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 3.4

Report on multimodal signal modelling

Due date of delivery: December 31st, 2017

Actual submission date: December 31st, 2017

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

Partner responsible for this deliverable: UNIGE

Version: 0.2

Dissemination Level: Public

Document Classification

Title	Report on multimodal signal modelling
Deliverable	D3.4
Reporting Period	M1-M24
Authors	Stefano Piana, Antonio Camurri
Work Package	WP3
Security	Restricted
Nature	Report
Keyword(s)	Multimodal, Data analysis, framework

Document History

Name	Remark	Version	Date
Stefano Piana	First version of the document	0.1	10/12/2017
Anna Rizzo	Final reviewed version	0.2	29/12/2017

List of Contributors

Name	Affiliation
Stefano Piana	UNIGE

List of reviewers

Name	Affiliation
Oshri Even Zohar	Motek
Antonella Trezzani	Lynkeus
Anna Rizzo	Lynkeus

Executive Summary

This deliverable serves to summarize the input devices, data formats and methodologies adopted in the process of developing algorithms, software modules and applications for movement principle and qualities analysis.

Section 1 introduces the report and lists its objectives whereas Section 2 gives an overview on the data capture systems that are used in the context of the project; in particular, a description of professional motion capture systems, used during the production of the WhoLoDancE repository, and low-end capture devices, that are used in the low-cost applications, is given.

Section 3 introduces the methodology followed in the design and development of movement analysis algorithm and software modules, in particular a conceptual framework is described where the qualities of movement are organized in a hierarchical way, going from physical signals to abstract, complex concepts.

Table of contents

1. Introduction

This deliverable serves to summarize the advancements in the development of multimodal data analysis techniques in the context of the WhoLoDancE project, this document will describe how data was captured and manipulated to be compatible with developed multi-modal analysis: data may be captured by a variety of different input devices, described in Section 2, that are characterized by very different performances and costs, from the expensive but very reliable optical motion capture system used to get professional-grade motion captures of movement to affordable devices that render the developed tools accessible by end-users.

After the definition of suitable input devices (Section 2), this document will describe the methodology adopted for analysing the captured data: Section 3 will introduce a computational framework to compute qualities and analyse multi-modal data in a hierarchical way, starting from physical signals to more abstract concepts and qualities.

2. Input devices for multimodal data acquisition and analysis

This Section will give an overview on the input and capture systems that were used for data acquisition and for the subsequent analysis of movement qualities and principles.

Professional capture systems

Two professional motion capture systems were used in the framework of the project, one installed at UNIGE's laboratory and one installed by Motek at Schram Studio in Amsterdam.

UNIGE capturing platform

The overall architecture for multimodal recordings is shown in the following figure.

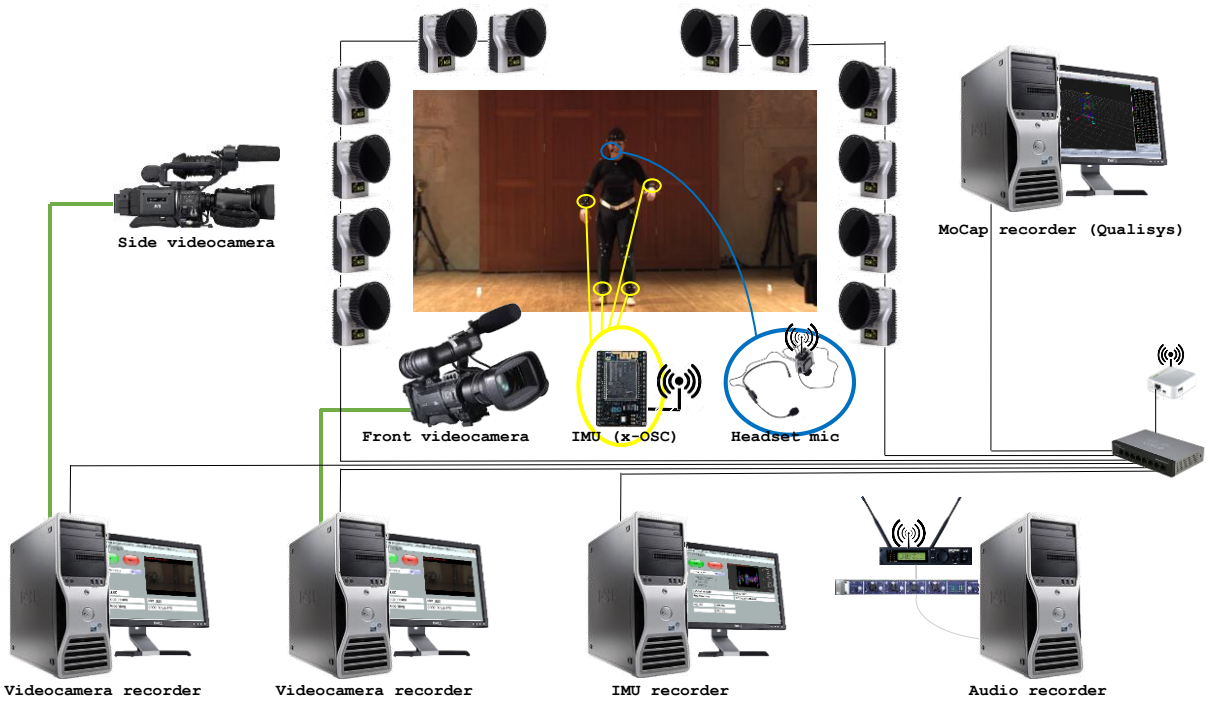

Figure 1. the architecture of the recording platform used by UNIGE

The performer movements were captured by a Motion Capture system (*MoCap recorder* in the picture). The performer also wore a headset microphone, which was used to record breathing noise for possible further analysis. Moreover, the performer also wore Inertial Measurement Units (IMU). Finally, two broadcast quality video cameras were observing the scene, one from the front and one from a side.

Synchronization was guaranteed by the EyesWeb XMI platform. On the MoCap recorder computer, EyesWeb was used to generate the reference clock used by all other recorders. The generated reference clock was sent to the other device in a format compatible with each specific device. As an example, the Qualisys Motion Capture system receives such clock encoded in an audio stream, in SMPTE format. Also, the two broadcast video-cameras and the *Audio recorder* used SMPTE encoded as an audio signal. The *IMU recorder* received the reference clock via network, through the OSC protocol.

To guarantee synchronization EyesWeb kept track, for every recorded frame or sample, of the timestamp when the data was received. As a matter of facts, not all streams can be hardware-synchronized (e.g., with a genlock signal), thus a software synchronization is performed by EyesWeb by keeping track of the time at which the data was received in a separate file, and using such information when playing back the data. IMU sensors or Kinect are examples of devices which were synchronized in this way.

Each computer of the recording system run an EyesWeb XMI application, that allowed to record multimodal data (video, audio, motion capture, sensors).

MOTEK capturing platform

In order to go through the motion-captures processes at Motek, a multi-system setup, with large capture volume (up to 25 x 25 Meters) was available. The type of mocap that was used, was the Passive optical motion capture, with the use of a VICON[1] T160 camera based system combined with a VICON VERO cameras system synchronized and captured on a workstation running Vicon BLADE2.0 data acquisition and analysis software suite.

[1] https://www.vicon.com/

Figure 2. Schram Studio in Amsterdam

The recording made at Schram studio included motion capture, video from two full-HD video cameras with audio (ambient).

Data captured by the MoCap systems were saved on the workstation running Vicon BLADe 2.0 and streamed to a second workstation running Autodesk MotionBuilder that was used to render live feedbacks given to the dancers.

Data formats of professional motion capture systems

Regarding the data formats, raw capture from both Vicon and Qualisys systems was converted to two agreed formats: FBX[2] and C3D[3]. Both Vicon and Qualisys can generate this file type. In particular, data recorded by Motek was recorded directly in FBX format then converted to C3D, meanwhile UNIGE data was recorded to C3D then retargeted to a compatible format and finally converted to an FBX file format, in order to be used throughout the project, for visualization and interactive projection. The C3D format is probably one of the most commonly used formats for that purpose. More specifically, C3D is a binary or ASCII file format for motion capture data used in animation, biomechanics and gait analysis to store motion capture data. The format is flexible enough to store 3D coordinates and any numeric data in a single file. However, C3D format has been developed specifically for motion capture, in addition to the FBX format, which is for 3D animation in general. C3D and FBX formats can be easily used in analysis and visualization frameworks like Unity and EyesWeb XMI or converted in other data formats more suitable to be managed in web-based applications developed in JavaScript programming language or by general purpose languages like Python.

Regarding our files, they contain a fully articulate skeleton, including the finger bones. Using a naming scheme compatible with Autodesk MotionBuilder[4] (FBX) was selected, in order to be used as an avatar real-time throughput for the volumetric projection.

[2] https://www.autodesk.com/products/fbx/overview
[3] https://www.c3d.org
[4] https://www.autodesk.com/products/motionbuilder/overview

So, the master data format that was used throughout the project is the FBX. FBX is a framework that allows someone to create, edit, and manage asset templates. An asset template defines the interface of an asset. In other words, it specifies the properties that an asset must have, in order to comply with a specific asset type.

FBX is designed to describe animation scenes and is supported by many 3D animation software packages to transfer files among them. It can contain geometries, textures, cameras, lights, markers, skeleton, and animation. One large advantage that MotionBuilder has over other 3D animation packages is that it can take any of the other file formats, such as the C3D format in our case, and translate them into the .fbx format. This allows MotionBuilder to work as a type of "universal translator" between not only different animation systems, but also different types of skeletal structures.

Data formats of video, audio and sensors

Video recordings were taken with professional video camera systems:

- recordings made by UNIGE include video cameras (720p @ 50FPS) encoded in mp4 h264 video codec and included ambient audio in aac format.
- recordings made by Motek included dual video cameras (1080p @ 30 FPS) encoded in mp4 h264 video codec and included ambient audio in aac format.

Sensors data, when available were recorded in CSV format, where each line of the file represented a frame of the sensor data, including the timestamp at which the data was recorded.

Low-end capture systems

In order to give access to the developed tools and application to more people without the need of having a professional motion capture system, a set of low-cost capture devices have been identified and selected to be used.

Kinect V2 RGB-D sensor

Microsoft Kinect represents a cheap and easy to use motion capture system, can provide full-body motion capture of multiple users at the same time, but its reliability and precision of measurement are far worse than the professional optical motion capture systems described above, these drawbacks had to be taken into account while designing and developing applications and movement qualities extraction module, production of the sensor has been discontinued in 2017 thus the use of alternatives (i.e., Intel RealSense[5]) is under investigation.

[5] https://www.mouser.it/new/Intel/intel-realsense-ZR300-dev-kit/

Figure 3. Microsoft Kinect For Windows V2 Sensor

Notch Sensors

Notch sensors (Figure 4) are an inertial-based motion capture kit that provide full-body motion capture: a set of 11 sensors allows the capture of movements and produces .fbx files that can be imported in the main animation and modelling software, including MotionBuilder.

Figure 4. A set of six notch sensors that allows upper body tracking

IMU Sensors

IMU sensors can be used to capture quantities related to movement such as accelerations and angular velocities of limbs. If correctly placed on a performer's body, they can provide means of extracting movement dimensions. They are a cheaper though less reliable alternative of motion capture systems: x-io xOSC sensors[6] (Figure 5) are an example of IMU sensors that have been used in the scope of the WhoLoDancE project.

Figure 5. X-io XOSC a 9-axis IUM sensor

Data formats for movement analysis

The data captured by devices can be sent to the analysis modules in two different ways, starting from real-time streams or off-line (i.e., streams stored on file or online repositories). In both cases, we distinguish different kinds of data-streams.

- Motion capture streams: these kind of data streams contain information about joint position and orientation (e.g., the absolute position of a joint or its orientation); live streams would be received as a sequence of frames that contain the orientation and/or position of each joint, while in the case of off-line streams the information can be read from csv, fbx or json files.
- Audio-visual streams coming from video/audio capture devices such as video cameras, microphones, or read from multimedia files of different formats (i.e., mp3, mp4, etc.).
- RAW data streams coming from sensors: these streams can contain data captured by IMU, physiological and/or other kind of sensors, in this case, for the real-time processing, each sensor streams each captured quantity, while off-line analysis will be computed from single column csv files containing raw sensor data.

3. Analysis methodologies for multimodal data analysis

This section introduces a computational framework that was followed in developing data- and model-driven algorithms and techniques to extract movement qualities from the available data from both high- and low-end capture sensors.

[6] http://x-io.co.uk/x-osc/

Multi-Layered Computational Framework of Qualities in Movement

Within the WhoLoDancE project, several movement principles and qualities were proposed (see D1.6). We now propose a Multi-Layered Computational Framework to analyse those quantities.

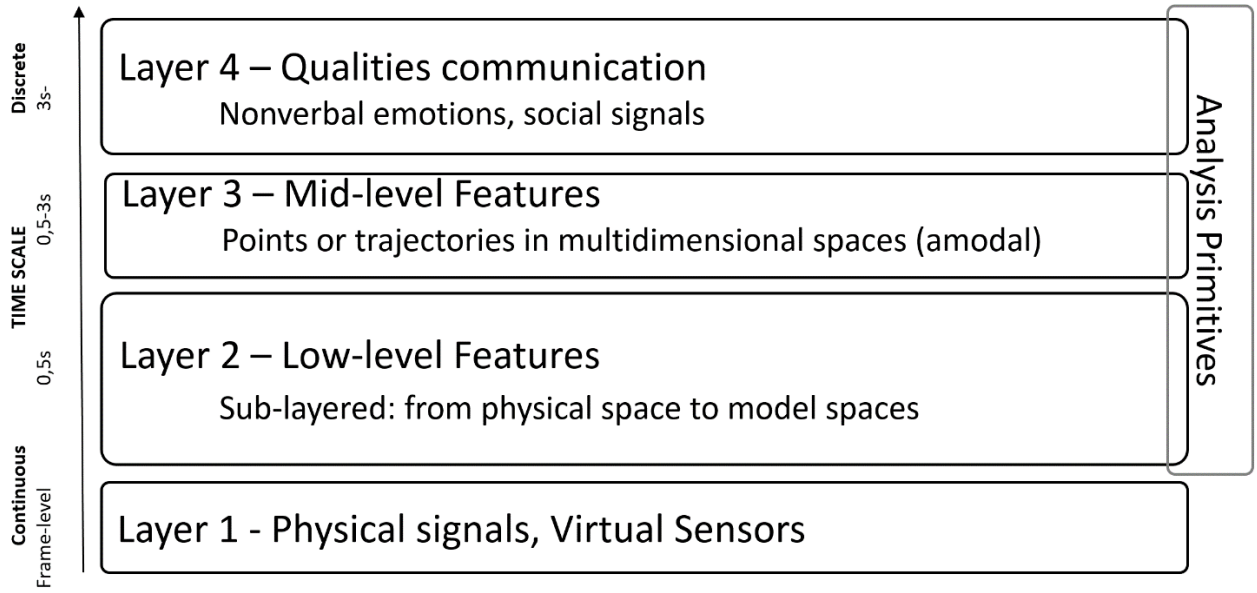

Figure 6. Multi-Layered Computational Framework of Qualities in Movement

It consists of four different layers and addresses aspects of movement analysis at different spatial and temporal scales:

> **Level 1 - points**: **physical data** that can be detected by (real or virtual) sensors in real-time (for example, position/orientation of the body planes).
>
> **Level 2 - Low-Level Features: Time-series**, not subject to interpretation, detected uniquely starting from instantaneous physical data on the shortest time span needed for their definition and depending on the characteristics of the human sensorial channels.
>
> **Level 3 - qualities**: **perceptual features**, starting from physical and sensorial data from level 2, computed on larger time intervals (typically 0.5-3s).
>
> **Level 4 - affects**: **perceptual and contextual features**, keeping into account context and narrative structure, i.e. how different qualities evolve along time. This usually requires a large time span.

The analysis methodologies of movement principles and qualities proposed in D3.5, are following this framework, where principles are organized hierarchically in spatial and temporal aspects.

Analysis of movement qualities

The work carried out in the framework of the WhoLoDancE project was characterized by a strict collaboration with choreographers and dancers in order to get inspiration in both identifying the vocabulary of principles and in the definition and refinement of algorithms for their automated measurement.

The qualities are organised in a hierarchical structure that follows the framework presented before (see Figure 1).

Layer 1 – Physical Signals: Virtual sensors

Layer 1 (*Physical signals*) grounds on the concept of *virtual sensor*, understood as a single physical sensor (or as the integration/fusion of data from many physical sensors) combined with signal conditioning (e.g., de-noising and filtering), and with techniques for extraction of specific raw data. For example, an RGB-D physical sensor (e.g., Kinect) may be associated with virtual sensors providing the 3D trajectories of specific body parts, the silhouette of the tracked bodies, and the depth image. At layer 1 thus data is captured by an array of virtual sensors, associated to a broad range of physical sensors, including motion capture, video cameras, microphones, and physiological sensors. We characterize each virtual sensor with its sampling rate and with the data it provides (e.g., an image, a 3D position, an acceleration, a numeric sample, an audio or a physiological signal).

Layer 2 – Low-Level Features: Time-series

Layer 2 (*Low-level Features*) receives the raw data from the array of virtual sensors at layer 1 and extracts a collection of features characterizing movement locally in time. That is, low-level features are usually computed instantaneously on the raw data, or on small buffers of a few samples, by using a sliding-window approach with maximum overlap. Thus, low-level features are represented as time-series having usually the same sampling rate as the raw data they are computed from. Time-series may be either univariate (e.g., kinetic energy) or multivariate (e.g., the *x*, *y* and *z* components of velocity).

Layer 3 – Mid-level features: perceptual features

Whilst analysis at layer 2 is local in time, layer 3 (*Mid-level Features*) deals with structural aspects, i.e., it computes features that either describe one single movement unit or, if movement units cannot be easily identified (e.g., in a dance performance consisting of a continuous stream of tightly interlaced movements), operates on time windows which are long enough to grab the dynamic evolution of movement along time.

Furthermore, features at layer 3 are at such a level of abstraction that they represent *amodal* descriptors, i.e., the level where perceptual channels integrate. This mean that, for example, *Fluidity* is a meaningful feature to characterize both audio and movement. Amodal descriptors enable the design of mapping strategies from movement to the sonic domain: we can analyse a movement starting from physical signals (layer 1) up to layer 3, and then we can map features at layer 3 back down to the physical signal in the sonic domain. Analysis and processing at layer 3 goes through two basic steps: segmentation and computation of amodal features.

Segmentation. The segmentation step identifies the analysis unit for layer 3. This can either be a single movement unit (e.g., a gesture) in a stream of movements or a time window of a defined duration. In the former case, segmentation may operate at different levels, which means that a movement unit may be, e.g., a single movement or a whole phrase. Depending on how segmentation is performed, layer 3 produces different outputs. If single movement units are isolated, these are conceived as events. This means that it is not possible to determine a sampling rate anymore. Rather, each single event is associated with a given time, typically the time instant when the movement unit ends. An array of values of features is associated with each of such events; this means that the output of layer 3 is, in this case, a specific position in a multidimensional feature space or, in other words, a location in a multidimensional map. If, instead, analysis is still performed on time windows, such windows are either not overlapped or partially overlapped. A sampling rate can still be determined, based on windows duration and overlap, and an array of values of features is computed for each time window. In this case, the output of layer 3 is a trajectory in a multidimensional feature space, or in other words, a path in a multidimensional map. Features computed at layer 2 are usually employed to perform segmentation.

One of the simplest techniques consists in analysing kinetic energy by applying a possibly adaptive threshold. More sophisticated techniques exploit, e.g., machine learning approaches, where a vector of values, obtained by applying analysis primitives to layer 2 time-series, is used to train and feed recognizers for distinguishing between pauses and movements. In case real-time analysis is not needed and an archive of dance performances is available, manual annotation can be carried out when automatic segmentation is not accurate enough.

Computation of features. Two major approaches are applied for computing mid-level amodal features:

- direct computation of mid-level features specifically defined and grounded on low-level features and/or physical signals (e.g., smoothness is involved in the computation of fluidity);

- application of *analysis primitives* to one or many low-level features. Unary operators can be applied, e.g., to retrieve salient events, and to estimate the complexity of a movement by computing, for example, sample entropy (Richman & Moorman, 2000) on one or more time-series of low-level features; see e.g., (Glowinski, et al., 2011). Binary and n-ary operators can be applied e.g., for measuring the relationships between time-series of low-level features computed on the movement of different body parts (limbs). For example, synchronization techniques are applied to evaluate coordination between hands (the so called intra-personal synchronization) or coordination between dancers in a group (i.e., inter-personal synchronization). Causality provides information on whether the movement of a joint leads or follows the movement of another joint in the body, or it can explain the leadership of a dancer or of the movement of a musician in a group (Glowinski, Gnecco, Piana, & Camurri, 2013). Predictive models are applied, e.g., to estimate the extent at which actual movement corresponds to or violate expectations (i.e., something related to tension).

Layer 4 – Expressive Qualities

While the previous layers focus mainly on features at a growing level of abstraction from Layer 1 to Layer 3, this layer mainly focuses on the nonverbal communication of movement qualities to an external observer. *Memory* and *Context* are factors that intervene mainly at this layer, characterized by observation within layered and longer time intervals. Both memory (the history of previous movement qualities) and context may influence how an external observer perceives and interprets a feature in terms e.g., of expectancy (Camurri, Krumhansl, Mazzarino, & Volpe, 2004), saliency (unexpected, rare, contrasting movements, may contribute to raise the sensitivity to specific movement features), and sensitivity (stillness may raise the sensitivity to very tiny movements). The factors may be modelled as possible biases in the measure of a feature to get a measure that better reflects the perceived quality of a movement. At layer 4, machine learning techniques are often employed to map a point or a trajectory in a multidimensional space obtained at layer 3 onto the nonverbal communicative intention an external observer perceives. Both supervised and unsupervised approaches are adopted. Considering, for example, communication of emotions, several approaches are available in literature, ranging e.g., from clustering (Glowinski, et al., 2011) to support vector machines (Piana, Staglianò, Odone, & Camurri, 2016), to several ways of integrating and fusing different classifiers; see examples in (Kleinsmith & Bianchi-Berthouze, 2013).

Bibliography

Camurri, A., Krumhansl, C. L., Mazzarino, B., & Volpe, G. (2004). An exploratory study of anticipating human movement in dance. *the Proceedings of this same Conference.*

Glowinski, D., Dael, N., Camurri, A., Volpe, G., Mortillaro, M., & Scherer, K. (2011). Toward a minimal representation of affective gestures. *IEEE Transactions on Affective Computing, 2*, 106-118.

Glowinski, D., Gnecco, G., Piana, S., & Camurri, A. (2013). Expressive non-verbal interaction in string quartet. *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, (pp. 233-238).

Kleinsmith, A., & Bianchi-Berthouze, N. (2013). Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing, 4*, 15-33.

Piana, S., Staglianò, A., Odone, F., & Camurri, A. (2016). Adaptive body gesture representation for automatic emotion recognition. *ACM Transactions on Interactive Intelligent Systems (TiiS), 6*, 6.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology, 278*, H2039--H2049.