D2.2 – Second version of the Dance software libraries for the analysis of expression, emotion, and social signals

TABLE OF CONTENTS

Introduction

Second version of The DANCE software libraries for the analysis of expressivity, emotion, and social signals is a collection of software modules, components and applications that are integrated in the DANCE Platform, grounded on the EyesWeb XMI software. EyesWeb XMI is a modular system that supports the creation of multimodal applications: it provides software modules, called *blocks*, that can be assembled intuitively to create programs, called *patches*, that process the input data (e.g., files, webcam videos, sounds, physiological data etc.) and generate multimodal output (e.g., audio, video etc).

Details on the DANCE platform can be found in Deliverable 4.2 and the software platform is available for public free download in the DANCE website at the following address:

http://dance.dibris.unige.it/index.php/2017-02-08-13-44-31/dance-platform-v2

This document provides a description of the new set of expressive qualities implemented as software libraries for the DANCE platform (see Deliverable D2.1 for a comparison with the first version of the software library), and presents a brief overview of the implementation of the main new algorithms: i.e., patches (Section 2) and blocks (Section 3) included in this second version of the DANCE software libraries. This document summarizes the main software library modules that were carried out during the second year of DANCE. More detailed description of the methodology, and on the results of the evaluation and analysis of expressive qualities and algorithms can be found in the scientific papers listed in Section 4 and in other papers in preparation.

Details on the software implementation of this software library are available in the same website dedicated to the DANCE Platform.

This library and the DANCE platform are used and exploited in scientific experiments, in public events, and in exploitation and dissemination activities, including the project "Atlante del Gesto_Genova" by the choreographer Virgilio Sieni. In this event , about 150 registered citizen are following weekly rehearsals (from January to March 2017) with Virgilio Sieni and his assistants, supported by the DANCE platform and libraries to enhance the learning and experience of expressive movement qualities by means of automated analysis of movement qualities and their sonification. This will result in a series of public performances in different sites in Genoa on 24, 25, and 26 March 2017, a main event in the third year of DANCE.

1. Multi-Layered Computational Framework of Qualities in Movement

Within the DANCE Project we have proposed a Multi-Layered Computational Framework [1]. It consists of four 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 - frames: **physical and sensorial data**, 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 new qualities proposed in this deliverable are presented following this framework. More details about the conceptual framework is available in [1].

2. Expressive Qualities in Movement

During the second year of DANCE we continued the approach defined in the DoW to investigate movement qualities, characterized by a strict collaboration with choreographers and dancers in order in get inspiration in both identifying the vocabulary of qualities and in the definition and refinement of algorithms to their automated measurement. In particular, we collaborated with the famous choreographer Virgilio Sieni to define a set of expressive qualities that corresponds to his expressive vocabulary, and we used this know-how to validate and extend our conceptual framework. Virgilio Sieni participated in a number of multimodal recording sessions during the first and second year of DANCE project (see Deliverable 5.2 for a detailed description). We were able to collect a repository of multimodal movement fragments for a number of qualities, and we developed new algorithms, in some cases inspired by some of the features. The repository also contributed to the training of machine learning module for some of the algorithms. The overall effort resulted in the second version of the DANCE software libraries.

The qualities are organized in a hierarchical structure that follows the framework presented in Section 1 (see Figure 1). In the following subsections, we present brief descriptions of each of these qualities. A *youtube* repository is an internal reference for some of the features at layer 3. We mainly introduced new qualities at Level 2 and Level 3, scaled from IMUs to full motion capture and in some cases other channels (e.g. audio for measuring respiration). In the following, w also provide brief explanation of some of the algorithms developed to compute the expressive qualities. In this Deliverable we do not discuss the basic features of Level 1, such as computation of the velocity of acceleration, but we mainly focus on description of Layer 3 mid-level features, which is considered the most important layer for movement quality [1]. All the implemented algorithms are integrated in the EyesWeb platform (see Deliverable D4.2 for the platform description). Figure 1 presents the list of the main expressive qualities presented in this deliverable. A full list is described in [1].

The DANCE software libraries are composed by two collections of patches[1]. The first collection uses the high-precise motion capture data and, thus, it can only be applied to the data recorded in laboratory settings (i.e., with a motion capture system such as Qualisys). These algorithms are typically used in scientific experiments on interactive sonification, including the ones planned for the third year of DANCE. Additionally, we also downscaled the algorithms to be used with the data collected with low-invasive sensors (such as Inertial Movement Units, IMUs).

[1] The patch is application created in the EyesWeb software, it is usually composed of several modules called "blocks".

These algorithms are developed to be used in ecological settings, e.g., during the real dance performance and public presentations and events planned in DANCE.

Motion captured (MoCap) data can be generated using dedicated hardware (e.g., the Qualisys motion capture system, in the case of the DANCE project). The algorithms compute movement features starting from MoCap data, i.e. 3D positions and rotations of the markers placed in the body of the dancer. For the analysis of dance movement, a set of 64 makers is used (see Deliverable D5.1 for more details on the markers configuration).

Inertial Movement Units (IMU) are devices endowed with sensors that capture, sample and transmit data such as acceleration, angular velocity and magnetic field in real-time. Depending on the expressive quality we use two different configurations of the IMU sensors. The full configuration (noted as IMU7) consists of 7 sensor: Four of them are placed on the dancer's limbs, and 3 on the back of the dancer. The second configuration (noted as IMU5) uses 4 sensors on the limbs but only one on the back. To compute the features that requires the body orientation (e.g., Section 2.1.5, 2.1.6, 2.1.7) the (manual) calibration of the system is required at the begging of data capturing.

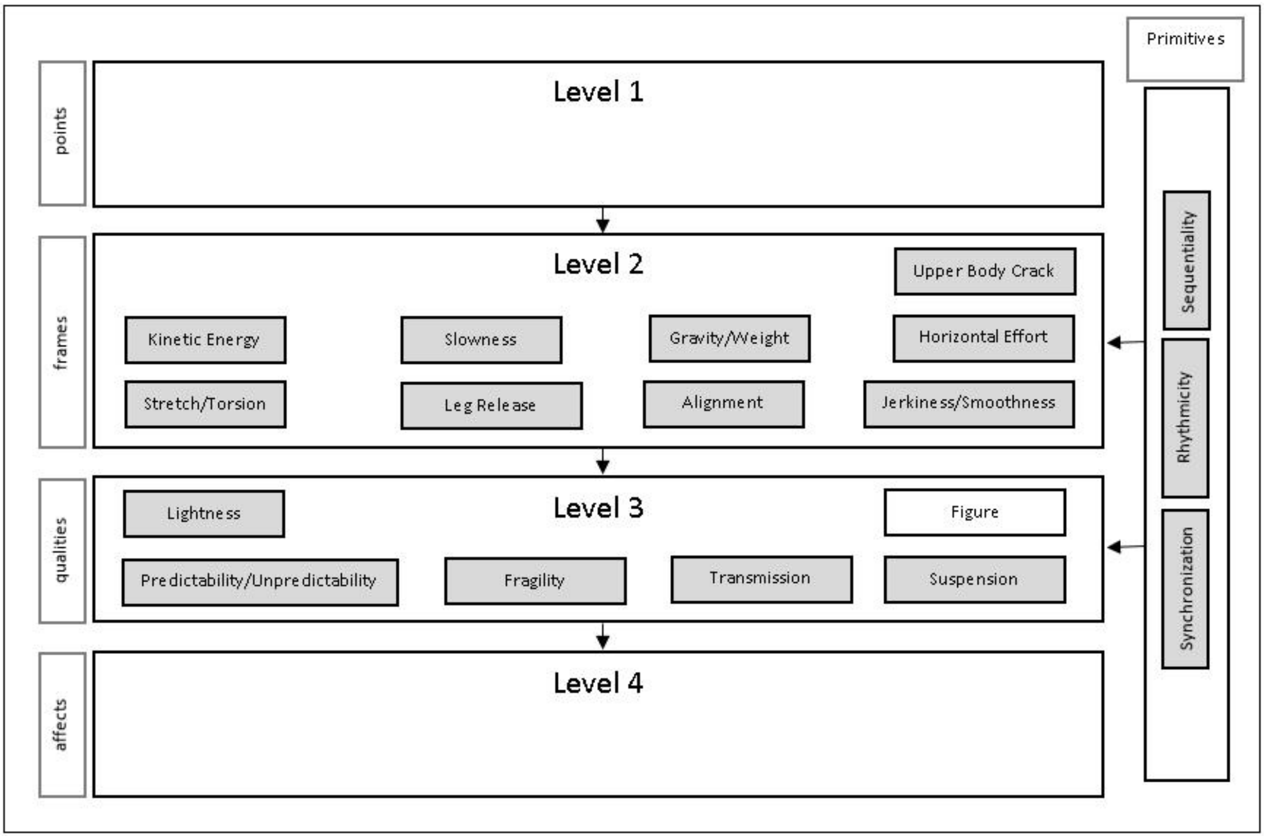


Figure 1. Multi-Layered Computational Framework of Qualities in Movement[2]

Usually the proposed features are normalized or rescaled to assume values ranging from 0 to 1 for practical reasons: to have standard values to be used in the sonifications and subsequent feature computations algorithms.

The EyesWeb software and DANCE software libraries can be downloaded from:

ftp://ftp.infomus.org/Evaluate/EyesWeb/XMI/Version_5.7.x/EyesWeb_XMI_setup_5.7.0.0.exe

The documentation of a patch that extracts the expressive qualities from an example segment of recordings can be found on the DANCE Project website:

http://dance.dibris.unige.it/index.php/2017-02-08-13-44-31/dance-platform-v2#qualityextraction

[2] The expressive qualities implemented in EyesWeb during the 2nd year of DANCE are signed in grey in Figure 1.

2.1. Level 2

The second layer of the Framework computes a low-level motion features at a small time scale (i.e., observable frame-by-frame) such as kinetic energy.

2.1.1. Kinetic Energy (Energia)[3]

Definition

The quantity of the energy that the human (i.e., dancer) possesses due to its motion. It can be computed for the whole body or some parts (e.g., hands).

Implementation (Mocap)

Kinetic energy is computed as the mass times the squared speed (tangential velocity) of joints. The velocity is obtained at the Level 1 of the framework by computing the 1'st derivative of the joints' position. Kinetic energy can be computed for one or more joints. In the latter case, it is a sum of the energy of single joints. The value is then normalized using a statistical approach to have an index ranging from 0 to 1.

Implementation (IMU5, IMU7)

It is computed following as the mass times the velocity. The velocity is obtained at the Level 1 by computing the integration of the joints' tangential acceleration. Kinetic energy is can be computed for one or more joints. In the latter case, it is a sum of the energy of single joints. The value is then normalized using a statistical approach to have an index ranging from 0 to 1.

2.1.2 Gravity/Weight (Gravità/Peso)

Definition

The weight is the quantity of energy spent by a performer in the downward direction (towards the floor or following the force of gravity) compared to the total amount of energy of the movement, it is thus a ratio between the vertical displacement and the overall one.

Implementation (Mocap)

to be implemented during 3rd year.

Implementation (IMU5, IMU7)

It is computed as a ratio between the vertical component of kinetic energy and the total (all axes) energy for each sensor separately.

2.1.3 Horizontal Effort (Piani orizzontali)

Definition

This feature is complementary to weight: it is the quantity of energy spent by a performer in the horizontal plane (i.e., in a perpendicular direction with respect to the force of gravity) compared to the total amount of energy spent in the movement, it is thus a ratio between the horizontal displacement and the overall one.

Implementation (Mocap)

to be implemented during 3rd year.

---

[3] All the terms were originally introduced in Italian and then translated to English. In parenthesis, we provide the original name in Italian.

Implementation (IMU5, IMU7)

It is computed as a ratio between the horizontal components of kinetic energy and the total (all axes) energy for each sensor separately.

2.1.4 Upper Body Crack (Incrinatura)

Definition

It is a single synchronized discontinuity of the upper body movement followed by the movement re-planning.

Implementation (Mocap)

To model movement re-planning our algorithm looks for direction changes. For this purpose, it uses the accelerations and velocities of the head and wrists' joints computed previously at the Level 1 from the 3D position data. Next, the algorithm detects: (1) abrupt variations of the absolute value of the head and wrists' accelerations and (2) zero crossings of the head and wrists' velocity. If (1) and (2) occur synchronously (synchronization measured with MECS, see Section 3.2) on the three joints, an upper body crack is detected. The window, on which synchronization between (1) and (2) is measured, is set to 0.3 seconds. The algorithm exploits an envelope extraction on 5 frames, so it is almost instantaneous with 0.05 seconds delay computed on data sampled at 100 fps.

Implementation (IMU5, IMU7)

Upper Body Crack computation is approximated by the following heuristics: we consider the start and end instants of small movements of both hands; then we measure the synchronization between start instants of the hands, and the synchronization of end instants of the hands. If synchronization emerges, and such events do not exhibit regular or periodic patterns, then we consider these as an approximation of *Upper body crack* (see Figure 2). The algorithm uses as input data the 3-axis linear acceleration from two accelerometers placed on the wrists of the dancer (see Figure 2, Phase A). First, for each hand, the algorithm detects the movement start and stop by applying thresholds (see Figure 2, th_1, th_2) on the acceleration value. Start and stop events define four time-series (right hand-start events ts, right hand-stop events ts, left-hand start events ts, left-hand stop events ts) (See Phase B, on Figure 2). These 4 time-series are provided as input to the two instances of our synchronization algorithm (Multi-Event Class Synchronization) (Phase C). In the last step (Phase D), the regularity of synchronization between hands using the Binary Pulse Regularity algorithm is measured. Only if both start events (i.e., the movement beginning of two hands) and stop events (i.e., the movement endings of two hands) are synchronized, and there is no regularity, an *Upper body crack* is detected (the algorithm outputs 1; otherwise the output is 0).

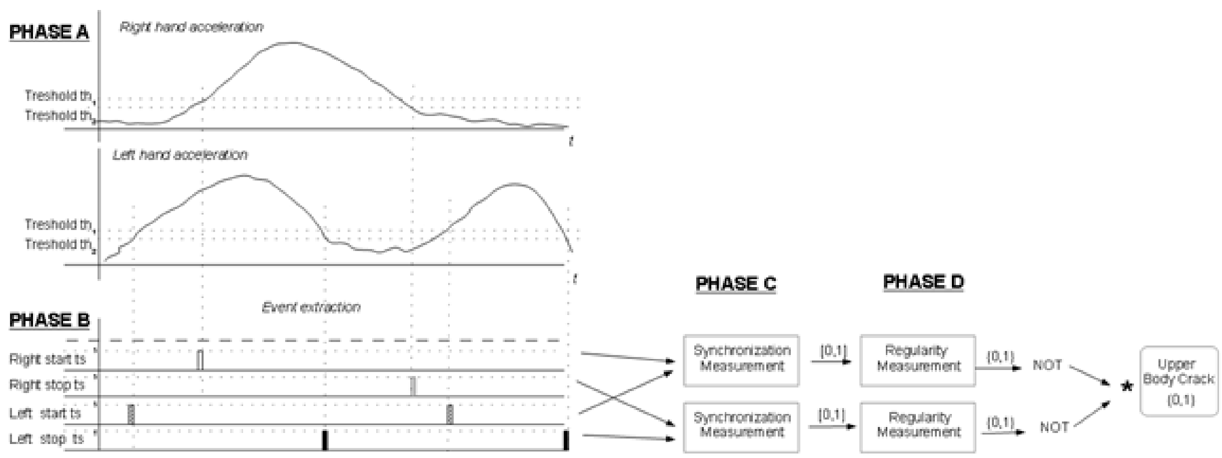


Figure 2. Upper body crack detection algorithm.

2.1.5. Leg Release / Collapse (Cedimento)

Definition

We refer to Leg Release as a sudden movement of the hips (and knees) toward the floor due to a sudden synchronized release of knees.

Implementation (Mocap)

It measures the synchronization of kinetic energy for vertical movements. In detail, a leg release is detected when abrupt changes of the vertical energy (computed on the vertical component of the hips' acceleration) occurred synchronously. Synchronization on both hips is evaluated with tolerance of 0.3 seconds.

Implementation (IMU7)

It measures synchronization of vertical movements between three different body plans. In more details, the algorithm uses the data of three accelerometers placed vertically on the back along the spine of the dancer. For each the sensors, the algorithm detects the start and the stop of the movement by applying a double threshold on the vertical component of the acceleration.

Starts and stops events are given as input to the Multi-Event-Class Synchronization algorithm (see Section 3.2), which computes the total degree of the synchronization of the plans. If the events are synchronized, then leg release will be detected.

2.1.6 Stretch and Torsion

Definition

Two quantities describing the degree of body stretch and torsion of the spine as projected on the sagittal and horizontal body planes.

Implementation (Mocap)

It is computed as the angle differences between 5 body planes: head, shoulder, torso, waist and knees. Stretch and Torsion are the normalized differences between angles of the planes projected on the sagittal and horizontal plane.

Implementation (IMU7)

It is computed as the angle differences between 3 body planes. In more details, the algorithms use the data of three accelerometers placed vertically on the back along the spine of the dancer. Stretch and Torsion are the values of normalized difference between angles projected on the sagittal and horizontal body planes.

2.1.7 Alignment

Definition

Degree of overall alignment of the body planes.

Implementation (Mocap)

It is computed as the inverse of the mean value between Torsion and Stretch, version MoCap (see Section 2.1.6).

Implementation (IMU7)

It is computed as the inverse of the mean value between Torsion and Stretch, version IMU7 (see Section 2.1.6).

2.1.8 Jerkiness/Smoothness

Definition

Jerkiness is the quantity of the discontinuity in the movement trajectory, smoothness is the inverse of jerkiness.

Implementation (Mocap)

It is computed as the 3rd derivative and the inverse of the 3rd derivative of the position.

Implementation (IMU5, IMU7)

It is computed as the 1st derivative and the inverse of the 1st derivative of the absolute acceleration.

2.1.9 Slowness

Definition

Slowness describes very slow, almost imperceptible but constant movements, related to very low, constant values of kinetic energy.

Implementation (IMU5, IMU7) and (Mocap)

It checks, for each joint or sensor whether its energy is low. In more details, using a sigmoid threshold, it verifies if the kinetic energy is in a rang (0, k] (strictly greater than zero) where k is a "very small" value.

The Table 1 shows the summary of all Level 2 qualities that were implemented during the second year of the DANCE Project (please check Deliverable D2.1 to see the algorithms of qualities previously implemented in DANCE).

Table 1. Summary of the new qualities at Level 2.

Quality	Version: IMU			Version: MoCap		
	Input	Output		Input	Output	
		Type	Values		Type	Values
Kinetic Energy	3D data from one or more IMU sensors	Local (separate for each part of the body or Global (one for the whole body)	real value from [0, 1]	3D position of one or more joints	Local (separate for each part of the body or Global (one for the whole body)	real value from [0, 1]
Gravity	3D data from one IMU sensor	Local	real value from [0, 1]			
Horizontal Planes	3D data from one IMU sensor	Local	real value from [0, 1]			

Upper Body Crack	3D data from two IMU sensors placed on hands	Global	integer value from {0, 1}	3D positions of two wrist and head	Global	integer value from {0, 1}
Leg Release	3D data from three IMU sensors placed on back	Global	integer value from {0, 1}	3D positions of hips	Global	integer value from {0, 1}
Torsion	3D data from three IMU sensors placed on back	Global	real value from [0, 1]	3D positions of all joints	Global	real value from [0, 1]
Stretch	3D data from three IMU sensors placed on back	Global	real value from [-1, 1]	3D positions of all joints	Global	real value from [-1, 1]
Alignment	3D data from three IMU sensors placed on back	Global	real value from [0, 1]	3D positions of all joints	Global	real value from [0, 1]
Jerkiness/ Smoothness	3D data from one IMU sensor	Local	positive real value	3D position of one joint	Local	positive real value
Slowness	3D data from one IMU sensor	Local	real value from [0, 1]	3D position of one joint	Local	real value from [0, 1]

2.2 Level 3

The third Layer of the Framework (see Section 1) segments the flow of movements in a series of single units (or gestures) and computes a set of mid-level features, i.e., complex features that are usually extracted on groups of joints or the whole body, and require significantly longer temporal intervals to be observed (i.e., between 0.5s and 5s).

2.2.1 Lightness (Leggerezza)

Definition

A necessary condition for a Light movement is Fluidity. Further, a fluid movement should include at least one of the following characteristics: (i) to exhibit a low amount of downward vertical acceleration following gravity (in particular on forearms and knees), (ii) each possible downward acceleration must be counterbalanced by an opposite "harmonic" upward movement (simultaneous or consequent); (iii) vertical downward acceleration movements are turned into a finalization on the horizontal plane.

Example

MoCap visualization of a dancer performing lightness can be seen at: https://youtu.be/OECmF14_dKA

Implementation (Mocap)

It is computed as the ratio between the full body vertical kinetic energy directed "toward the air" and the total kinetic energy. In more details, first we compute the sum vertical kinetic energy of the all body joints moving "up". Then we divide the "up" energy, by the total kinetic energy of all the joints.

Implementation (IMU5, IMU7)

It is extracted as opposite to mean value of Weight computed on n sensors (see Section 2.1.2 for the Weight definition). Its value is in the interval [0, 1].

$$L = \frac{\sum_{i=1}^{n} W_i}{n}$$

2.2.2 Fragility (Fragilita)

Definition

A sequence of non-rhythmical upper body cracks and leg releases (see Section 2.1.4 and 2.1.5). It emerges, for example, when moving at the boundary between balance and fall, resulting in short movements with continuous interruption of motor plans. The resulting movement is non-predictable, interrupted, and uncertain.

Example

MoCap visualization of a dancer performing fragility can be seen at: https://youtu.be/jDqJa__7tD8

Implementation (Mocap)

It detects non-regular sequence of upper body cracks or leg releases in 1 a second time window. Upper body cracks and leg releases are extracted at Level 2 of the framework (see Sections 2.1.4 and 2.1.5), and are accumulated in a 1 second buffer on which the detection of non-rhythmical patterns is performed. To check non-rhythmicality we measure the inter-event time durations and we check if these intervals are regular or not (see Section 3.3 for details on algorithm).

Implementation (IMU5, IMU7)

It detects non-regular sequence of upper body cracks or leg releases in 1 a second time window. It uses Binary Pulse Regularity algorithm (see Section 3.3) on the top of upper body cracks or leg releases algorithms. If at least 3 upper body cracks or leg releases appear within one second with a constant time interval then algorithm outputs 0 (=no fragility), otherwise it outputs 1 (it means contribute to fragility).

2.2.3 Transmission (Trasmissione)

Definition

The transmission refers to circulation of body kinetic energy between different body planes. It may include resonances between different body parts, or changes of the body part that is leading movement.

Example

MoCap visualization of a dancer performing transmission can be seen at: https://youtu.be/8CmE7m4HTLs

Implementation (Mocap)

It detects a sequential transfer of the energy between the consecutive body plans. In more details, 5 body planes are considered: head, shoulder, torso, waist and knees. For each body plane the rotational and translational kinetic energy is computed and summed up. Next, the algorithm finds the maximum of the energy between the five plans. If the maximum of the energy "moves" between adjacent planes creating the sequence, transmission is detected. Whether the energy transmission sequence is created or not, is measured using the Sequence Detector primitive (see Section 3.4 for details).

Implementation (IMU)

It is computed as in the MoCap case but with less planes (3 instead of 5).

2.2.4 Suspension (Sospensione)

Definition

Suspension refers to non-directional holding of energy on one of the body planes. The body or some parts of it may, for example, waving or rippling. The movements of suspension are often highly predictable and repetitive.

Example

MoCap visualization of a dancer performing suspension can be seen at: https://youtu.be/QP7sWRMc3n8

Implementation (Mocap)

It detects whether the maximum of the energy is retained over a period of time on one body plan. In more details, 5 body planes are considered: head, shoulder, torso, waist and knees. For each body plane the rotational and translational kinetic energy is computed, summed up and normalized by the total amount of energy of the body (sum of the energy of all planes). The suspension is detected if the one plane has a greater energy respect to the all other ones for at least 0.5 seconds.

Implementation (IMU7)

It detects whether the maximum of the energy is retained over a period of time on one body plan of three body planes: head, shoulder, torso, and waist. It works similarly to the MoCap version of the algorithm.

2.2.5 Figure

Definition

Articulation of alignments and unisons, consolidation of balance.

Example

MoCap visualization of a dancer performing figure can be seen at: https://youtu.be/pK6SrYq2Ooc

Implementation (IMU) - to be implemented during 3rd year.

Implementation (Mocap) - to be implemented during 3rd year.

2.2.6 Predictability

Definition

Sequence of rhythmical upper body cracks and/or leg releases.

Implementation (Mocap)

It detects a sequence of at least three upper body cracks or leg releases evenly distributed in one second time window. Upper body cracks and leg releases are detected at Level 2 of the framework (see Sections 2.1.4 and 2.1.5), and are accumulated in a 1 second buffer. Next, Binary Pulse Regularity algorithm is applied to check the event rhythmicality by checking whether inter-event time is constant over the buffer (see Section 3.3 for details on algorithm).

Implementation (IMU) - to be implemented during 3rd year.

2.2.7 Unpredictability

Definition

Single upper body cracks or leg releases in a certain time period. The eventual repetitions of upper body cracks make the movement predicable (see Section 2.2.6).

Implementation (Mocap)

The algorithm detects a unique and unrepeated upper body crack or leg release in 1 second time window. Upper body cracks and leg releases are detracted at Level 2 of the framework, and are accumulated in a 1 second buffer on which the detection of non-rhythmical patterns is performed. To check the uniqueness over the time buffer Binary Pulse Regularity algorithm is applied (see Section 3.3 for details on algorithm).

Implementation IMU - to be implemented during 3rd year.

The Table 2 shows all Level 3 qualities that were implemented during the second year of the DANCE Project (please check Deliverable D2.1 to see the algorithms of qualities previously implemented in DANCE).

Table 2. Summary of the new qualities at Level 3.

Quality	Version: IMU		Version: MoCap	
	Input	Output	Input	Output
Lightness	Gravity computed for each sensor: values from [0, 1]	real value from [0, 1]	Gravity computed for each joint: values from [0, 1]	real value from [0, 1]
Fragility	Events of Upper Body Crack or Release: discrete values from {0,1}	discrete value from {0, 1}	Events of Upper Body Crack or Release: discrete values from {0,1}	discrete value from {0, 1}
Transmission	Energy computed for each sensor: values from [0, 1]	discrete value from {0, 1}	Energy computed for each joint: values from [0, 1]	discrete value from {0, 1}
Suspension	Energy computed for each sensor: values from [0, 1]	discrete value from {0, 1}	Energy computed for each joint: values from [0, 1]	discrete value from {0, 1}
Predictability			events of Upper Body Crack or Release: discrete values from {0,1}	discrete value from {0, 1}
Unpredictability			events of Upper Body Crack or Release: discrete values from {0,1}	discrete value from {0, 1}

3. Analysis primitives

In this Section we present new basic algorithms that are used to compute several features of Level 2 and Level 3. Most of them were coded in a form of “blocks”, i.e., reusable modules for EyesWeb Platform (see Deliverable 4.2 for details on platform).

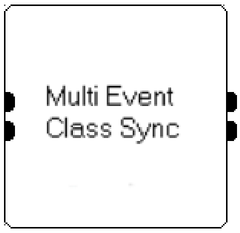
3.1 Saliency

We introduce the algorithm to model any expressive quality that lasts for a long period of time in time.

Implementation:

- We compute the histogram (between 0 and 1, bins size 0.1) of the input feature over a long (1 second) and a short (0.1 second) time span.
- We extract the maximum of each histogram and we compute their normalized (in [0, 1]) distance D1: if the "recent" values of the input feature are very different from the "less recent" ones then the histograms' peaks can be more "far".
- To deal with cases in which they are not far, we also compute the normalized (in [0, 1]) difference D2 between the maximum value of the "recent" histogram and the corresponding (i.e., in the same position, not necessarily the max) value of the "less recent" histogram.
- We compute normalized (in [0, 1]) Inhibition of Saliency I as the leaky integral of D1*D2.
- Saliency = (1-I)*D1*D2

3.2 Synchronization



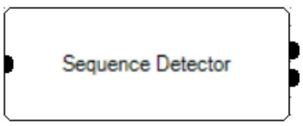
We propose a new algorithm named Multi-Event-Class Synchronization (MECS). MECS computes the degree of synchronization between several time-series containing several types of events. Since events are no longer of the same typology, they belong to different *classes of events.*

The algorithm evaluates a single synchronization index related to each class separately. It relies its computations on the temporal distances between the timings at which events belonging to the different classes appear. Moreover, MECS allows establishing and computing the synchronization degree between *sequences of events* i.e. macro-events defined by specific time constraints between original events found in the time-series.

3.3 Regularity

This block implements the Binary Pulse Regularity algorithm. It works on a continuous stream of binary (i.e., 0 and 1) values. The inter-event interval, i.e. the distance between two consecutive synchronization events (values 1 in the time series), are used to measure regularity: three or more events are considered *regular* if the corresponding inter-event intervals differ for a number of samples lower than a given threshold value. By increasing the threshold, we introduce tolerance in the regularity estimation.

3.4 Sequentiality



This block works, as the previous one, on a continuous stream of positive integer values, in which a zero means that there is no event, while a non-zero identifies an event of a particular class (i.e., each positive integer number corresponds to a class). In the block's parameter we can specify which is the sequence of events we want to be detected by the block. For example, we may want to detect the sequence 1 1 2 2 3 4. Another parameter specifies the maximum distance between two events that is allowed in detecting the sequence. In the previous example, if the parameter is set to 2, then the sequence is detected in the stream of values 1 0 1 2 0 0 2 0 0 3 0 4 and it is not detected in the stream 1 1 2 0 0 0 3 4. Finally, there is another parameter that specifies whether the sequence has to be detected in the specified order or bidirectionally.

4. References

1. Camurri, A., Volpe, G., Piana, S., Mancini, M., Niewiadomski, R., Ferrari, N., Canepa, C., The Dancer in the Eye: Towards a Multi-Layered Computational Framework of Qualities in Movement, 3rd International Symposium on Movement and Computing, MOCO 2016.
DOI: http://dx.doi.org/10.1145/2948910.2948927.

In this paper we introduce a multilayered framework for the analysis of expressive qualities of movement. The framework is composed of four levels and it provides a conceptual background to develop computational models of expressive qualities.

2. Alborno, P., Piana, S., Mancini, M., Niewiadomski, R., Volpe, G., Camurri, A., Analysis of Intrapersonal Synchronization in Full-Body Movements Displaying Different Expressive Qualities, in Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI 2016), Paolo Buono, Rosa Lanzilotti, and Maristella Matera Eds., ACM, n. 136, pp. 143, New York, USA, 2016

In this paper we present a study of intra-personal synchronization between limb movements in different expressive qualities. Intra-body synchronization (or lack of synchronization) is an important attribute contributing to explaining several expressive qualities e.g., Fragility (see Section 2.1.4). The results of this study were used to develop some of the algorithms presented in Section 2.

3. Piana, S., Coletta, P., Ghisio, S., Niewiadomski, R., Mancini, M., Sagoleo, R., Volpe, G., Camurri, A., Towards a Multimodal Repository of Expressive Movement Qualities in Dance, 3rd International Symposium on Movement and Computing, MOCO 2016, 5-6 July 2016, Thessaloniki, Greece

In this paper we present the multimodal repository for the analysis of expressive movement qualities. Further, the methodology we applied to create this repository is discussed.

4. Kolykhalova, K., Alborno, P., Camurri, A., Volpe, G., A serious games platform for validating sonification of human full-body movement qualities, 3rd International Symposium on Movement and Computing, MOCO 2016, 5-6 July 2016, Thessaloniki, Greece.

In this paper we propose a serious game platform for the evaluation, comparison, and validation of movement qualities and of their sonifications.

5. Piana, S., Alborno, P., Niewiadomski, R., Mancini, M., Volpe, G., Camurri, A., Movement Fluidity Analysis Based on Performance and Perception in Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI 2016), ACM, n. 1629, pp. 1636, New York, USA, 2016. doi: 10.1145/2851581.2892478

In this paper we propose a methodology for studying expressive qualities in dance. An example of such analysis is also presented, consisting of a preliminary study of fluid movements. Fluidity is an important component of several expressive qualities discussed in this document e.g., Lightness (see Section 2.2.1.).

6. Camurri, A., Cera, A., Piana, S., Canepa, C:, Alborno, P., Volpe, G., Kolykhalova, K., Niewiadomski, R., Mancini, M., Interactive sonification of movement qualities - a case study of fluidity, Interactive Sonification Workshop (ISon 2016), Bielefeld, 15-16 December 2016.

In this paper we investigate how expressive qualities of human full-body movements can be expressed by sound. Using the Fluidity as leading example of movement quality, we propose (i) an algorithm to detect this quality from full-body movements, and (ii) a model of interactive sonification to convey Fluidity through the auditory channel.