D4.3 – DANCE System Platform, Final version

1. INTRODUCTION 3

2. PORTABLE DANCE PLATFORM 3

3 BODY GROUPS ORIENTATION EXTRACTION 4

3.1 Extraction from IMU 5

4 SONIFICATION PLATFORM 5

4.1 Sonification platform overview 6

4.2 Sonification of Lightness 8

4.3 Sonification of Fragility 8

5 SOCIAL SIGNAL DETECTION 8

5.1. C1: Stability 10

5.2. C2: Anticipation 11

5.3. C3: Suddenness 12

5.4. C4: Richness 13

5.5. C5: Physical Contact 13

5.6. C6: Gaze 14

5.7. Computation of Leadership 15

5.8 Evaluation 15

BIBLIOGRAPHY 16

1. Introduction

The *DANCE software platform,* based on the EyesWeb XMI research platform, allows synchronized recording, playback, and analysis of a multimodal stream of data. Its main characteristics have been illustrated in Deliverables 4.1 and 4.2. In this document we describe the improvements and extensions we introduced in the platform during the third year of DANCE. We report below the main characteristics of the platform, taken from D4.1 and D4.2:

- creation of a multimodal repository of recordings of movement qualities;
- fine-grain synchronization of multimodal data
- segmentation of the recordings in fragments, according to the chosen qualities
- playback and testing of the repository
- extraction of the movement features and qualities (for single as well as multiple users)
- real-time interactive sonification
- design and development process of scientific experiments of DANCE
- design and development of the prototypes of applications
- design and development of artistic projects exploiting the results of the DANCE project (e.g., artistic performances)

During the third year of DANCE, the main improvements to the DANCE platform concerned the following points:

- we consolidated scalability, that is, the possibility to run the platform only with mobile sensors and a portable device, see Section 2 for the details;
- we added extraction of the body planes orientation (e.g., orientation of head, shoulders, torso, hips) to the palette of expressive movement qualities available in the DANCE software libraries, see Section 3 for the details;
- we integrated the movement quality sonification modules into the DANCE platform, see Section 4 for the details;
- we added the extraction of social signals (i.e., Leadership in a duo of dancers) to the DANCE software libraries, see Section 5 for the details.

In Section 4, together with a description of the integration of the sonification modules into the DANCE platform, you will also find instructions to download, install and run a version of the DANCE software platform demonstrating how movement qualities of a dancer can be sonified in real-time

2. Portable DANCE platform

We defined a portable version of the DANCE platform, by embedding all the software modules for the extraction of the movement features and their sonification in a single portable device. Figure 1 below shows an instance of the portable architecture of the DANCE platform. A dancer is wearing a number (from 1 to 7 in the current version) mobile sensors (e.g., x-OSC) on relevant joints (wrists, ankles, backbone) that stream data over a wi-fi connection set up on the mobile device (e.g., using the hotspot function available on Android devices). The captured data consists of: linear acceleration (acc), 3D orientation (mag), and rotational acceleration (gyro). The mobile device runs a down-scaled version of the DANCE platform, consisting of a reduced set of algorithms for computing low and mid-level movement qualities, like, for example, Smoothness, Kinetic Energy, Lightness, and Fragility. The down-scaled platform is also endowed with a mobile version of the sonification modules, that are executed locally, and the audio output is produced by the device or bluetooth speakers.

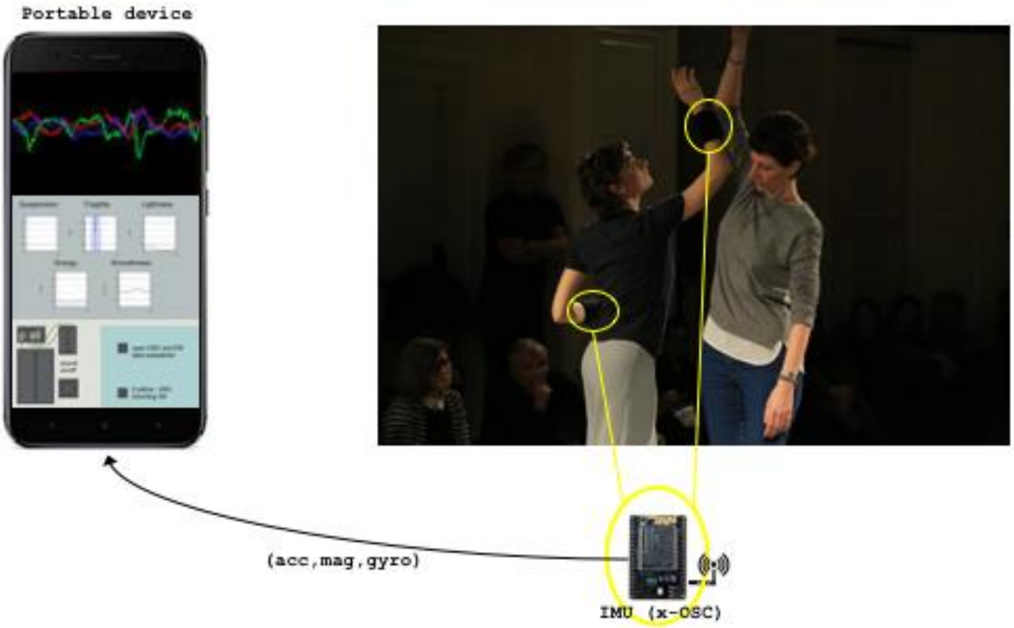

Figure 1. The portable version of the DANCE platform. Movement data (i.e., linear acceleration, 3D orientation and rotational acceleration) is captured in this case by two wireless sensors on the dancer's wrists. The sensors are connected to the local wi-fi network hosted by a portable device (e.g., a smartphone) that runs a reduced portable version of the DANCE platform. The mobile device computes a set of low and mid-level movement qualities (e.g., Smoothness, Kinetic Energy, Lightness, Fragility) and produces the corresponding sonification, sent to a bloutooth loudspeaker.

The current prototype of the portable DANCE consists of a proof-of-concept, tested for example and the DG Connect event of the dinner at La Lanterna in Rome in occasion of the celebrations of the Treaty of Rome, for the demo/performance "Europa: Gestures of History, Dancing Science and Art not to forget EU identity". In this event, the dancer weared two mobile sensors (on both wrists) and a laptop was used to run the application based on the portable version of the platform and software modules.

3 Body groups orientation extraction

During the third year of the DANCE Project, we defined algorithms to extract the 3D orientation of the dancer's body groups: head, shoulders, waist, legs, feet. We consider each group as a separate body and we compute its 3D orientation, by considering the standard T pose (the user is standing still, with arms raised laterally at the shoulder level) as the one with groups oriented in parallel frontal directions. By comparing the convergence/divergence of the body groups we can detect whether, for example, the body posture is collapsed or extended, and if there is an horizontal-lateral torsion.

We compute a vector describing the angles between adjacent body groups: head, shoulders, waist. Such angles could be the expression of postural tension (see D5.1) and Stretch/Torsion (see D2.2), as demonstrated, for example, in paintings or sculptures, e.g., the Discobolus, where such angles express the posture dynamics in terms of energy loaded in the transversal muscles ready to be released into movement.

Figure 3 depicts 3 examples of body groups orientation extraction during a dance performance.

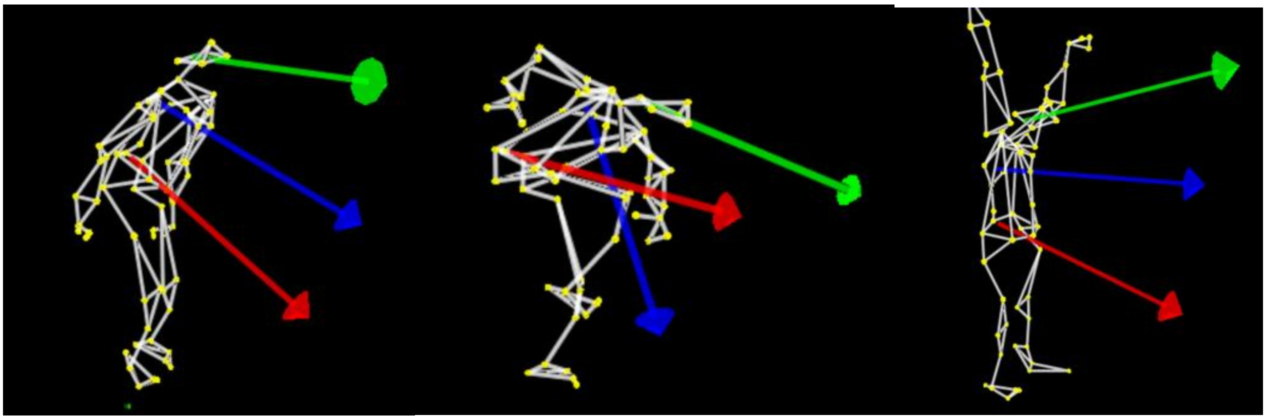

Figure 3. Body groups orientation extraction.

3.1 Extraction from IMU

The DANCE platform includes software modules to extract the 3D orientation (e.g., yaw, pitch and roll) of mobile wearable sensors endowed with 3D accelerometer, gyroscope and compass. The low-level algorithm has been ported from the project described in [Madgwick et al., 2011]. A demo of the software modules in action can be seen at: https://youtu.be/BXsGWoOMtmU

By attaching multiple sensors on the dancer's body, we can extract the rotation of individual body groups: head, shoulders, waist, legs. As described in D4.1 and D4.2, the DANCE platform supports input from x-OSC sensors[1]. They have been exploited in the evaluations described in D5.3, D1.3, and D1.4. For example, during the public performances "Di Fronte agli Occhi degli Altri" (Genoa, 24 March 2017) and "Esodi" (Genoa, 25 March 2017) described in D6.4, we conducted preliminary tests of the extraction of body groups orientation with dancers participating to the events.

4 Sonification platform

During the third year of the project, we created a sonification platform by integrating the sonification modules developed in Max MSP into the DANCE platform. In particular, we integrated the sonification algorithms of Fragility and Lightness that we exploited for the experiments described in D1.3 and D1.4.

The final version of the sonification platform can be downloaded from the web page:

http://dance.dibris.unige.it/index.php/2017-02-08-13-44-31/dance-platform-v3

To install the platform, please follow these steps:

- download and install the EyesWeb XMI 5.7.2.0 application (first link on the above web page)
  - run the exe file and execute the installer to install EyesWeb XMI 5.7.2.0
- download and install the movement quality extraction patch (second link on the above web page)
  - run the exe file and execute the installer; to execute it, a valid EyesWeb 5.7.2.0 installation must be already present in your system (see the previous step)
  - **the installer will create a link on your desktop named "Movement Features Extraction"**
- download and unzip the sonification patch (third link on the above web page)
  - unzip the archive in a user folder (e.g., Documents or Desktop folder), then you can delete the archive

To run the platform, please follow these steps:

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

- run the sonification patch by opening the user folder in which you have unzipped the sonification patch (e.g., inside the Documents or Desktop folder) and executing the file named "sonification_patch"
- run the movement quality extraction patch by double clicking the desktop link previously created on your desktop by the movement quality extraction patch installer, named " Movement Features Extraction"

After a few seconds, you will see the movement quality extraction patch interface, showed in Figure 4. The interfaces is provided with some pre-recorded sensor data of a dancer displaying the Light and Fragile movement qualities. By pressing the "Previous trial" and "Next trial" buttons you can select the movement quality. The "Play" button starts the playback and movement quality extraction process. The movement qualities extracted from the dancer's movement are sent to the sonification patch in real-time and the corresponding sonification is generated.

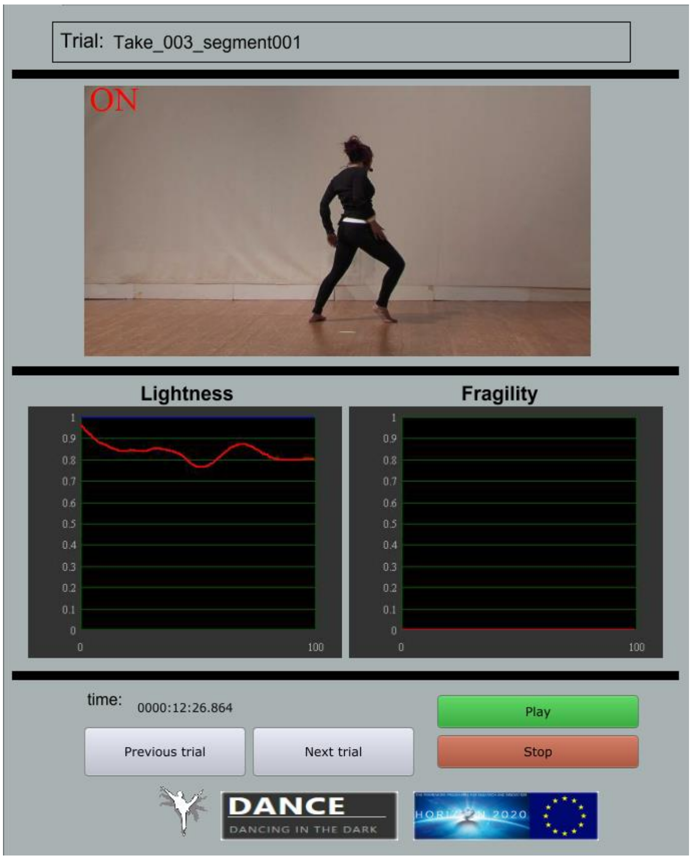

Figure 4. The movement quality extraction patch interface.

4.1 Sonification platform overview

Figure 5 illustrates the structure of the sonification platform, resulting from the integration of the DANCE platform (developed using EyesWb XMI) and Max MSP.

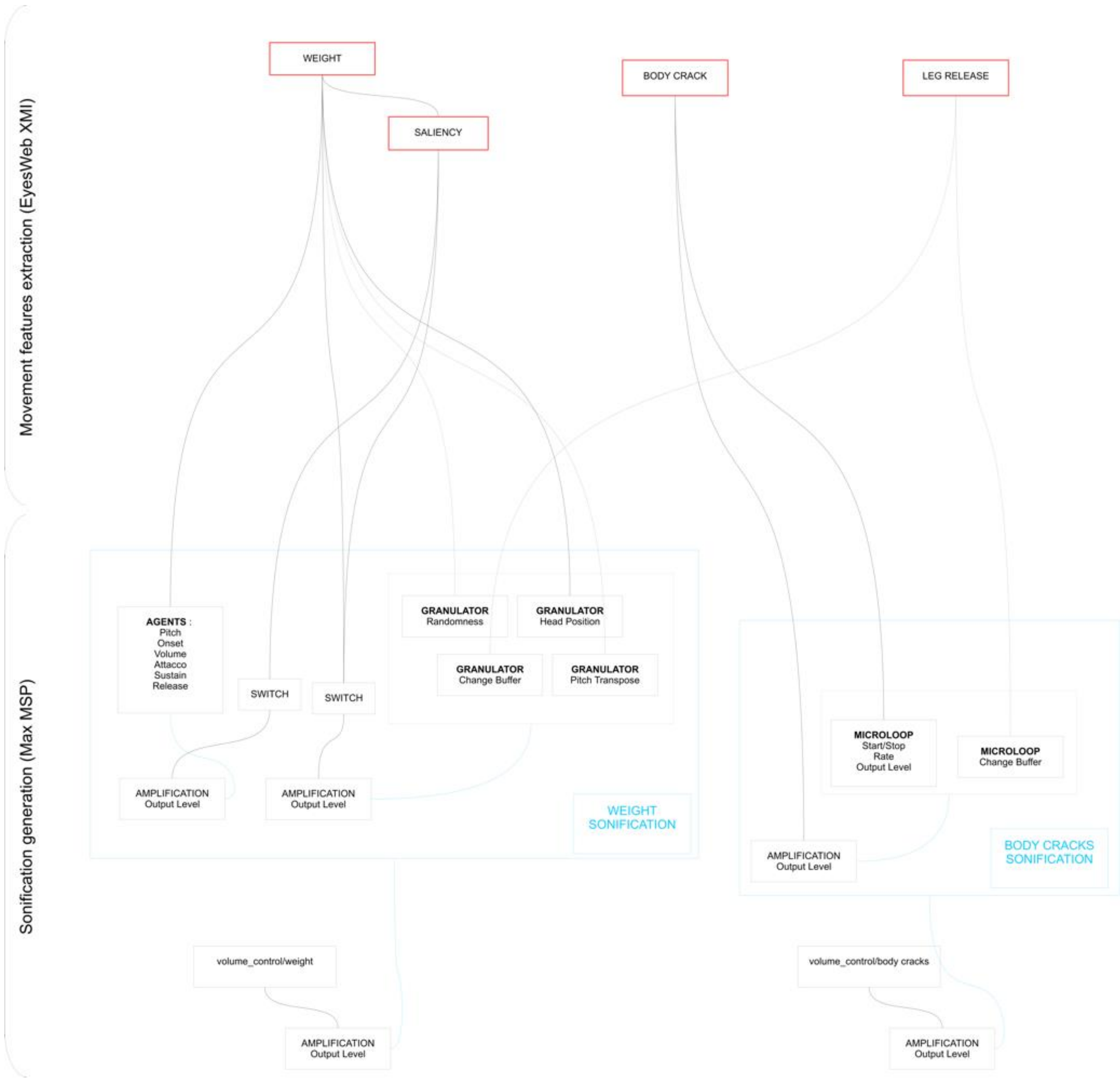

Figure 5. Sonification platform for the Lightness and Fragility movement qualities. In the upper half, movement qualities are extracted using the EyeWeb XMI application from a dancer's movement in real-time. In the lower half, the sonification patch, created using Max MSP, is provided the movement qualities as input and generates the corresponding sonification to communicate Lightness and Fragility.

In the upper part of the diagram, 4 low-level movement qualities and primitives are extracted using the EyesWeb XMI software, starting from the acceleration of the dancer's wrists:

- *Weight* - it is the quantity of energy spent by a performer in the downward direction (towards the floor or following the force of gravity); it is computed as a ratio between the vertical component of kinetic energy and the total (all axes) energy;
- *Saliency* - it models the fact that recent values of a movement feature are perceived as salient with respect to the older ones; it is computed by consider the distance between the peak P1 of the histogram of the Weight quality on a "recent" short buffer (around 1s) and the peak P2 of the histogram on a longer buffer (around 4s), the more the distance, the higher the Saliency, and vice-versa;
- *Upper body crack* - an upper body crack is detected when abrupt changes of the acceleration (e.g., the start and end instants of small movements) of the hand movements occur synchronously but non-periodically;

- *Leg release* - a leg release is detected when abrupt changes of the vertical energy (computed on the vertical component of the hips acceleration) occurred synchronously using a similar approach to the one used for Upper body cracks.

To sonify Lightness, we look for smooth movements with no "weight", i.e., which do not contain a vertical acceleration directed to the floor. Then, to compute Fragility we look for the presence of single a-rhythmic Upper body cracks and Leg releases. We provide a brief overview of the sonification algorithms in the following sections.

4.2 Sonification of Lightness

A *light* sound can be imagined as the production of soft and light elements, gently pushed away in all directions by the body movements, through an invisible medium, like air, wind, breath. We metaphorically treated a *light* body as an "instrument" that generates sound thanks to a soft interaction with other sonic elements. *Light* sounds have an inertial property: they live not only during the movement, but also after its end, they do not stop as soon as the movement stops. This inertial behavior has to be free and unpredictable. To convey this idea, we provide light sounds with three properties: i) autonomously floating elements, in a quasi-absence of gravity, ii) the impression that these elements are excited by a rush of energy (the dancer's movement), iii) a flashing counter-impression of weight, of something heavy and slow, dully impacted by the mass of a body losing Lightness.

4.3 Sonification of Fragility

A *fragile* movement is produced by an overall weak body. *Fragile* movements are fractured and incoherent. For this reason, we designed the *fragile* sonification to be multi-layered, in order to underline a multiplicity of weak, independent points.
We used four sample playback engines in a loop, playing very short, partially overlapping sounds. The loops are triggered and are audible very briefly, according to the values of Upper body crack parameter. A high number of body cracks slows down the loops' rate, allowing a better separation of the individual sounds, while a low number of body cracks accelerates the loops' rate so much that they become almost a continuous sound.

The nature of the looped sounds is of paramount importance to communicate Fragility. We recorded, selected and isolated various sound of breaking objects: light metal objects, dry leaves, branches, wood sticks. A sample may have significant events (e.g., loud cracks, which last between 50 and 100 ms) in certain points and less important content (e.g., small cracklings interwoven with silence) in the other points. At each trigger, the playback engine randomly selects portions of the sound (between 100 and 200 ms) that will be played back and used in the sonification. The Upper body crack parameter control a final, subtle variation in the amplitude of each repetition of a sound, to assure a further level of instability.

5 Social signal detection

We propose a model for computing nonverbal leadership in full-body unstructured complex physical activities, e.g., a dance performance of a duo. The nonverbal behaviors of the leader in such activities go beyond simple movement anticipation of the leader by the followers. Thus, in our model we consider a large spectrum of nonverbal behaviors that might be considered by human observers to determine who is the leader in a group. We illustrate algorithms to compute the cues of nonverbal leadership.

We choose to study contact dance improvisations since they are an unstructured physical activity, where there is no predefined leader. In particular we asked a duo of dancers to mirror continuously movements of each other; thus, in a consequence, both participants initialize physical actions (or gestures) alternately and imitate each other. This is a perfect scenario to study the nonverbal leadership. Additionally, in order to collect a possible large set of nonverbal cues (that goes beyond the simple movement anticipation) we introduce to the data collection procedure the variety of the conditions which includes the sensory deprivation in the performers (e.g., lack of vision, lack of physical contact). In this way we could collect and record the evidences of different strategies that humans use to communicate their movements or future movement intentions to the other.

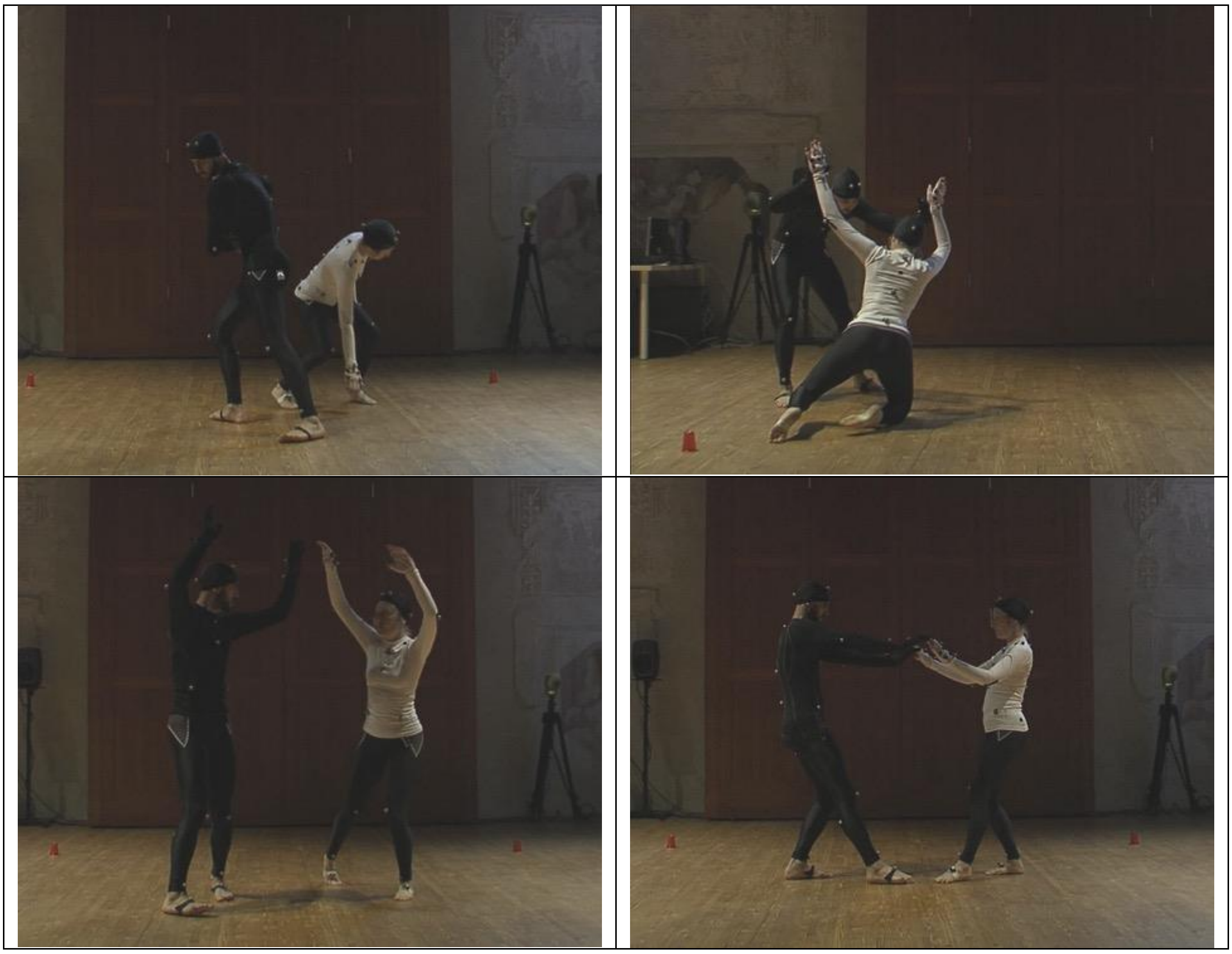

Figure 2. One couple of dancers in four different conditions: (top-left) no eye contact and no touch contact; (top-right) no eye contact and only touch; (bottom-left) only eye contact and no touch contact; (bottom-right) both eye contact and touch contact.

Below, we present the algorithms to compute 6 cues C1 - C6. In our approach We assume that different cues may have different importance (weights). Thus we introduce the importance parameter *Imp*, which is a positive long integer value expressing the contribution level of the given cue in the computation of Leadership. The lower the value (greater than 0) the higher the contribution of the cue, e.g., 3 means that it contributes in a small way, 2 in medium way and 1 in a great way. This value can be manually setup or can be the result of the annotation process.

We now describe 3 concepts in order to simplify the notation of the following sections.

Groups of joints

First, the algorithms presented below can be computed on groups of joints: left + right wrist, left + right ankle, or wrists + ankles (full-body). For the sake of simplicity, we present each of the algorithms below for a generic group J, where i can vary as it follows:

- J=w for the group "left + right wrist"
- J=a for the group "left + right ankle"
- J=fb for the group "wrists + ankles"

Absolute speed and curvature

On a single joint's 3D trajectory, we compute the absolute speed s, as it follows:

Absolute tangential speed s of a single joint 3D trajectory defined in a parametric way, such as:

$$j(t) = (x(t), y(t), z(t))$$

is computed as it follows:

$$s(t) = \sqrt{x'^2 + y'^2 + z'^2}$$

Similarly, 3D curvature k is computed as it follows:

$$\kappa = \frac{\sqrt{(z''y' - y''z')^2 + (x''z' - z''x')^2 + (y''x' - x''y')^2}}{(x'^2 + y'^2 + z'^2)^{\frac{3}{2}}},$$

Absolute tangential speed and curvature of a group of joint J are defined as the mean absolute tangential speed and curvature of the single joints included in J.

Auxiliary function G

We finally define the auxiliary function $G_k(m,n)$, corresponding to the amount of frames in a time interval in which the discrete variable m is greater than the discrete variable n increased by a constant k. We set k=0.5 for all the algorithms presented below, and the definition of the function becomes:

$$G_{0.5}(m, n) = number\ of\ frames \mid (m - n) > 0.5$$

In other words, the above function quantifies how long the value of a variable is "very much higher" than the value of another variable.

5.1. C1: Stability

Stability is extracted by computing the following 3 low-level features on the dancers' wrists, ankles and whole body (i.e., wrists and ankles):

- Variance
- Smoothness
- Variability

Variance (Var)

It is the statistical function computed by the following equation:

$$Var(s_J) = \sum_{h=1}^{M} (s_{J,h} - \mu)^2$$

The variable s_J is, in our case, the absolute tangential speed of a group of joints J on the buffer 1,...,M.

Smoothness

It is the Smoothness Index (SmI), described in [Mazzarino & Mancini, 2009] as a simplified version of the Pearson correlation of trajectory curvature and absolute tangential speed:

$$SmI(s) = \frac{1}{\kappa s}$$

where s and k are the absolute tangential speed and curvature defined at the beginning of this section.

Variability

It is computed as the difference between the actual joint velocity and the low-pass filtered velocity of the same joint. If a joint is moving in with high variability, then the low-pass filtered version of the joint velocity will be very different from the unfiltered one (which contains highly variable, i.e., high frequency, components). Instead, a less variable joint velocity (e.g., a joint moving almost constantly) will be almost identical to the corresponding low-filtered version.

We compute variability on the absolute tangential speed s of joints and groups of joint as it follows:

$$V(s) = |s - LPF(s)|$$

Computation of Stability

Given 2 dancers A and B and a group of joints J, we compute the above 3 low-level features, obtaining 18 values. We also apply the G_k function described above to the resulting values, that is:

$$G_{0.5}(Var(s_J^A),Var(s_J^B)) = GVar_J^A = number\ of\ frames\ |\ (Var(s_J^A) - Var(s_J^B)) > 0.5$$

$$G_{0.5}(Var(s_J^B),Var(s_J^A)) = GVar_J^B = number\ of\ frames\ |\ (Var(s_J^B) - Var(s_J^A)) > 0.5$$

$$G_{0.5}(SmI(s_J^A),SmI(s_J^B)) = GSmi_J^A = number\ of\ frames\ |\ (SmI(s_J^A) - SmI(s_J^B)) > 0.5$$

$$G_{0.5}(SmI(s_J^B),SmI(s_J^A)) = GSmi_J^B = number\ of\ frames\ |\ (SmI(s_J^B) - SmI(s_J^A)) > 0.5$$

$$G_{0.5}(V(s_J^A),V(s_J^B)) = GV_J^A = number\ of\ frames\ |\ (V(s_J^A) - V(s_J^B)) > 0.5$$

$$G_{0.5}(V(s_J^B),V(s_J^A)) = GV_J^B = number\ of\ frames\ |\ (V(s_J^B) - V(s_J^A)) > 0.5$$

Next, we apply the following algorithm to compute the value of Stability (C1) of Dancer A (StabA) and stability of Dancer B (StabB):

if$((GVar_J^B - GVar_J^A) > 0)$	if$((GSmi_J^B - GSmi_J^A) > 0)$	if$((GV_J^B - GV_J^A) > 0)$
$A_J^1 = 1/Imp_{C1}, B_J^1 = 0$	$A_J^2 = 1/Imp_{C1}, B_J^2 = 0$	$A_J^3 = 1/Imp_{C1}, B_J^3 = 0$
else	else	else
$B_J^1 = 1/Imp_{C1}, A_J^1 = 0$	$B_J^2 = 1/Imp_{C1}, A_J^2 = 0$	$B_J^3 = 1/Imp_{C1}, A_J^3 = 0,$

$$C1_J^A = max(A_J^1, A_J^2, A_J^3) \qquad C1_J^B = max(B_J^1, B_J^2, B_J^3)$$

5.2. C2: Anticipation

We base the computation of Anticipation (Ant) on the Event Synchronization (ES) algorithm defined by [Quiroga et al., 2002]. Given two simultaneously measured discrete univariate time series x_n and y_n, n=1,...,N we define events corresponding to the local maxima of the two signals. Given a time threshold τ (i.e., two events in x_n and y_n are considered synchronized if their distance in time is within τ), the ES algorithm described in [Quiroga et al., 2002] computes two quantities: Q_τ and q_τ.

The first one expresses the degree of the synchronization of two time series, the second expresses "delay asymmetry". The second one is equal to 1 if all events in x_n precede those in y_n, it is equal to -1 if all events in y_n precede those in x_n, it is equal to zero if all events in x_n occur at the same time of those in y_n.

Given 2 dancers A and B and a group of joint J, we apply ES on the time series containing local maxima of the group's speed s, obtaining the value of q_τ:

$$G_{0.5}(ES(s_J^A),ES(s_J^B)) = GES_J^A = number\ of\ frames \mid q_\tau(s_J^A, s_J^B) > 0.5$$

$$G_{0.5}(ES(s_J^B),ES(s_J^A)) = GES_J^B = number\ of\ frames \mid q_\tau(s_J^A, s_J^B) > 0.5$$

The resulting values indicate whether the movements of A on the group of joints J precede those of B (and vice-versa).

Next, we apply the following algorithm to compute C2 of both dancers on group of joints J:

if($(GES_J^A - GES_J^B) > 0$)

$$C2_J^A = 1/Imp_{C2}, C2_J^B = 0$$

else

$$C2_J^B = 1/Imp_{C2}, C2_J^A = 0$$

5.3. C3: Suddenness

Alpha-stable distributions have been introduced by [Lévy, 1925]. An alpha-stable distribution can be modeled by a probability density function characterized by four parameters (α, β, γ, δ):

- $\alpha \in (0,2]$ is the characteristic exponent that defines the width of the distribution;
- $\beta \in [-1, 1]$ the determines the skewness of the pdf;
- $\gamma > 0$ is the dispersion parameter and corresponds to variance in Gaussian distributions;
- $\delta \in (-\infty, \infty)$ is the shift from the origin of center of the pdf and corresponds to the mean value in Gaussian distributions.

We exploit alpha-stable distributions to detect sudden movements. Given a discrete time series containing the tangential velocity of a group of joints J, we fit an alpha-stable distribution AS to the probability density function of the speed of group J:

$$AS(s_J) = (\alpha_J, \beta_J, \gamma_J, \delta_J)$$

Then, for each joint J we scale α_J and we multiply it by γ_J:

$$Sud_J = \gamma_J * (1 - \frac{\alpha_J}{2})$$

In this way, Sud will increase (i.e., the movement will be considered more suddenn) if α_J is low (i.e., the tails of the probability density function are thicker) and γ_J is high (i.e., the probability density function is has greater variance.

Given the two dancers A and B and a group of joints J, we compute:

$$G_{0.5}(Sud_J^A, Sud_J^B) = GSud_J^A = number\ of\ frames \mid (Sud_J^A - Sud_J^B) > 0.5$$

$G_{0.5}(Sud_J^B, Sud_J^A) = GSud_J^B = number\ of\ frames \mid (Sud_J^B - Sud_J^A) > 0.5$

and:

if$((GSud_J^A - GSud_J^B) > 0)$

$C3_J^A = 1/Imp_{C3}, C3_J^B = 0$

else

$C3_J^B = 1/Imp_{C3}, C3_J^A = 0$

5.4. C4: Richness

We model richness of movement with the energy of movement. This modeling is justified by the feedback we received from the annotators, who used the term "richer" to identify the movements of the leader that were "larger", "with a greater excursion".

Given a discrete time series containing the tangential absolute speed s of a group of joints J, we define the Kinetic Energy KE as it follows:

$$KE_J = \frac{1}{2} m * {s_J}^2$$

where m is an approximation of the mass of joint J.

Given the two dancers A and B and a group of joints J, we compute:

$$G_{0.5}(KE_J^A, KE_J^B) = GKE_J^A = number\ of\ frames \mid (KE_J^A - KE_J^B) > 0.5$$

$$G_{0.5}(KE_J^B, KE_J^A) = GKE_J^B = number\ of\ frames \mid (KE_J^B - KE_J^A) > 0.5$$

and:

if$((GKE_J^A - GKE_J^B) > 0)$

$C4_J^A = 1/Imp_{C4}, C4_J^B = 0$

else

$C4_J^B = 1/Imp_{C4}, C4_J^A = 0$

5.5. C5: Physical Contact

This cue models the fact that a dancer is pushing with her hands toward the hands of the other dancer, or pulling the hands of the other dancer toward her.

Given dancers A and B, we compute the speed of movement of each dancer's hands with respect to the other dancer's body:

$s_{RHA}^B = speed\ of\ A's\ Right\ Hand\ toward\ B$

$s_{LHA}^B = speed\ of\ A's\ Left\ Hand\ toward\ B$

$s_{RHB}^A = speed\ of\ B's\ Right\ Hand\ toward\ A$

$s_{LHB}^A = speed\ of\ B's\ Left\ Hand\ toward\ A$

To identify if and which dancers hands are in contact during the performance, we compute the distance between all the dancer's pairs of joints:

$$d_{RHB}^{RHA} = distance\ between\ A's\ Right\ Hand\ and\ B's\ Right\ Hand$$

$$d_{LHB}^{LHA} = distance\ between\ A's\ Left\ Hand\ and\ B's\ Left\ Hand$$

$$d_{RHB}^{LHA} = distance\ between\ A's\ Left\ Hand\ and\ B's\ Right\ Hand$$

$$d_{LHB}^{RHA} = distance\ between\ A's\ Right\ Hand\ and\ B's\ Left\ Hand$$

We filter out the pairs that have a distance lower than 15 cm, then we compute:

if($d_{RHB}^{RHA} < 15cm$)

$$PC_1 = s_{RHA}^{B} + s_{RHB}^{A}$$

else

$$PC_1 = 0$$

if($d_{LHB}^{LHA} < 15cm$)

$$PC_2 = s_{LHA}^{B} + s_{LHB}^{A}$$

else

$$PC_2 = 0$$

if($d_{RHB}^{LHA} < 15cm$)

$$PC_3 = s_{LHA}^{B} + s_{RHB}^{A}$$

else

$$PC_3 = 0$$

if($d_{LHB}^{RHA} < 15cm$)

$$PC_4 = s_{RHA}^{B} + s_{LHB}^{A}$$

else

$$PC_4 = 0$$

And then:

$$PC = PC_1 + PC_2 + PC_3 + PC_4$$

$$PC_{rev} = -PC$$

$$G_{0.5}(PC, PC_{rev}) = GPC = number\ of\ frames \mid (PC - PC_{rev}) > 0.5$$

$$G_{0.5}(PC_{rev}, PC) = GPC_{rev} = number\ of\ frames \mid (PC_{rev} - PC) > 0.5$$

Finally:

if($(GPC - GPC_{rev}) > 0$)

$$C5^{A} = 1/Imp_{C5},\ C5^{B} = 0$$

else

$$C5^{B} = 1/Imp_{C5},\ C5^{A} = 0$$

5.6. C6: Gaze

We approximate the dancer's gaze with the direction of her head. This is the only cue that is not computed on the dancer's wrists or ankles. Instead, we compute the line passing from the back and the front head middle point:

$$gaze_A = line\ representing\ A's\ gaze\ direction$$

$$gaze_B = line\ representing\ B's\ gaze\ direction$$

We also compute the line between A and B as the line passing from the dancers' sternums:

$dir_{A,B} = line\ from\ A\ to\ B$

We define define the gaze of A toward B and the gaze of B toward B as it follows:

$$gaze_{A,B} = gaze\ of\ A\ toward\ B = angle(\ gaze_A\ , dir_{A,B})$$

$$gaze_{B,A} = gaze\ of\ B\ toward\ A = angle(\ gaze_B\ , dir_{A,B})$$

We check whether the two angles above are smaller than 45 degrees:

$$gaze_1 = number\ of\ frames\ |\ (gaze_{A,B} < 45)\ and\ (gaze_{B,A} > ((gaze_{A,B} + 10))$$

$$gaze_2 = number\ of\ frames\ |\ (gaze_{B,A} < 45)\ and\ (gaze_{A,B} > ((gaze_{B,A} + 10))$$

if$((gaze_1 - gaze_2) > 0)$

$$C6^A = 1/Imp_{C6}, C6^B = 0$$

else

$$C6^B = 1/Imp_{C6}, C6^A = 0$$

5.7. Computation of Leadership

Once we computed all the above cues, we sum up their values for A and B:

$$L_A = \sum^{J\in\{w,a,fb\}} \llbracket(C1\rrbracket_J^A + C2_J^A + C3_J^A + C4_J^A) + C5^A + C6^A$$

$$L_B = \sum^{J\in\{w,a,fb\}} \llbracket(C1\rrbracket_J^B + C2_J^B + C3_J^B + C4_J^B) + C5^B + C6^B$$

And we compare them:

if$((L_A - L_B) > 0)$

A is the leader

else

B is the leader

5.8 Evaluation

A paper is in preparation, presenting the evaluation of the above computational model of Leadership in a dance performance. Preliminary results show that the model correctly predicts the leadership perceived by human observers.

Bibliography

Madgwick, S. O., Harrison, A. J., & Vaidyanathan, R. (2011, June). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on* (pp. 1-7). IEEE.