WHOLODANCE

Whole-Body Interaction Learning for Dance Education

Call identifier: H2020-ICT-2015 - **Grant agreement no**: 688865

Topic: ICT-20-2015 - Technologies for better human learning and teaching

Deliverable 3.7

Second report on software platform and libraries

Due date of delivery: June 30th, 2018

Actual submission date: July 17th, 2018

Start of the project: 1st January 2016
Ending Date: 31st December 2018

Partner responsible for this deliverable: LYNKEUS

Version: 1.0

Dissemination Level: Public

Document Classification

Title	Second Report on Software Platform and Libraries
Deliverable	3.7
Reporting Period	M15-30
Authors	Stefano Piana, Ksenia Kolikhalova , Michele Buccoli
Work Package	WP3
Security	Public
Nature	Report
Keyword(s)	Software platform, software libraries, Eyesweb, Unity, Java script

Document History

Name	**Remark**	**Version**	**Date**
Stefano Piana		V0.1	15/06/2018
Ksenia Kolikhalova	Added game restricted graph theory approach updates	V0.2	19/06/2018
Michele Buccoli	Added Data Driven libraries	V0.3	28/06/2018
Stefano Piana	Summary, model-driven methodologies	V0.4	29/06/2018
Stefano Piana	Applications, integration libraries	V0.4	1/07/2018

List of Contributors

Name	**Affiliation**
Stefano Piana	UNIGE
Ksenia Kolikhalova	UNIGE
Michele Buccoli	POLIMI

List of reviewers

Name	Affiliation
Oshri Even Zohar	MOTEK
Edwin Morley-Fletcher	Lynkeus

Executive Summary

This deliverable describes the movement analysis libraries and applications developed in the context of the project.

Section 1 introduces the report and lists its objectives, Section 2 describes the developed libraries and software modules for model-driven and data-driven movement analysis. Section 3 provides an overview of developed modules for the integration of the modules with the partner's Unity-based project and sonification environments, finally Section 4 gives an overview on the developed applications and demonstrations.

Table of Contents

1 Introduction

This deliverable serves to summarize the advancements in the development of the movement analysis libraries and software modules in the context of the WhoLoDancE project; this document focuses on the description of the software libraries for the automatic analysis of movement principles, movement dimensions and qualities identified in D1.6 and described in D3.5, on the description of libraries and modules for the integration of the feature extraction modules with other tools developed by the partners and on the description of applications that use such libraries in the context of the project.

2 Model-driven and Data-driven analysis libraries

2.1 Model-driven libraries

2.1.1 Feature extraction modules (patches)

This section will describe modules developed to extract movement features derived from the list of movement principles described in D1.6 Different modules where developed to analyse data coming from recorded motion capture sequences (taken from the repository), real-time from motion capture devices, and real-time from low-cost capture devices such as IMU sensors.

2.1.1.1 Input Sources

The developed analysis modules can handle data coming from a variety of sources:

- Motion Capture recordings from FBX files, captured using professional motion capture systems or low-end sensors (i.e, Notch sensors).
- Live Streams coming from IMU sensors, Inertial measurement units equipped with accelerometers, gyroscopes and magnetometers (I.e, X-OSC, smartphones, tablets)
- Live Streams captured by low-end motion capture devices (I.e, Microsoft Kinect, RGB-D Cameras, MYO sensors)

<u>Motion Capture recordings from FBX files</u>

A module for the extraction of motion captured sequences stored in FBX file format was developed and is now distributed with EyesWeb XMI 5.7.2.0, The module consists of an EyesWeb Block [Camurri 2000] that reads data from an input FBX file, Figure 1 shows the block as presented in the EyesWeb environment; The block has the following parameters:

- Filename: the input file
- Activation type, the block can be controlled basing on frames, time or in polling mode
- EoF behaviour, Loop, or stop.
- Joint orientation extraction, Local orientation or global orientation.

Figure 1: The FBX extractor block

The block extracts the whole set of tracked points (first output) at each frame and the associated orientations (second output).

The FBX files can also be read with standard (FBX native tools and 3D packages) software tools like MAYA, Xsi, Motionbuilder, and 3D max. in addition The UNITY engine platform can also read those files.

Live streams coming from IMU sensors

Data coming from IMU or smartphones live streams is managed via the OSC protocol, 9 axes IMU are received as a single OSC message: Figure 2_shows the module that receives data streams from IMU sensors and smartphones: the first three inputs correspond to the three gyroscope components, the second three outputs correspond to the measured acceleration and the last three outputs are the measured magnetic field.

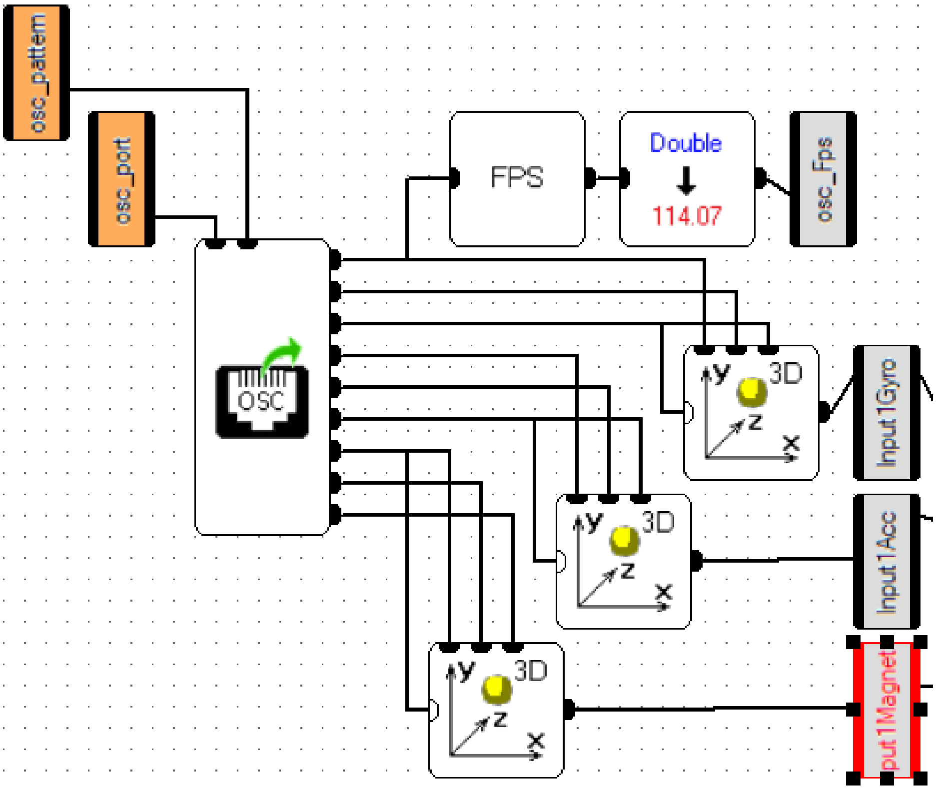

Figure 2: OSC receiving module

To manage data coming from MYO sensors we developed a module to directly stream data captured by the sensor's internal IMU and myograph sensors.

Figure 3: Myo sensors extractor module

Live Streams captured by low-end motion capture devices

The EyesWeb environment can extract motion capture data from Kinect (V2)[1] sensors and other RGB-D low-cost cameras such as the Intel Real Sense[2] cameras. Figure 1 shows the blocks that extract data from the Kinect V2 and Real Sense Depth cameras, the output include colour images, depth maps, user silhouettes and coordinates (2D or 3D).

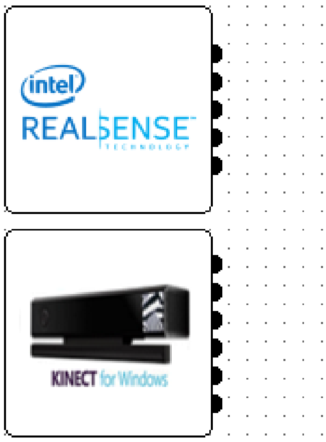

Figure 4: Modules for motion capture from Realsense and Kinect v2 Sensor

2.1.1.2 Feature Extraction modules

A series of software modules and EyesWeb Patches has been developed to extract model-based approximations of movement principles and movement dimensions, the extraction of such quantities is performed as follows:

- The input data (Mocap, IMU) is pre-processed and normalized
- Movement features and quantities are extracted from the pre-processed data
- The extracted movement-related quantities are given as output in the following possible formats: live OSC streams, CSV or Json files, UDP streams.

The following model-based movement dimensions are extracted by means of EyesWeb XMI modules, definitions of the features and relations with movement principles are specified in D1.6:

- Low-level: speeds, accelerations, angular velocities, displacement.
- Mid-level: Energy, Smoothness, Dynamic Symmetry, Weight, trunk stretch and torsions.
- High Level: Lightness, postural tensions, origin of movement, balance, transmission of energy.

Figure 5 shows the EyesWeb Patch that extracts Mid-level (Energy, Smoothness, Weight, Dynamic symmetry) movement dimensions from FBX files.

[1] https://developer.microsoft.com/en-us/windows/kinect
[2] https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html?_ga=2.211764629.1200355089.1530715893-1469761141.1530715883

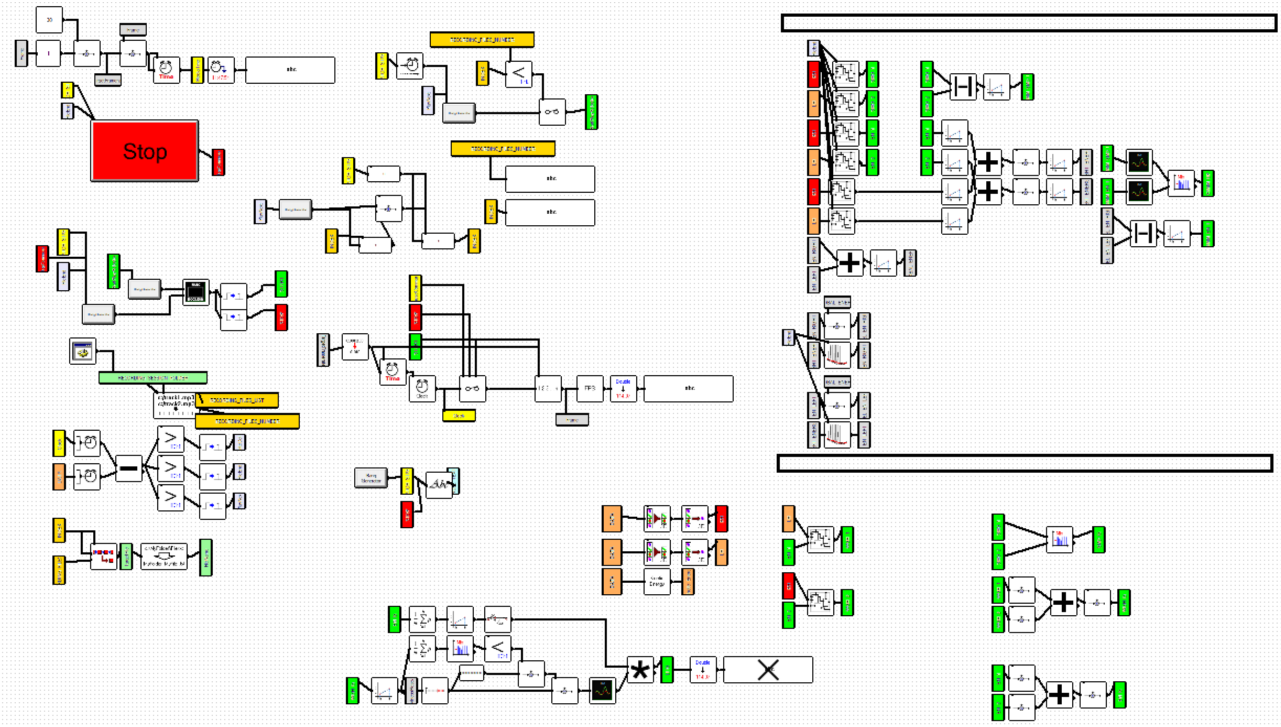

Figure 5: Eyesweb Patch that extracts mid-level model-based movement dimensions from FBX files

2.1.2 Updates on model-based methodologies

This section of deliverable provides the information about Evaluation Online Survey that was designed, in order to validate that the proposed approach in Work Package 3 of Graph-Restricted Game is an efficient way to identify most important joints during the movement sequence. We evaluated the relevance of our method to the concepts of ”origin of movements” and ”movement propagation”.

<u>Recall: Method of Graph-Restricted Games</u>

A computational method for the analysis of expressive full-body movement qualities is introduced, which exploits concepts and tools from graph theory and game theory. The human skeletal structure is modelled as an undirected graph, where the joints are the vertices and the edge set contains both physical and nonphysical links.

Physical links correspond to connections between adjacent physical body joints (e.g., the forearm, which connects the elbow to the wrist). Nonphysical links act as “bridges” between parts of the body not directly connected by the skeletal structure, but sharing very similar feature values. The edge weights depend on features obtained by using Motion Capture data. Then, a mathematical game is constructed over the graph structure, where the vertices represent the players and the edges represent communication channels between them. Hence, the body movement is modelled in terms of a game built on the graph structure. Since the vertices and the edges contribute to the overall quality of the movement, the adopted game-theoretical model is of cooperative nature.

A game-theoretical concept, called Shapley value, is exploited as a centrality index to estimate the contribution of each vertex to a shared goal (e.g., to the way the movement quality on which, on a case-by-case, one focuses, is transferred among the vertices).

Visualization

In order to simplify the evaluation of the algorithm and make it visually observable, we generated video sequences portraying a skeletal representation of a performer, where at each video frame, one of the joints is highlighted with a red colour, so that the highlighted joint is the joint with the maximum Shapley value among the joints set extracted by the proposed method for that frame.

The proposed method is applied to a data set of 100 Motion Capture movement recordings with length from 10-15s (Figure 6), segmented out of the Genova Recording Session (March 2016).

In the following table can be found the number of the recordings and segmented range of seconds that were used in order to apply the method of Graph- Restricted Games.

Table 1: Number of trials and segments range

Recordings 21 March 2016	Recordings 22 March 2016
Trial 003: 18-29, 29-37,37-44, 48-59, 59-67	Trial 002: 03-18, 21-33, 36-50
Trial 006: 09-23, 23-31	Trial 003: 02-17, 20-33, 36-49
Trial 008: 13-21, 23-34, 38-47	Trial 004: 04-14, 14-24, 26-36, 36-48, 49-59
Trial 010: 03-10,	Trial 005: 03-21, 23-39, 45-56
Trial 012: 05-14, 14-25, 25-37, 37-46, 46-54, 54-64	Trial 006: 03-12, 12-20, 27-44, 50-60,60-70
Trial 014: 05-15, 15-24, 24-34, 34-45	Trial 007: 18-28, 29-36, 36-43, 50-59
Trial 016: 06-15, 16-27, 30-41, 42-51	Trial 008: 07-14, 14-21, 21-29, 29-37
Trial 018: 12-22, 30-40, 47-54	Trial 009: 07-19, 19-28, 29-38, 39-48, 49-58, 58-66
Trial 021: 13-19, 22-29, 33-42, 48-56, 60-68, 72-82, 82-91, 96-107	Trial 011: 05-15, 15-26, 27-37, 37-47,48-58, 58-68, 69-79, 79-89, 89-99
Trial 022: 15-26, 26-37, 37-51, 51-61. 61-72, 74-86, 86-98	Trial 013: 05-13, 13-21, 21-29
Trial 025: 01-09, 14-22, 32-41, 45-55, 57-65, 72-81	Trial 014: 05-13, 13-22, 22-31, 31-41, 41-52
Trial 026: 35-45, 45-54, 54-61, 62-67, 72-80, 80-88, 94-102	Trial 016: 05-14, 14-23, 23-33, 33-43

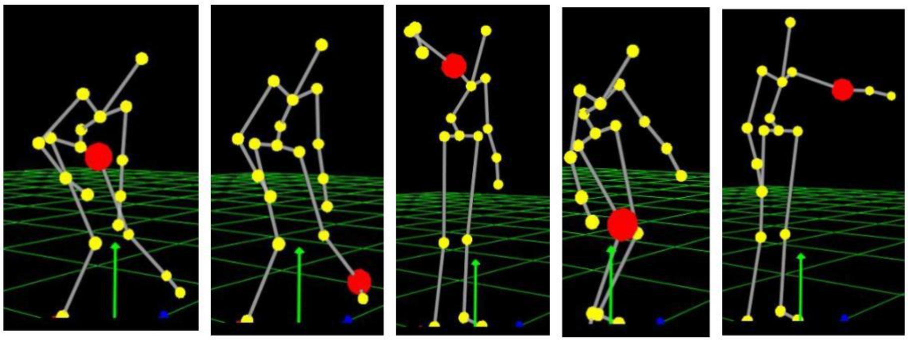

Figure 6: These sequence of images represents a dance sequence and the corresponding transition of the highlighted joint, which represents the joint with the maximum Shapley value at each frame.

<u>Survey Website for evaluation</u>

In order to evaluate the proposed method, we designed a survey website: **(http://www.infomus.org/Tools/OriginOfMovementThreeVideos/index.php)** to collect user ratings on the developed method. This survey is designed in order to investigates the perception of the origin of movement, and relevance of our proposed method for identifying important joints to the concept of "origin of movements" .

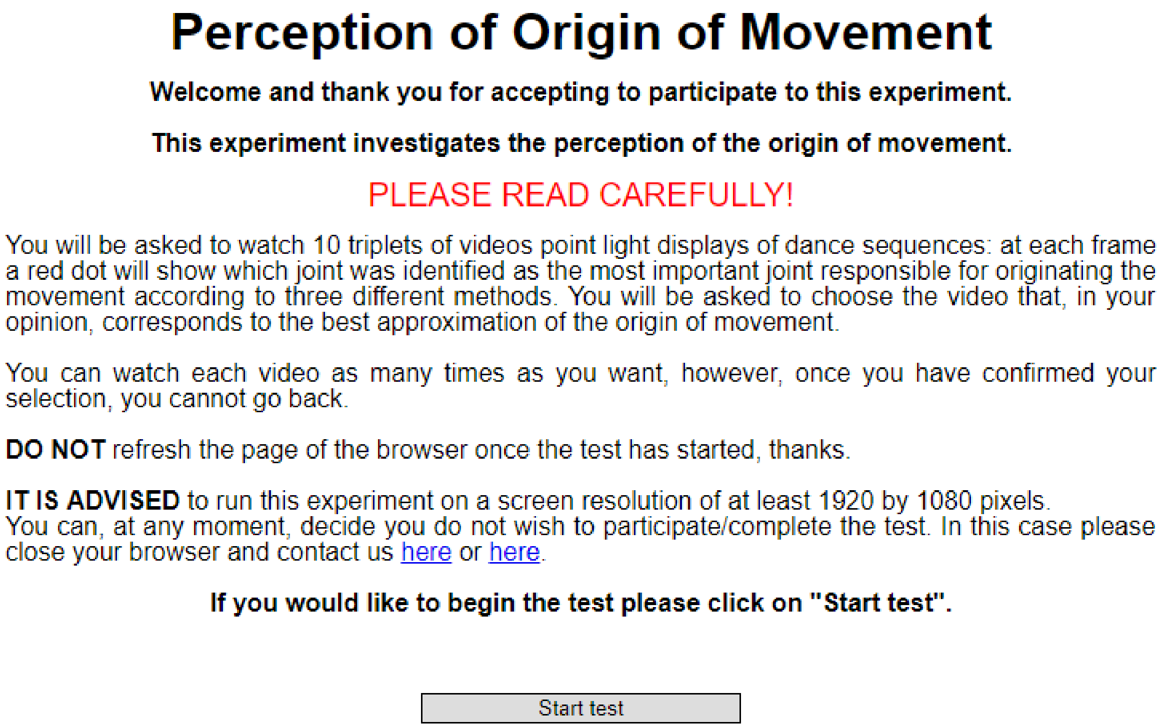

Figure 7:Website for the evaluation of method, introductory page

As the user visits the page, a description of the task is given (visible in Figure 7), then the user can choose to proceed with the task or quit it at any moment. No sensible data is collected during the procedure.

Once the user accepts to participate and perform the task, a series of triplets of videos is proposed (Figure 8). Each among the three videos displays a skeletal representation of a dancer performing the same dance sequence. Each video has one highlighted joint (in red). Such a joint corresponds to the most important joint according to one of the following criteria: (i) maximum Shapley value (ii) maximum velocity, and (iii) random choice. The order of the three criteria is randomized among the three videos, so that the specific criterion applied to each video is unpredictable.

The participant is asked to pick the video that better represents the evolution of the most important joint responsible of originating the dancer's movement. Once a user has selected one video, he/she is also asked to declare how much she is confident about her choice by selecting a value from 1 to 5 on a Likert scale. The participant can see all the videos as many times as desired, and has to answer both questions (video choice and confidence) before proceeding to the next triplet of videos. Each participant has to rate ten triplets of videos proposed from a selection of one hundred triplets, using a Latin square selection method.

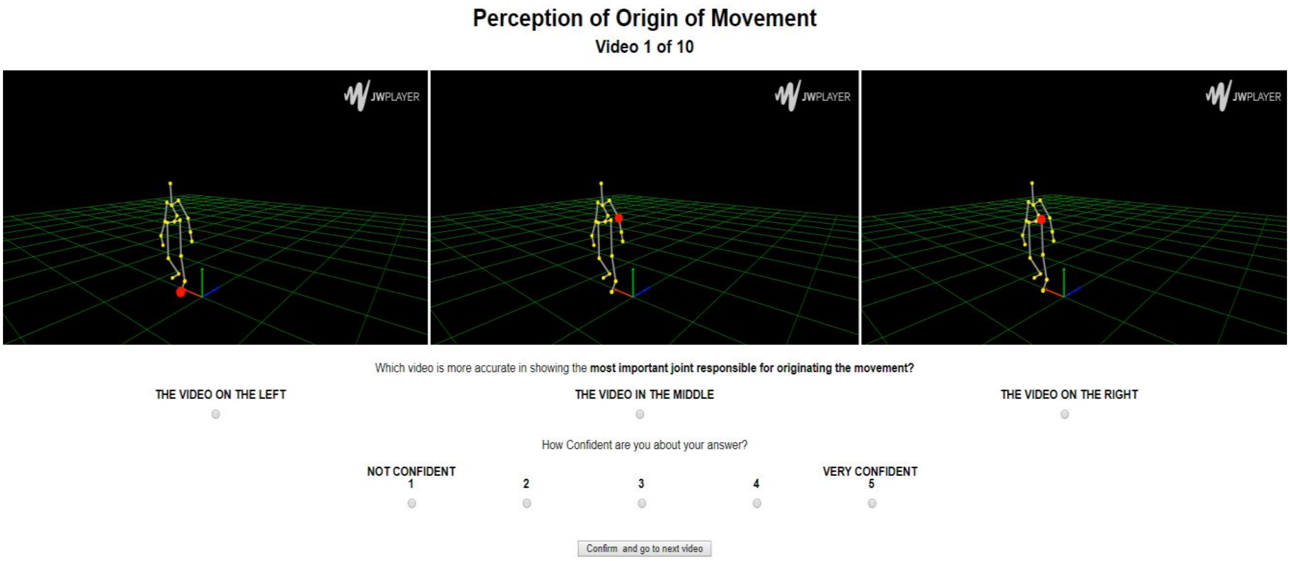

Figure 8:Website for the evaluation of the proposed method

Results of the questionnaire are saved on a .csv file, including a unique Id of the participant. The file is generated when the participant accepts to perform the task. For each of the ten proposed triplets of videos, the following information is collected:

- Id of the dance sequence proposed;
- order used to propose the three alternative videos, and corresponding extraction method
- used (Shapley, velocity, random);
- participant's choice among the three videos;
- participant's confidence.

<u>Participants</u>

The website was submitted to dance experts and choreographers with different level of expertise: professionals, semi-professionals and beginners. The evaluation of the methods was performed in the framework of the H2020 ICT WhoLoDanCe project by international partners. Moreover, the additional participants (dance practitioners) from Genova, that used to collaborate with Casa Paganini Infomus Research Center in the previous projects, also took part in evaluation.

Total number of 22 participants took part in the evaluation. They had specified different level of expertise, out of 3 possibilities: professional, semi-professional and beginner.

The general information is the following:

1) Professionals: 8 participants (3 male, 5 female), with age – mean 42,75 years, std 9,56 years.
2) Semi-Professional: 6 participants (3 male, 3 female), with age mean 30 years old and std 4,47 years.
3) Beginners: 8 participants (6 male, 2 female), with age mean 35,5 years and std 7,4 years.

In the following Table 2 the detailed information about participants is presented.

Table 2: Participant's informations

	Participant Gender	**Age**	**Expertise**
#1	male	25	semi
#2	male	30	beginner
#3	female	24	beginner
#4	male	40	professional
#5	male	28	semi
#6	male	45	professional
#7	female	36	professional
#8	female	37	professional
#9	male	56	professional
#10	female	35	semi
#11	female	29	beginner
#12	male	43	beginner
#13	female	27	semi
#14	male	34	beginner
#15	male	39	beginner
#16	male	29	semi
#17	female	28	professional
#18	female	55	professional

#19	female	45	professional
#20	female	36	semi
#21	male	40	beginner
#22	male	45	beginner

Evaluation Results

We recall, that participants of the online experiment had to select among the three videos displays-skeletal representation of a dancer performing the same dance sequence. Each video had one highlighted joint (in red) that corresponds to the most important joint according to one of the following criteria: (i) maximum Shapley value (ii) maximum velocity, and (iii) random choice.

The order of the three criteria were randomized among the three videos, so that the specific criterion applied to each video was unpredictable.

We constructed survey in this way, in order to have three different types of stimuli. This allowed us to investigate, if the proposed computational method for the analysis of expressive full-body movement qualities is perceived better than naive visualization of maximum velocity. We particularly were interested in investigation the concept of "origin of movements". "Origin of movement" is the point at which a movement is said to originate. This refers to specific body parts, which can be either distal (in the limbs or head) or central (in the torso).

The results of the validation of proposed method are promising.

Based on the opinion of dance experts, we were able to conclude that Shapley value method was selected in the majority of the cases. The extracted meaningful joints via computation of Shapley value proved to be relevant to the concept of "origin of movement in dance", based on different level expertise dancers' opinions.

The detailed results of the choice among the 3 types of different stimuli are presented in the Table 3 and on the diagram shown on Figure 9.

Table 3: The choice of participants (in percentage)

Participants\Method	Shapley Value	Velocity	Random
Professional	90	8,75	1,25
Semi-Professional	83,33	10	6,66
Beginner	67,5	25	7,5
All Participants	80	15	5

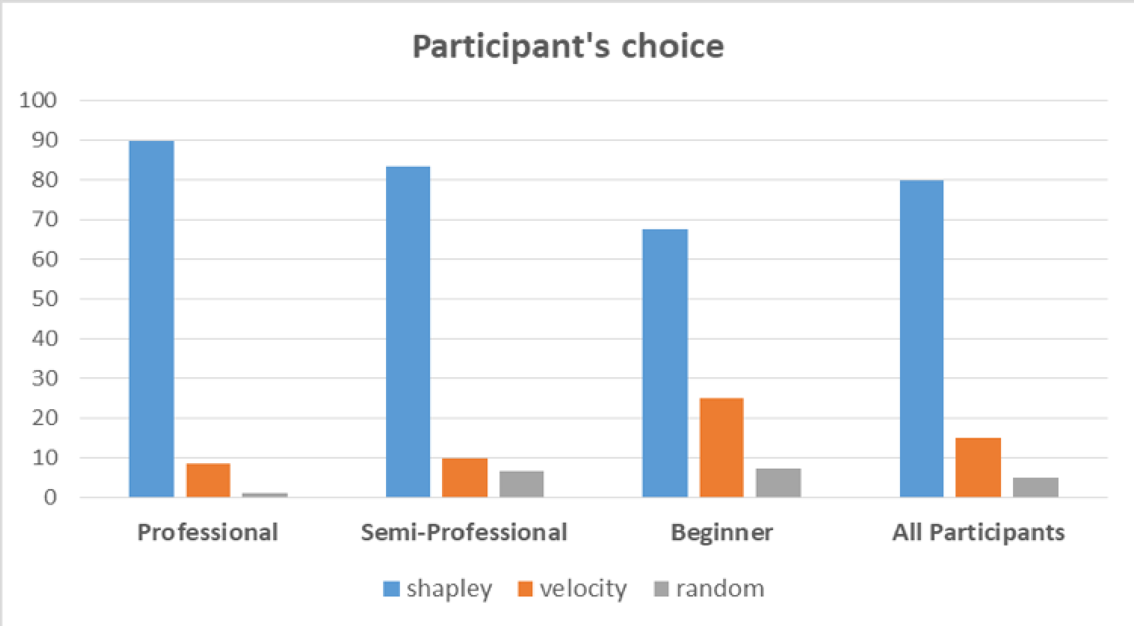

Figure 9: The choice of the participants (in percent)

All participants, despite the different level of professionalism in dance, selected visualization of joint with maximum Shapley value, as a video that is more accurate in showing the most important joint responsible for originating the movement (selected in 80 percent of the cases).

The collected results also showed, that our proposed method was selected more often proportionally to the level of expertise of dancers. Higher the level of expertise - in more cases visualization corresponding to the "Shapley value joint" was selected.

Moreover, the participants showed to be quite confident in the choices they did. In 45,45 percent of the cases they were confident and 41,48 percent - neutral. The diagram is shown on the Figure 10.

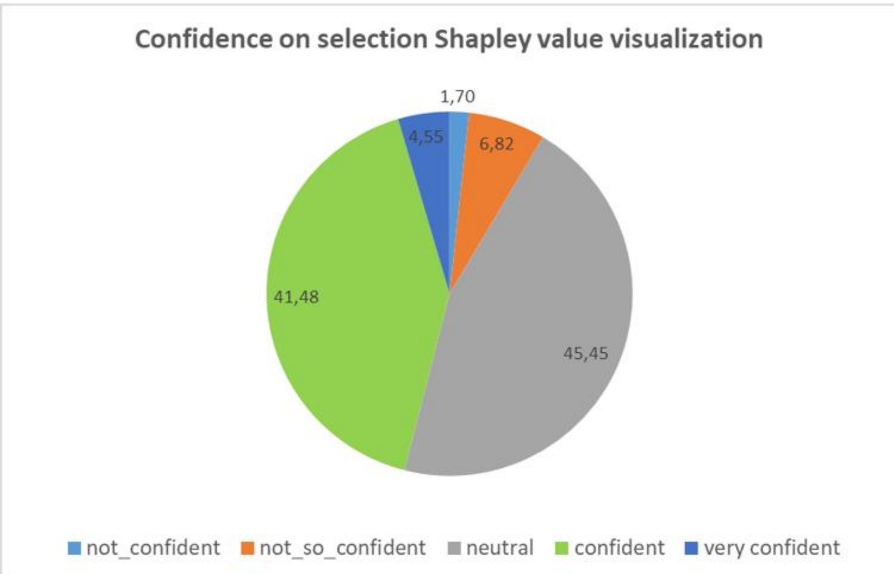

Figure 10: Diagram of confidence level of respondents

The results of application of proposed method to a dataset of movement segments are promising, and show that, on the data set used for its preliminary evaluation, the method was able to extract meaningful joints.

By observing the visualized results on the moving skeleton, we can notice the relevance of the extracted joint with respect to the concept of "Origin of movement in dance" (see D1.6), which was defined as the point at which a movement is said to originate. This refers to specific body parts, which can be either distal (in the limbs or head) or central (in the torso).

2.2 Data-driven libraries

In the WhoLoDancE project, data-driven methodologies involve using manually-annotated examples to train machine learning techniques to understand the relationship between a low-level representation of the motion and its high-level description. This involves analysis of the annotations, extraction (computation) of the numerical representation (*features*), using algorithms for feature selection and machine learning and, finally, evaluate the results. We detail the libraries we use in the following subsections.

For the sake of the data-driven methodologies for extraction of movement qualities, we develop the first prototypes using the Python language. We chose Python because of its high-level semantics and the wide popularity among data scientists and researchers, which make Python flexible from prototyping algorithms to developing the final applications. A summary of the libraries and their use is shown in Table 4.

2.2.1 Analysis of annotations

Having a large number of annotations from different users, we may need to analyse the consensus among them, i.e., how much they agree on the annotations, in order to spoil possible inconsistent annotations that would make the training to fail because of ambiguous data. For example, if the same movement is rated as "very fluid" by one choreographer and "not fluid at all" by another one, the algorithms would hardly find the relationship between the same motion and two extremely different descriptions.

For the analysis of the segmentation, we used the Python library **Pandas**, which provides high-performance data structures and data analysis tools [McKinney 2011]. We stored the annotations as a DataFrame structure, which is a kind of table that can be manipulated using Pandas' methods. From this table, it was easy to spot the ambiguities by looking at overlapping annotations with highly different values.

2.2.2 Extraction of features

For the data-driven methodologies, we considered the performances acquired through motion capture (MoCap files) and stored as FBX files, a proprietary filetype from Autodesk. In order to handle this files with Python, we developed a library that loads FBX files in a MoCap structure. The library makes use of Python wrappers for the **FBX Software Development Kit**, and the MoCap structure was described in ***D4.3 Analysis and Integration of Generic Application Framework***. Just as a reminder, the structure stores information about the 3D position of dancer's limbs over time, as well as their 3D orientation.

From the 3D position and orientation, we performed signal processing techniques for extracting a set of features that are representative of the motion. In this regard, we used the popular Python libraries **NumPy** and **SciPy** [Oliphant, 2006]. The former provides structures for numerical computation, such as arrays and matrices, with the most common operation, such as mean, raising to power, square roots, etc. The latter provides more advanced methods for scientific computation, including several signal processing techniques such as Fast Fourier Transform, filtering operations, etc. These libraries were also used in post-processing stage, to clean the output of the prediction and make it more robust to fluctuations.

2.2.3 Training the algorithms

A machine learning algorithm should be designed to focus on the main pattern in the data, while neglecting the small fluctuations of the input data. This design is crucial to make the algorithm robust to noisy data, and able to generalize to examples that the algorithm was not trained with. For this reason, we need to use feature selection strategies, which selects the most salient features, or a combination of them, that contains the greatest amount of information. After that, we use the set of selected features to train the data-driven algorithms. We cannot predict which algorithm is going to perform better and, as a matter of fact, different scenarios may require different kinds of algorithms.

The **Scikit-learn** package [Pedregosa 2011] is an extremely popular Python library for the machine learning community. It implements several state-of-the-art techniques for feature selection, regression (predicting an output in a continuous range of values), classification (predicting an output in a finite set of values). Moreover, it also provides an easy interface for a set of common actions that are performed during training, such as split the dataset into training and validation set, cross-validation training, building a pipeline of processing and evaluating the predictive power of the algorithm by comparing the predicted and the expected output, computing the most common performance metrics in information retrieval.

The Scikit-learn package, however, does not implement deep learning techniques, which are a family of machine-learning techniques that is achieving state-of-the-art performance in many tasks. Deep learning requires GPU-acceleration to speed up the process using parallelism, and a large number of annotated examples to understand the pattern and the relationship between input and output. While the current status of annotations does not contain such numbers, we expect to collect more annotations in the next months. We have started investigating deep learning techniques using the package **Keras** [Chollet 2015], which provides an interface for developing deep networks. We used Keras with TensorFlow [Abadi 2016] as a layer of numerical GPU-accelerated computation.

Table 4: List of libraries used for data-driven methodologies

Library	Description	Use
Pandas	Data structures and data analysis tools	Storing and analysis of the annotations
Numpy	Data structures and numerical computation	Extraction of features
Scipy	Scientific computation, including signal processing techniques	Extraction of features
Scikit-Learn	Algorithms and techniques for feature selection, machine learning, evaluation of performance	Training the algorithms; evaluation the performance

Keras and TensorFlow	Interface for building deep learning techniques; lower layer for numerical computation.	Training the algorithms

3 Integration Libraries

This section will describe libraries developed to integrate The EyesWeb XMI based libraries and tools with the Unity and sonification environments used by the project's partners.

3.1 Unity Integration and communication library

In order to integrate the movement analysis modules with applications and tools developed by partners, UNIGE developed a series of modules to receive and send data to the Unity environment.

The following data can be streamed and received between the Unity and the EyesWeb environments

- Avatar joints coordinates and rotations
- Float quantities, matrices and timeseries (i.e, timeseries of an extracted movement dimension)
- Primitive types (i.e., booleans, integers)

This allows the analysis of any movement of an animated avatar or of any data captured in real-time through both environments, the generation of 3D, VR and AR displays. Figure 11 shows an example of a 3D visualization in Unity of an avatar animated using one of the dance sequence downloaded from the

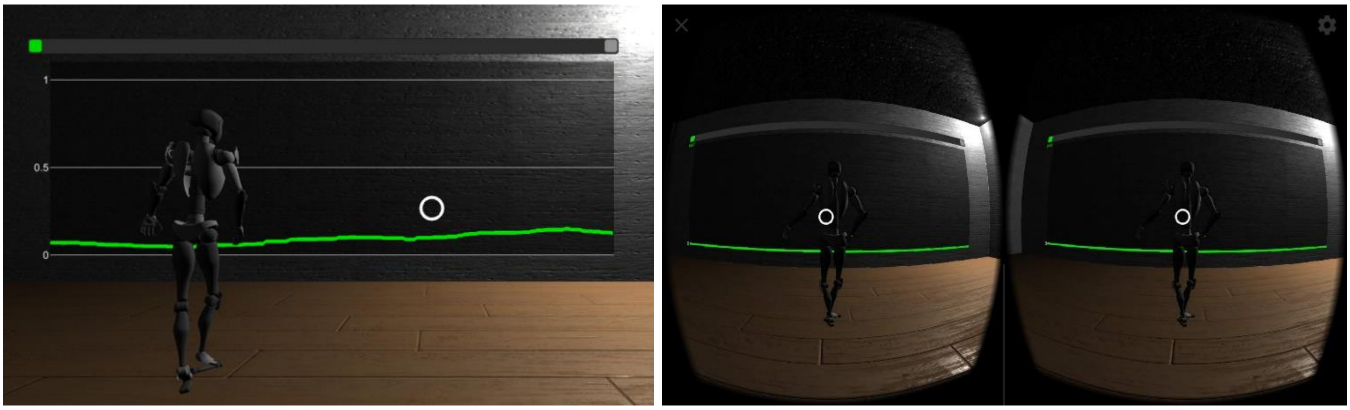

Figure 11: 3D (left) and VR (Right) visualization of a dance sequence and corresponding plot of a movement feature (Energy)

WhoLoDanCe repository analysed by Eyesweb XMI modules, this visualization can be displayed on pc, smartphones using a google cardboard[3] VR headset (Figure 11) or more advanced VR headsets (i.e., HTC Vive[4]) and MR headsets (Hololens[5]).

[3] https://vr.google.com/cardboard/
[4] https://www.vive.com/us/
[5] https://www.microsoft.com/en-us/hololens

3.2 Sonification environments communication protocol and libraries

The feature extraction modules can be interfaced with various sonification and electronic music composition environments (Max[6], Pd[7], supercollider[8]), the most commonly used way of communicating between EyesWeb XMI and such environments is the OSC protocol [Wright 2005], all the extracted movement qualities and dimension are sent through OSC using the following pattern for the OSC tag:

*/SourceDeviceType/DeviceId/***features***/FeatureName*

Where the device type "*SourceDeviceType*" can be MoCap, IMU, MYO; the "DeviceId" is used to identify the corresponding device in case there are multiple devices of the same type and the feature name identifies the specific movement dimension. The schema in Figure 12 shows how the communication between the EyesWeb environment and sonification environment is implemented.

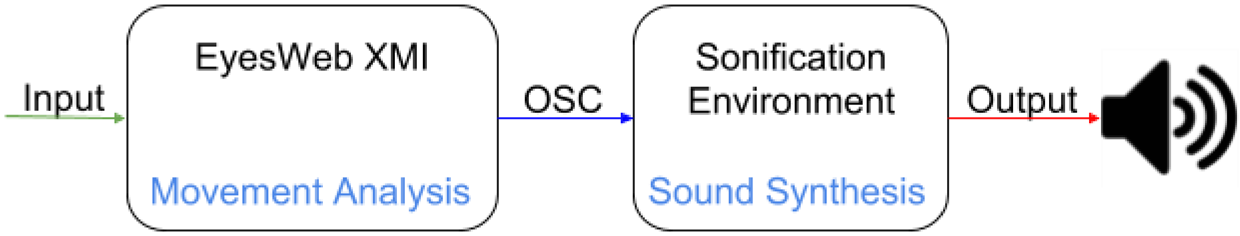

Figure 12: Communication schema between EyesWeb XMI and sonification environments

4 Applications

4.1 Movement Sketching

The movement sketching tool was developed to exploit the potentials of the feature extraction modules developed by UNIGE and the similarity search engine developed by Peachnote, using it dance practitioners, students, and professionals can create their own recordings of dance sequences (by performing them), and query the repository in order to find similar dance segments. This allows the users to compare, correct and integrate their interpretation and the ones of professionals and teachers. It makes use of the movement analysis libraries to extract movement dimensions and is integrated with the similarity search engine provided by Peachnote to query and get recordings that are similar to the ones performed by users basing on the extracted movement dimensions.

Users can record movements using low-end motion capture systems (e.g., Notch Sensors) and query the system to get similar movements in terms of movement and movement qualities. The tool is (at the moment) a standalone application that runs on a PC. Results can be viewed on a PC or on low cost VR displays such as Google Cardboard (more VR displays such as HTC VIVE and/or Microsoft Hololens are planned to be supported in the near future).

[6] https://cycling74.com/products/max/
[7] https://puredata.info/
[8] https://supercollider.github.io/

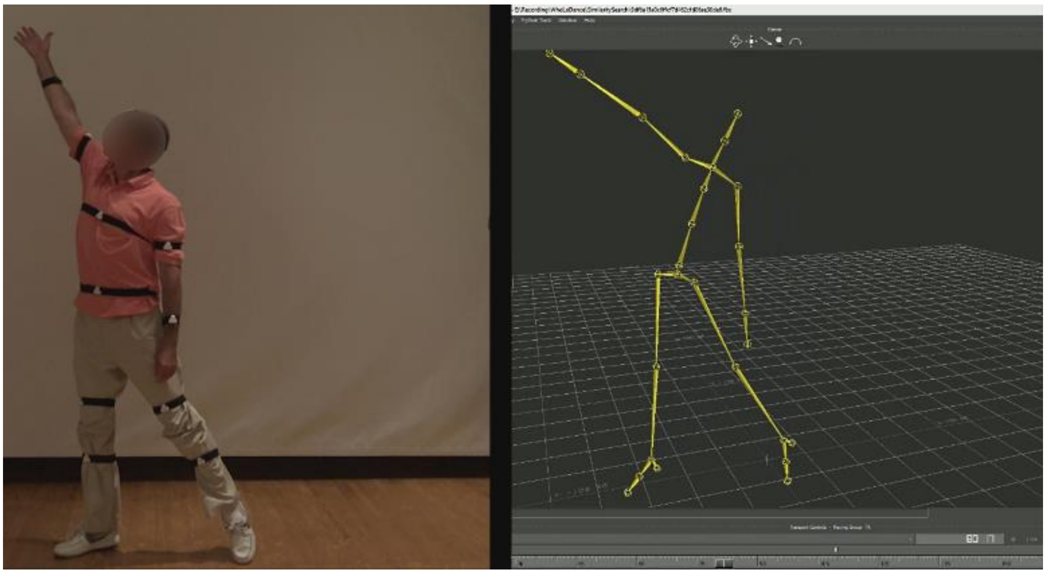

Figure 13: A user recording a dance sequence with low-cost capture deices (left) and the capture results (right)

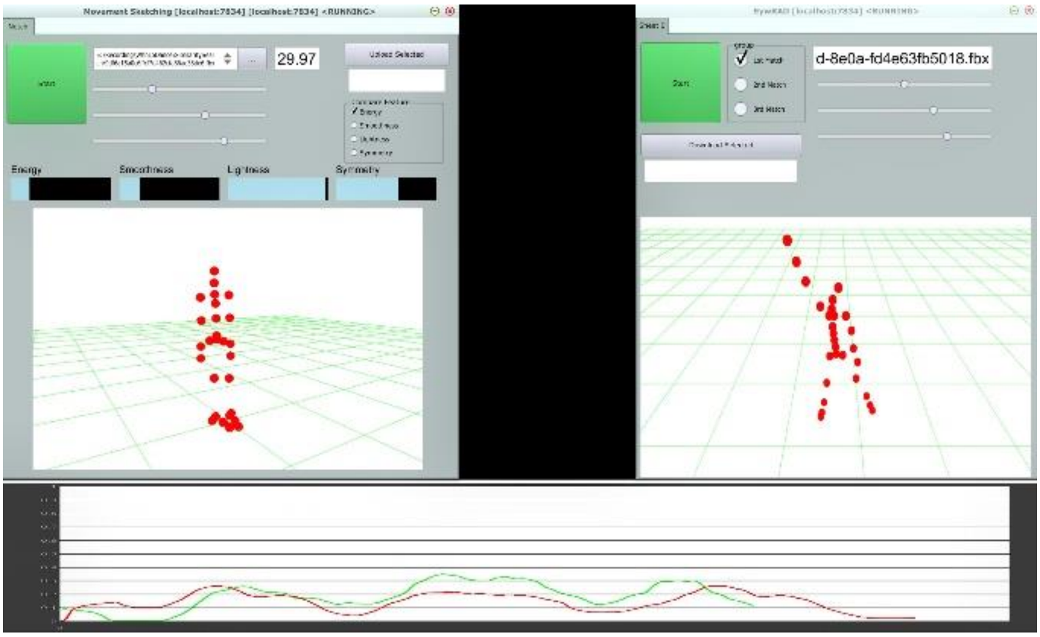

Figure 14: the first version of the movement sketching application GUI: the captured movement sequence is analysed (left) and the analysis results are shown (on the right)

4.2 (Real-Time) Movement Sonification

This application was developed to exploit and demonstrate the usage of real-time data acquisition from low-cost capture devices and movement analysis.

A performance demo stages an experimental dialog between a dancer and a computer-based system that is able to react to the qualitative dimensions of dance movements, internally linked to specific body states and emotional expressions. The demo explores the creative possibilities of interactively controlling sound synthesis and video in order to monitor both expressive dance and minute body movements. Such an interactive relationship is interesting since it links the musical outcome to the expressivity of movement, so essential in contemporary dance.

In this application, the analysis modules are interfaced with Supercollider which is used to generate and control the real-time sonification. The demonstration acquires the movement of a dancer by means of a Kinect V2 sensor, 4 IMUs, and two Myo sensors, the movement of the dancer influences a visual output and controls the sonification. The following qualities are extracted and used to control the visual and acoustic output: energy, muscular tension of arms, symmetry, weight, lightness.

5 References

[Camurri 2000] Camurri, Antonio, et al. "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems." Computer Music Journal 24.1 (2000): 57-69.

[McKinney 2011] McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." Python for High Performance and Scientific Computing (2011): 1-9.

[Oliphant, 2006] Oliphant, Travis E. A guide to NumPy. Vol. 1. USA: Trelgol Publishing, 2006.

[Pedregosa 2011] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[Chollet 2015] Chollet, François. "Keras." (2015): 128.

[Abadi 2016] Abadi, Martín, et al. "TensorFlow: A System for Large-Scale Machine Learning." OSDI. Vol. 16. 2016.

[Wright 2005] Wright, Matthew. "Open Sound Control: an enabling technology for musical networking." Organised Sound 10.3 (2005): 193-200.