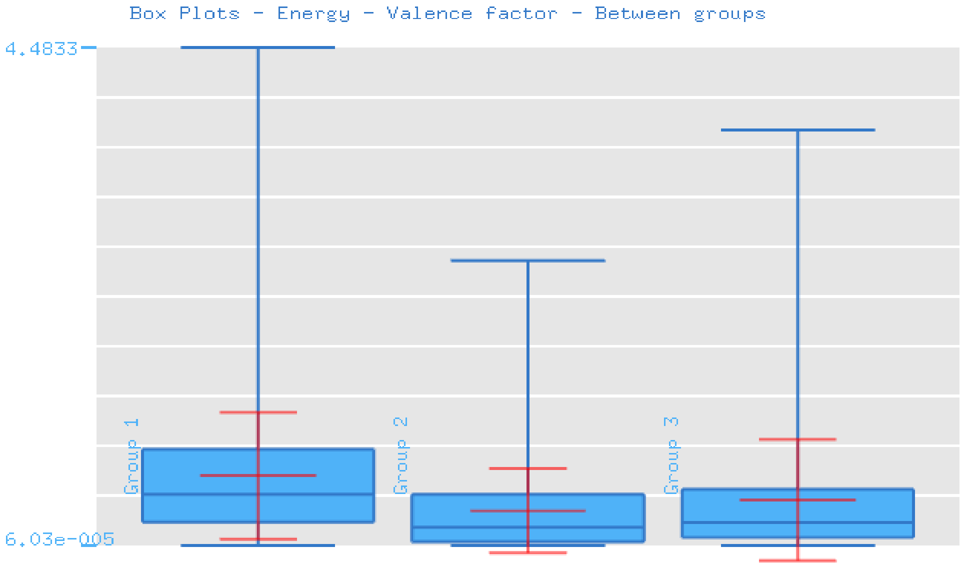


Figure 45 - Kinetic energy for each valence level (+, -, neutral)

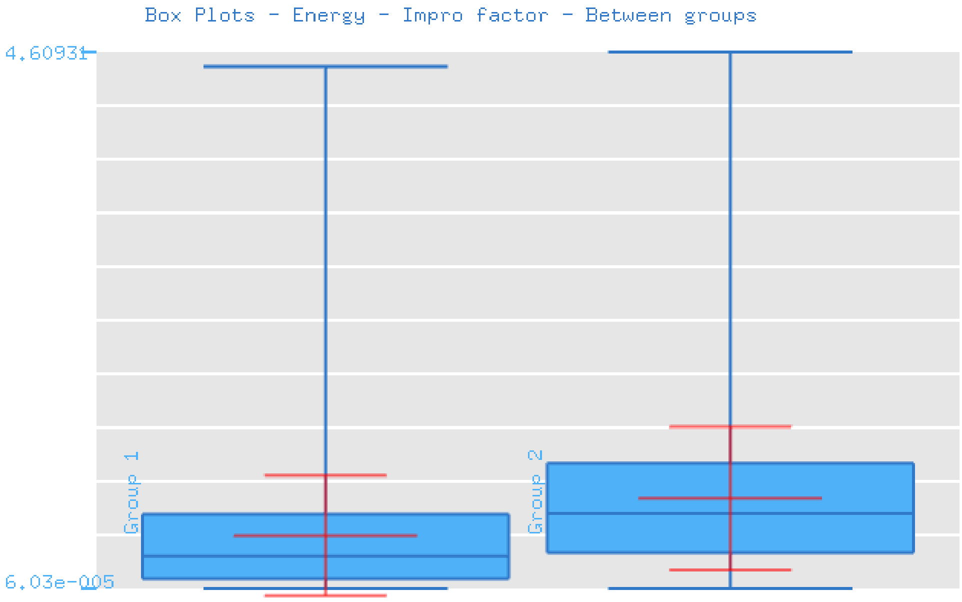


Figure 46 - Kinetic energy for improvisation (left) and choreography (right)

6 References

1. Z. Zhang: Microsoft Kinect Sensor and Its Effects. IEEE Multimedia, vol.19, no.2,4–10 (2012)
2. E. E. Stone and M. Skubic: Evaluation of an Inexpensive Depth Camera for Passive In-Home Fall Risk Assessment. International Conf. on Pervasive Tech. for Healthcare, 71–77 (2011)
3. Y. Kim, M. Lee, S. Nam and J. Park: User Interface of Interactive Media Art in a Stereoscopic Environment. Lecture Notes in Computer Science, vol. 8018, 219–227 (2013)
4. R. Laban: A Vision of Dynamic Space. London: The Falmer Press (1984)
5. M. Müller: Information Retrieval for Music and Motion. Springer-Verlag (2007)
6. J. Tilmanne and N. d'Alessandro: MotionMachine: A New Framework For Motion Capture Signal Feature Prototyping. Proc. of EUSIPCO 2015, Nice, France
7. N. dAlessandro et al.: Towards the Sketching of Performative Control with Data. Proc.s of the eNTERFACE Summer Workshop on Multimodal Interfaces, 2013.
8. B. Burger and P. Toiviainen: MoCap Toolbox A Matlab Toolbox for Computational Analysis of Movement Data. Proc. of the 10th Sound and Music Computing Conference, (SMC). Stockholm, Sweden: KTH Royal Institute of Technology (2013)
9. RAMToolkit, http://interlab.ycam.jp/en/projects/ram/ram_dance_toolkit
10. LibMapper, http://idmil.org/software/libmapper
11. armadillo, C++ linear algebra library http://arma.sourceforge.net/
12. mlpack, scalable machine learning library http://www.mlpack.org/
13. openFrameworks, C++ creative coding library http://www.openframeworks.cc/
14. U. of Cambridge. The hidden markov model toolkit(htk), 2009. http://htk.eng.cam.ac.uk
15. Merriam-Webster Dictionary: online, http://www.merriam-webster.com. (last consulted on December 1st 2015)
16. de Leva, P.: Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. J. Biomech. 29, 1223–1230 (1996)
17. Algorithm Implementation: Convex Hull https://en.wikibooks.org/wiki/Algorithm_Implementation/Geometry/Convex_hull/Monotone_chain
18. Algorithm Implementation: Findley's Polygon Area http://alienryderflex.com/polygon_area
19. Karg, M.; Samadani, A.-A.; Gorbet, R.; Kuhnlenz, K.; Hoey, J.; Kulic, D.: Body Movements for Affective Expression: A Survey of Automatic Recognition and Generation. Affective Computing, IEEE Transactions on , vol.4, no.4, 341-359, Oct.-Dec. 2013
20. Kee, D.; Karwowski, W.: LUBA: an assessment technique for postural loading on the upper body based on joint motion discomfort and maximum holding time. Applied Ergonomics, vol.32, no.4, 357-366, 2001.

21. Kee, D.; Karwowski, W.: Ranking systems for evaluation of joint and joint motion stressfulness based on perceived discomforts. Applied Ergonomics, vol.34, no.2, 167176, 2003.

22. Andreoni, G.; Mazzola, M.; Ciani, O.; Zambetti, M.; Romero, M.; Costa, F.; Preatoni, E.: Method fo r movement and gesture assessment (MMGA) in ergonomics. Digital Human Modeling, Lecture Notes in Computer Science, Vol.5620, 591-598, 2009.

23. Thierry Ravet, Joëlle Tilmanne, and Nicolas d'Alessandro. 2014. Hidden Markov Model Based Real-Time Motion Recognition and Following. In Proceedings of the 2014 International Workshop on Movement and Computing (MOCO '14). ACM, New York, NY, USA, , Pages 82 , 6 pages.

24. C. Larboulette and S. Gibet, A review of computable expressive descriptors of human motion. In Proceedings of the 2nd International Workshop on Movement and Computing (MOCO '15). ACM, New York, NY, USA, 21-28, 2015.

25. Fiebrink, R. Real-time human interaction with supervised learning algorithms for music composition and performance. PhD thesis, Princeton University, 2011.

26. Fiebrink, R., P. R. Cook, and D. Trueman. Human model evaluation in interactive supervised learning. Proceedings of ACM CHI, Vancouver, May 712, 2011.

27. Wright, M, Freed, A., Open Sound Control: A New Protocol for Communication with Sound Synthesizers. Proceedings of the International Computer Music Conference, 1997.

28. O. Mayor, J. Llop, and E. Maestre. Repovizz: amultimodal on-line database and browsing tool formusic performane research. InProceedings of Int.Symposium for Music Information Retrieval, 2011.

29. M. Bachynskyi, G. Palmas, A. Oulasvirta, J. Steimle, and T. Weinkauf, Performance and Ergonomics of Touch Surfaces: A Comparative Study using Biomechanical Simulation. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). 1817-1826, 2015.

30. D. Schwarz, G. Beller and Sam Britton: Real-Time Corpus-Based Concatenative Synthesis with CataRT In DAFx (2006)

31. Trueman, D. and R. Luke DuBois. "PeRColate." http://music.columbia.edu/PeRColate/

32. Cook, P. R. Synthesis ToolKit in C++, Ver-sion 1.0. InSIGGRAPH 1996, Course 17 and 18, Creating and Manipulating Sound to EnhanceComputer Graphics. Available from ACM SIG-GRAPH, 1996.

33. Hakonen, M.; Piitulainen, H. & Visala, A. Current state of digital signal processing in myoelectric interfaces and related applications Biomed. Signal Processing Control, Elsevier, 2015, 18, 334-359.

34. LEAP Company LEAP motion sensor, https://www.leapmotion.com, 2013.

35. Han, J., Shao, L., Xu, D., and Shotton, J. Enhanced computer vision with Microsoft Kinect sensor: A review. IEEE Trans. Cybernetics 43, 5 (2013), 1318–1334.

36. Thalmic Labs. Myo sensor, 2013. http://www.myo.com.

37. Kosmidou, V., and Hadjileontiadis, L. Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data. IEEE Trans. on Biomed. Engin. 56, 12 (2009), 2879–2890.

38. Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K., and Yang, J. A framework for hand gesture recognition based on accelerometer and EMG sensors. IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans 41, 6 (2011), 1064–1076.
39. Wang, N., Chen, Y., and Zhang, X. Realtime recognition of multi-finger prehensile gestures. Biomedical Signal Processing and Control 13 (2014), 262–269.
40. Zhang, Y., and Harrison, C. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In Proc. 28th Annual ACM Symposium on User Interface Software & Technology (2015), 167–173.
41. Tolksdorf, S. Myo javascript library, 2015. http://developerblog.myo.com/myo-diagnostics-page/
42. Mitsuhashi, N., Fujieda, K., Tamura, T., Kawamoto, S., Takagi, T., and Okubo, K. BodyParts3D: 3D structure database for anatomical concepts. Nucleic acids research 37, suppl 1 (2009), D782–D785.
43. Sheffield, S. Human anatomy and physiology, 2015. http://www.getbodysmart.com/
44. Tiansijie, S. Csg plugin for three.js, 2015. https://github.com/tiansijie/ThreeCSG
45. Zhang, C., and Chen, T. Efficient feature extraction for 2d/3d objects in mesh representation. In Proc. Intern. Conf. Image Processing, vol. 3, IEEE (2001), 935–938.
46. Cabello, R. Three.js WebGL framework Javascript library, 2010. http://threejs.org
47. Schnur, D. Attractive javascript plotting for jquery, 2015. http://www.flotcharts.org/
48. Google Data Arts Team. Dat.gui - a lightweight graphical user interface for changing variables in javascript, 2013. https://github.com/dataarts/dat.gui
49. Ververidis, D. Resources for myo3dhand, 2015. https://www.youtube.com/watch?v=4NDe0cXXPAQ
50. Myo market, https://market.myo.com/