Quaternionic signal processing techniques for automatic evaluation of dance performances from MoCap data

Dimitrios S. Alexiadis and Petros Daras, *Senior Member, IEEE*

***Abstract*—In this paper, the problem of automatic dance performance evaluation from human Motion Capture (MoCap) data, is addressed. A novel framework is presented, using data captured by Kinect-based human skeleton tracking, where the evaluation of user's performance is achieved against a gold-standard performance of a teacher. The framework addresses several technical challenges, including global and local temporal synchronization, spatial alignment and comparison of two "dance motion signals". Towards the solution of these technical challenges, a set of appropriate quaternionic vector-signal processing methodologies is proposed, where the 4D (spatiotemporal) human motion data are represented as sequences of pure quaternions. Such a quaternionic representation offers several advantages, including the facts that joint angles and rotations are inherently encoded in the phase of quaternions and the three coordinates variables (X,Y,Z) are treated jointly, with their intra-correlations being taken into account. Based on the theory of quaternions, a number of advantageous algorithms are formulated. Initially, global temporal synchronization of dance MoCap data is achieved by the use of quaternionic cross-correlations, which are invariant to rigid spatial transformations between the users. Secondly, a quaternions-based algorithm is proposed for the fast spatial alignment of dance MoCap data. Thirdly, the MoCap data can be temporally synchronized in a local fashion, using Dynamic Time Warping techniques adapted to the specific problem. Finally, a set of quaternionic correlation-based measures (scores) are proposed for evaluating and ranking the performance of a dancer. These quaternions-based scores are invariant to rigid transformations, as proved and demonstrated. A total score metric, through a weighted combination of three different metrics is proposed, where the weights are optimized using Particle Swarm Optimization (PSO). The presented experimental results using the Huawei/3DLife/EMC^2 dataset are promising and verify the effectiveness of the proposed methods.**

***Index Terms*—Skeleton tracking, Motion Capture data, dance analysis, Quaternions, vector signal processing**

I. Introduction

Future social networks move towards immersive, collaborative environments [1] that can support real-time realistic interaction between humans, via human motion tracking and understanding [2]. In a related application scenario, described by the Huawei 3DLife/EMC^2 grand challenge[1], an online dance class is considered. A teacher is able to illustrate to online users Salsa choreography steps of their choice. After viewing the captured sequence, another online user, such as a student, may attempt to mimic the dance steps and then get feedback from a system about her/his performance. In this work, a number of technical challenges, needed to be addressed for the described scenario of interaction, are studied. In the following subsections, the exact problem is formulated and the proposed methodology is summarized along with the contributions of the paper.

A. Problem formulation

The problem of dance analysis can be considered as a special case of human activity analysis [2], [3], [4]. The majority of relevant papers in human analysis addresses the problem of human motion/action recognition. The current paper shifts the focus from the action classification perspective ("which action was performed?") to the evaluation objective ("how well was it performed?"). Evaluating and ranking the performance of users based on MoCap data is a challenging research problem that has not been adequately addressed so far.

The exact formulation of the problem can be summarized as follows. A "gold" template MoCap dance sequence for a specific dance choreography is prerecorded. The "gold" performance is executed by a teacher under a specific audio sample for the given choreography. The motion of an amateur dancer (probably located at a distant place) is then captured, as he/she attempts to execute the specific choreography under the same audio sample. Firstly, since the two MoCap dance sequences were captured at different time instances they are not globally synchronized (in time) with each other. Secondly, the captured MoCap data refer to different global coordinate systems, since dancers may have been captured by different setups and are allowed to be placed anywhere with respect to the capturing sensor. Thirdly, the MoCap system is of low-cost and thus far from an ideal one, introducing tracking inaccuracies, missing values and incomplete capture sequences. Finally, the amateur dancer may have executed only a part (subsequence) of the whole choreography. Given these facts, the first problem that needs to be addressed is the global spatiotemporal alignment of the incomplete amateur's MoCap sequence with the reference one. Local temporal synchronization may also need to be addressed, in order to compensate local phase differences, if this is allowed in the dance evaluation scenario. Solving these problems will first of all enable the visual inspection of the amateur dance in comparison to the "gold" one, which is useful

The authors are with the Information Technologies Institute, Centre for Research and Technology - Hellas, 6th km Charilaou - Thermi, GR-57001, Thessaloniki, Greece, Tel. +30 2310 464160 (ext. 277), Fax. +30 2310 464164 (e-mail: dalexiad@iti.gr; daras@iti.gr).

[1]http://perso.telecom-paristech.fr/~essid/3dlife-gc-11

for dance training. The second problem addressed in the paper is the automatic evaluation of the executed amateur dance by calculating an overall score, as well as instantaneous scores for his/her performance. This would enable an intelligent system to provide feedback to the dancer on how to improve his/her performance, making the presence of a teacher unnecessary.

B. System overview and methodology - Main Contributions

An overview of the proposed system is given in Fig. 1. The MoCap data, for both the professional and the amateur dancer performers, is obtained via human skeleton tracking from depth-maps, captured by Microsoft Kinect sensors. The tracked joint positions, which constitute vector signals, are represented using quaternions. Then, using a set of appropriate quaternionic signal processing techniques, global temporal synchronization of the amateur dance with the "gold" one is realized. Local phase differences can also be compensated by a local synchronization method. Additionally, a fast quaternionic method for spatially aligning the two dances is applied, if this is needed. Given that the dances have been synchronized, they are compared to each other, in order to provide a set of dance evaluation scores. Finally, different scores are combined with the use of appropriate weights to produce a final score.

In this paper, MS Kinect sensors are used for capturing dancers motion. However, the proposed processing framework could potentially be applied with human motion data obtained by other technologies, such as 3D optical MoCap technologies (e.g. VICON[2]), wearable-sensors [3], [4], [5], or even from monocular visual data [6], [7], [8] or multi-view visual data [9]. The use of Kinect sensors in our system makes it viable for a large range of users, including home enthusiasts. Notice that although the Kinect sensor has recently attracted the attention of many researchers [10], [11], not much attention has been paid on Kinect-based human motion analysis.

In order to address the problems formulated in paragraph I-A, a quaternionic signal processing framework for dance vector signal analysis is proposed. Quaternions have been extensively used in computer graphics to represent rotations. Quaternions theory has also recently been used in various computer-vision applications, such as in color image analysis via quaternionic Fourier transforms [12], color image registration [13], motion estimation in color image sequences [14], [15], image classification [16] and others. However, to the authors knowledge, it has rarely been used in a human motion analysis application, such as the one addressed in the paper. Some relevant works are given in section II.

Using the proposed quaternionic framework, the 4D (spatiotemporal) motion data are encoded and handled in a holistic manner, letting the three coordinates variables (X,Y,Z) to be treated jointly and their intra-correlations to be taken into account. Additionally, rotations are inherently encoded in the phase of quaternions. Thus, for example, using a quaternionic represention of the joint positions, joint angles are inherently encoded in the phase of quaternions. More importantly, using the mature (although quite complicated) theory of quaternions, a number of algorithms that are endowed with significant advantages can be formulated. For instance, as described in this paper, fast algorithms for the estimation of rigid transformations can be formulated, or correlation measures that are invariant to rigid transformations can be proposed and exploited.

[2]http://www.vicon.com/

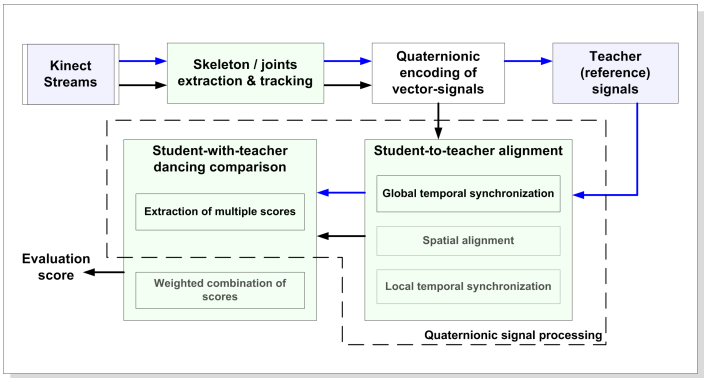


Fig. 1. A schematic description of the overall system. Optional block operations are denoted with gray font.

The main contributions of the paper can be summarized as follows: i) It is among the first few papers that focus on the problem of automatically evaluating and ranking the performance of users, based on their motion, as captured by MoCap technologies; ii) To the authors' knowledge, it is the first paper that extensively evaluates the proposed methodologies based on ground-truth data. We also test other relevant state-of-the-art methods [17] with these ground-truth data, that have not been previously assessed; iii) The paper presents a novel quaternionic framework for MoCap data analysis and describes methodologies for the global and local synchronization of MoCap sequences, as well as their fast spatial alignment; iv) It introduces the use of quaternionic correlation metrics that couple the dependencies of the three dimensions and present invariance to global rigid transformations.

C. Paper organization

The rest of the paper is organized as follows: In section II, we present the related work in the field of human motion analysis and evaluation. In section III, we briefly describe the Huawei-3DLife/EMC[2] dataset, as well as the Kinect human skeleton tracking module, used for tracking of the dancers' movements. In section IV, we provide the necessary quaternions theory and describe the proposed quaternionic or other representations of dance (generally human motion) signals. In section V, we present the proposed methodologies for the student-to-teacher temporal synchronization and spatial alignment. Next, in section VI, we propose a set of quaternionic or other metrics that can be used as dance evaluation scores. In section VII, we describe the methodology for the selection of appropriate weights to be used for combining multiple score metrics. Finally, in section VIII, we present a set of experimental results, before concluding in section IX.

II. Related work

Quaternions have been used in various scientific fields, such as in computer graphics and biomechanics, in order to

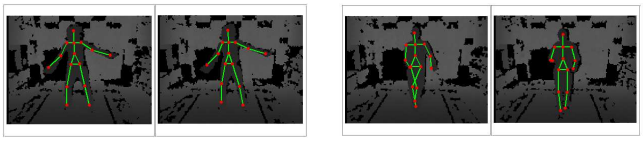
represent 3D rotations of body segments. For example, in [18], the problem of motion of virtual characters is addressed, by making use of quaternions to create distance metrics between multiple character poses. The objective is to blend multiple poses, with the aim to generate both interactive and expressive motions of virtual characters in computer graphics. The use of quaternions provides both computationally efficiency and mathematical robustness (e.g. avoiding the gimbal-lock effect in the Euler-angle representation of rotations). In [19], quaternions are also used to represent rotations and a quaternions-based algorithm for rigid body motion interpolation between two poses is proposed, with application to biomechanics.

Numerous works for human motion analysis from MoCap data can be found in the literature. However, most of them deal with the motion segmentation and classification task, while only a few of them address the motion evaluation problem. Here, we shortly present the most relevant works.

A human motion analysis application, where the use of quaternions has been proposed, is the one described in [20]. The problem of human motion segmentation is addressed. More specifically, the segmentation of MoCap data into dinstinst motion patterns is efficiently addressed using Principal Component Analysis (PCA), Probabilistic PCA or Gaussian Mixture Models (GMMs). Quaternions are used to represent rotations of joints, relative to their parent joints in a skeletal body hierarchy. However, the benefit of using quaternions is not clearly explained and the paper focuses on the motion segmentation task. A skeletal body hierarchical representation, similar to that of [17], is used.

In another more recent work [21], the authors efficiently address the problem of segmenting a motion stream into district motion patterns and recognizing them. The two problems (segmentation and recognition) are treated simultaneously. This is achieved by introducing and exploiting a similarity measure, nominated as kWAS, that is based on Singular Value Decompositions (SVD) of the motion data. kWAS measures the similarity of two motions patterns by calculating the angular similarity (inner product) of singular vector pairs and using the corresponding singular values for weights. A kWAS-based algorithm is proposed to treat segmentation and classification jointly. The algorithm presents high accuracy. However, the kWAS measure is not invariant to rotations of the human subject with respect to a static global coordinate system. Thus, the efficiency of the algorithm was evaluated with high-accuracy optical MoCap technology (VICON), where the positions of different body segments are given relative to a moving coordinate system at the subject's pelvis.

The motion classification task is also addressed in [22]. An interactive dancing game using optical MoCap technology is described, where a virtual partner recognizes and responds to the player's movement. The proposed system is able to classify the player's moves among eight different template dance patterns, in a continuous manner. In order to account for temporal speed variations, a progressive block-based matching method is proposed, which imitates Dynamic Time Wrapping (DTW) by locally matching blocks of frames in ascending time order. The system was evaluated through user studies and a questionnaire's filling.



(a) Anne-Sophie-k vs Anne-Sophie-k (b) Bertand vs Gabi

Fig. 2. Skeleton-tracking snapshots: (a) Anne-Sophie-k (professional female dancer) in choreography c2. Notice that the two dance instances are very similar for the professional dancer; (b) Bertand (professional male dancer) vs Gabi for choreography c3. Generally, tracking is robust. However, there are a few cases where tracking of some joints is lost for few frames, due to self-occlusions, as in the example shown in (b).

A more relevant work that addresses the dance evaluation problem is presented in [23]. A dance training system based on MoCap and virtual reality technologies is described. The system evaluates the dissimilarity between the dances of a learner and a virtual teacher and provides feedback. For each time instance (frame), the dance dissimilarity is given from the euclidean distance between the virtual teacher's template posture and the student's posture. A total score is obtained by averaging over all frames. Three commonly used features, the joint positions, velocities and angles, were used and evaluated. However, the evaluation was performed only within a binary classification framework, by assessing how well the system can discriminate between similar and dissimilar dance motions.

In another recent and interesting Kinect-based relevant work [17], a gesture classification system for skeletal wireframe motion is presented. A hierarchical angular skeleton representation is proposed and independence to the global coordinate system (camera position and orientation) is achieved based on a set of heuristics. Specifically, the tracked torso is fitted with a single reference frame and the orientations of the joints are parameterized using that frame. Therefore, the system's invariance to transformations depends on how well and robust the torso has been tracked and fitted to the reference frame. The dance evaluation problem is also shortly addressed, but no experimental results are given.

Finally, some other works related to dance analysis can be found in the literature [24], [25], [26]. However, in these works emphasis is given mainly on dance segmentation and choreographic representations and is based on the assumption that the dancers are professional. Additionally, they introduce the musical information for dance analysis. For example, in [24] the authors describe an approach for the representation of dance gestures in Samba and propose a method that searches for shared elements in dance and music at the metrical level. Similarly, in [25] musical information is introduced for motion structure analysis. A method that automatically detects the musical rhythm and segments the original motion into primitive dance motions is described.

III. Input Data and Skeleton Tracking

A. Dataset

The dataset from the Huawei 3DLife/EMC^2 Grand Challenge was utilized in this work. It contains recordings of Salsa dancers, captured by a variety of equipment, including a Microsoft Kinect sensor. Recordings of two professional dancers

(a male and a female) and 13 amateur (student) dancers (8 males and 5 females) in six different Salsa choreographies (c1-c6), constitute the dataset. For this work, we use only the Kinect depth-map recordings, making the approach viable for a large range of users. "Ground-truth" evaluation ratings, provided by Salsa dance experts, are also included in the database.

B. Kinect-based skeleton tracking

The depth maps of the dataset were recorded using the OpenNI[3] API. Therefore, the OpenNI high-level skeleton tracking module was utilized for detecting the captured dancer and tracking his/her body joints. More specifically, using the OpenNI tracking module, the positions of 15 joints (Head, Neck, Torso, Left and Right Shoulder, L/R Elbow, L/R Wrist, L/R Hip, L/R Knee and L/R Foot) are tracked, as shown in Fig. 2. Our experiments showed that skeleton tracking is quite effective with the underlying dance captures, although there are a few cases where tracking of some joints (mainly the knees) is lost for a few frames, due to self-occlusions (see Fig. 2(b)). Obviously, more accurate future releases of OpenNI could potentially improve tracking or the Microsoft's Kinect SDK[4] could have been employed for more accurate tracking in a real-time application (unfortunately the Kinect SDK cannot be used with pre-recorded data). Additionally, more accurate approaches for human motion capture, e.g. using wearable sensors or "gold-standard" 3D optical MoCap technologies (e.g. VICON), could potentially increase the accuracy of the proposed automatic evaluation methods.

IV. Quaternions theory and skeleton data representations

The skeleton tracking module provides the positions of the dancer's joints for each frame. The position of the j-th joint with time is a 3D vector signal

$$\vec{p_j}(t) = [X_j(t), Y_j(t), Z_j(t)]^{\mathrm{T}}, \;\; j = 1, .., J, \;\; t = 0, .., T-1, \quad (1)$$

where J = 15 is the total number of tracked joints and T is the total number of frames.

We propose an automatic dance evaluation approach that represents and handles the skeleton tracking data using hypercomplex numbers and specifically quaternions [27], [13], in order to handle the three coordinate variables X, Y and Z in a holistic manner, i.e. jointly. Therefore, before going into the details of the proposed representations, we provide some fundamental theory on quaternions.

A. Notes on quaternions

1) Fundamentals: Quaternions theory, introduced by Hamilton [27], constitutes a generalization of complex numbers theory, where instead of a scalar imaginary part, a 3D "vector" imaginary part is considered. More specifically, a quaternion q is composed of its scalar (real) part $\mathcal{S}(\mathrm{q})$ and a vector part $\mathcal{V}(\mathrm{q})$:

$$\mathrm{q} = \mathcal{S}(\mathrm{q}) + \mathcal{V}(\mathrm{q}), \quad \mathcal{S}(\mathrm{q}) = q_s, \;\; \mathcal{V}(\mathrm{q}) = q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}, \quad (2)$$

where $q_s, q_i, q_j, q_k \in \mathbb{R}$ and $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ are three distinct imaginary units, such that

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1, \quad (3a)$$

$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \;\; \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \;\; \mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \quad (3b)$$

This equation defines the (Hamilton) product of quaternions, which is a non-commutative operation.

The conjugate of a quaternion q is given by $\overline{\mathrm{q}} = \mathcal{S}(\mathrm{q}) - \mathcal{V}(\mathrm{q})$ and its modulus or norm by

$$|\mathrm{q}| = \sqrt{\mathrm{q}\overline{\mathrm{q}}} = \sqrt{\overline{\mathrm{q}}\mathrm{q}} = \sqrt{q_s^2 + q_i^2 + q_j^2 + q_k^2}. \quad (4)$$

The quaternions with zero scalar part, are referred to as "pure" quaternions, while those with unit modulus are referred to as "unit" or "unitary".

Euler's formula for complex numbers is generalized in the quaternionic algebra. It holds: $e^{\boldsymbol{\mu}\phi} = \cos(\phi) + \boldsymbol{\mu}\sin(\phi)$, where $\boldsymbol{\mu}$ is a unit pure quaternion ($\boldsymbol{\mu}^2 = -1$). Therefore, any quaternion can be expressed in its polar form: $\mathrm{q} = |\mathrm{q}|e^{\boldsymbol{\mu}_\mathrm{q}\phi_\mathrm{q}}$, where

$$\boldsymbol{\mu}_\mathrm{q} = \mathcal{V}(\mathrm{q})/|\mathcal{V}(\mathrm{q})|, \quad \phi_\mathrm{q} = \tan^{-1}(|\mathcal{V}(\mathrm{q})|/\mathcal{S}(\mathrm{q})) \quad (5)$$

are referred to as the "eigen-axis" and "eigen-phase" (or "eigen-angle"), respectively.

Quaternions theory has many analogies with complex numbers theory; However, as dictated by (3), quaternions' multiplication is non-commutative, introducing difficulties in the use of quaternions. Additional theoretical notes on the quaternionic algebra can be found in Appendix A, while for a more detailed presentation the reader is referred to [13]. Throughout the rest of the paper, we tried to keep a uniform notation, expressing real numbers by italic letters (e.g. "q_s"), full quaternions by normal (non-italic) letters (e.g. "q") and pure quaternions by bold symbols (e.g. "**p**").

2) Quaternionic cross-covariance: The cross-covariance of two ("pure" in our case) quaternionic signals $\mathbf{p}(t)$ and $\mathbf{q}(t)$ is given by:

$$\mathrm{C}(\tau) = \mathcal{C}\{\mathbf{p}^v(t), \mathbf{q}^v(t)\} = \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{p}^v(t)\,\overline{\mathbf{q}^v(t-\tau)}, \quad (6)$$

where $\mathbf{p}^v(t) = \mathbf{p}(t) - \mathbf{p}^c$ and $\mathbf{q}^v(t)$ (calculated similarly) are the varying parts of the vector signals, calculated by subtracting their centroids

$$\mathbf{p}^c = \frac{1}{T}\cdot\sum_{t=0}^{T-1}\mathbf{p}(t) \quad (7)$$

and $\mathbf{q}^c$, respectively.

Notice that the Hamilton product of the two pure quaternions **p** and **q** is a full quaternion C. Any rotation between **p** and **q** is effectively encoded in the phase of C. This is a key element towards achieving rotation invariance in the correlation-based metrics defined in next sections. Notice moreover, that a similar covariance metric could not be defined

[3] http://www.openni.org/

[4] http://www.microsoft.com/en-us/kinectforwindows/

using standard 3D vectors notation. The use of the vectors inner product, instead of the quaternions product, would not let us introduce rotation invariant metrics, as explained and demonstrated in subsection VI-B.

3) Quaternionic Fourier Tranform: The forward discrete QFT of a ("pure" in our case) quaternionic signal $\mathbf{p}(t)$ is defined similarly to [12]:

$$\widehat{\mathrm{P}}(u) = \sum_{t=0}^{T-1} e^{-\boldsymbol{\mu}\frac{2\pi u}{T}t}\,\mathbf{p}(t), \quad u = -T/2, \ldots, T/2-1, \quad (8)$$

where $\boldsymbol{\mu}$ is any unit pure quaternion ($\boldsymbol{\mu}^2 = -1$). The inverse QFT is obtained from (8) by changing the sign of the exponential and summing over u instead of t. Actually, due to the fact that quaternion's multiplication is not commutative, the QFT in (8) is one of the two different versions of QFT, the QFT-Left. The QFT-Right is defined as in (8) but with the exponential term multiplying $\mathbf{p}(t)$ from the right. For further details on the QFT the reader is referred to [12].

B. Skeleton data representations

1) Absolute joint positions: Moxey et al. [13] used "pure" quaternions to encode color images as vector fields. Adopting a similar representation scheme, we use pure quaternions to encode the absolute 3D position of each joint during time, as follows:

$$\mathbf{p}_j^{\mathrm{ab}}(t) = \mathbf{i}\cdot X_j(t) + \mathbf{j}\cdot Y_j(t) + \mathbf{k}\cdot Z_j(t), \quad j = 1, .., J, \;\; t = 0, .., T-1. \quad (9)$$

2) Pre-filtering: In order to suppress high-frequency tracking noise, which may affect subsequent calculations (e.g. the calculation of joint velocities), we apply low-pass filtering to the absolute joint positions, as a preprocessing step. Pre-filtering is performed in the discrete QFT space to handle jointly the three coordinate variables. Filtering is applied to the absolute position of each joint separately. The transform axis $\boldsymbol{\mu}$ (see (8)) is selected here equal to $\boldsymbol{\mu} = \frac{\sqrt{3}}{3}\mathbf{i} + \frac{\sqrt{3}}{3}\mathbf{j} + \frac{\sqrt{3}}{3}\mathbf{k}$, in order to treat equally the three dimensions. In the discrete QFT domain, we apply a gaussian window:

$$G(u) = e^{-2\left(\frac{u}{\sigma \cdot T}\right)^2}, \quad (10)$$

with $\sigma = 0.25$. With this value of σ and for $u/T \approx 1/10$ (this corresponds to a continuous-time frequency equal to 1/5 of the sampling frequency $f_s = 30$Hz, i.e. 6Hz), the gaussian reduces to $\sqrt{2}/2$. This means that we have a cut-off frequency (-3dB half-bandwidth) equal to 6Hz, which is a reasonable high frequency, even for fast motions such as in dancing. Indeed, according to our experiments, the dance signals are not practically affected. An example video, showing the effect of pre-filtering to the dance MoCap data, can be found in the Supplementary material.

3) Relative joint positions: The use of the 3D absolute joint positions is possible for the evaluation task. However, due to the global motion, the 3D absolute positions of the joints (especially their low-frequency components) are highly correlated. An example is given in Fig. 3(a). Consequently, in order to make evaluations on a per-joint basis and subsequently identify the joints which contribute more to the dancer's performance, it makes more sense to "subtract" the global motion. Therefore, we use the joint positions relative to the global dancer's position:

$$\mathbf{p}_j(t) = \mathbf{p}_j^{\mathrm{ab}}(t) - \mathbf{p}_1^{\mathrm{ab}}(t), \quad j = 2, 3, \ldots, J, \quad (11)$$

where $\mathbf{p}_1^{\mathrm{ab}}(t)$ is the torso's absolute position, which contains the global motion. An example is given in Fig. 3(b). We keep the original torso position as a feature, letting $\mathbf{p}_1(t) = \mathbf{p}_1^{\mathrm{ab}}(t)$. There are, however, some issues to be discussed at this point: **I1)** Notice that the relative joint position $\mathbf{p}_j(t)$ has still a small amount of correlation with its parent's joint position (see 3(b)). For example, the left foot's position is normally slightly correlated with the left knee's position. Therefore, if robust and noise-free accurate tracking was guarantied, an ever more sensible approach would be to use the joint's position relative to its parent joint. However, this is not always the case. For example, tracking of the knees is sometimes lost for a few frames (due to self occlusions), while feet are correctly tracked. In this case, subtracting the foot position from its parent joint (knee) position would propagate noise/errors to the foot position, although it was robustly tracked. On the other hand, the torso position (global position) is always robustly tracked. Therefore, in order to avoid such error-propagation situations and given that the joint-to-parent correlations are not very high, we preferred to use the representation of (11). **I2)** One could possibly opine that joints near the torso (e.g. shoulders and hips) do not induce strong motion signals relative to the torso. Thus, the underlying Signal-to-Noise Ratio (SNR) will not be high and evaluation for such joints will produce poor results. Therefore, these joints should not be used (or they could be even used for a more robust estimation of the torso position, as in [17]). We preferred however not to use such heuristic arguments. Instead, we let the automatic algorithms of section VII to decide whether and how much these joints contribute to the dancer's performance. Avoiding such heuristics makes the evaluation approach more generic.

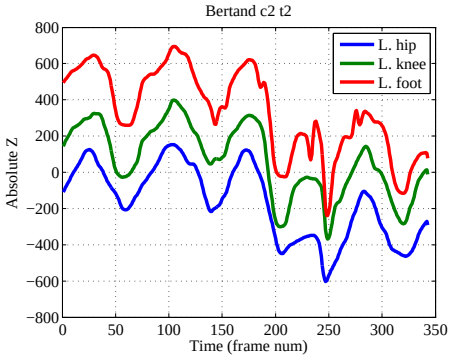
4) Relative joint velocities: In order to use the dynamics of dancing movements, we consider also the instantaneous relative velocities of the joints:

$$\mathbf{v}_j(t) := \frac{\partial \mathbf{p}_j(t)}{\partial t} \quad \leftarrow \quad \mathbf{p}_j(t) - \mathbf{p}_j(t-1), \quad j = 1, 2, \ldots, J. \quad (12)$$

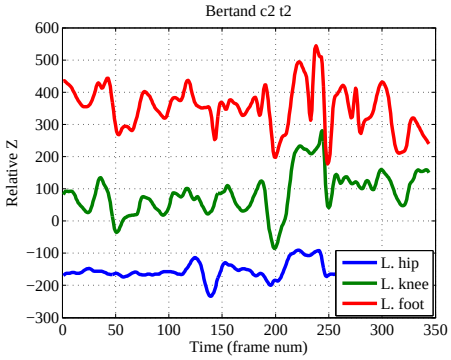
Notice that longer derivative filters (instead of $[-1, 1]^{\mathrm{T}}$) could have been used for approximating the derivative. However, according to our experiments, the lowpass pre-filtering of subsection IV-B2 seems to be adequate for removing high-frequency tracking-noise components.

5) Rigid transformations with the quaternionic dance representations: Consider an original dance performance and its rigid-body transformed version, which was obtained by scaling the dancer by k, rotating about an axis $\vec{\mu}$ through an angle θ and translating by $\vec{d}$. If the original dance is represented by the quaternionic absolute joint positions $\mathbf{p}_j^{\mathrm{ab}}(t)$, as in (9), then the transformed version is expressed by (we drop j for notational simplicity):

$$\mathbf{p}_{\mathrm{RST}}^{\mathrm{ab}}(t) := k \cdot \mathrm{R}\,\mathbf{p}^{\mathrm{ab}}(t)\,\overline{\mathrm{R}} + \mathbf{d}, \quad \mathrm{R} = e^{\boldsymbol{\mu}\frac{\theta}{2}}, \quad (13)$$

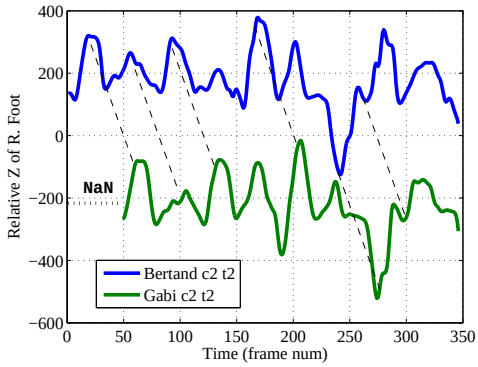


(a) Absolute joint positions

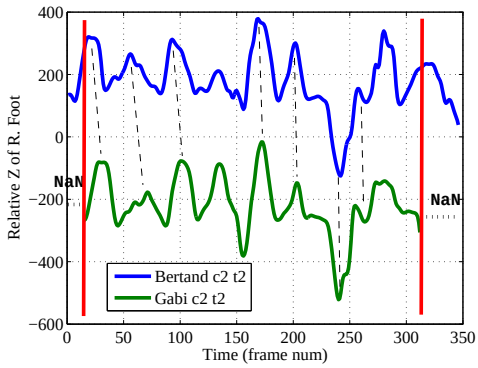


(b) Relative joint positions

Fig. 3. (a) Absolute position signals (Z component) of the left hip, knee and foot for Betrand (professional dancer) in c2 t2. (b) The corresponding relative positions. In each graph, signals were intentionally shifted in the vertical axis, for better visualization. In (a), the quaternionic cross-correlation coefficients are: hip-knee: 0.96, hip-foot: 0.92, knee-foot:0.96; In (b), the quaternionic cross-correlation coefficients are: hip-knee: 0.64, hip-foot: 0.29, knee-foot: 0.63. Similar values of correlation were observed for other dance performances.



(a) Before global synchronization



(b) After global synchronization

Fig. 4. Global temporal synchronization between Bertand c2 t2 and Gabi c2 t2: (a) The shortest captured sequence was NaN-padded at the beginning. (b) After the estimation of the global time shift, one sequence is (circularly) shifted and the common subsequence is extracted (red lines). See subsection V-A.

where $\boldsymbol{\mu}$ is the axis of rotation in a quaternionic representation and R is a quaternion that expresses the rotation (for details see Appendix A), while $\mathbf{d}$ is a pure quaternion that expresses the translation. Due to the linearities of (11) and (12), equation (13) holds also for the relative joint positions $\mathbf{p}_j(t)$, as well as for the relative joint velocities $\mathbf{v}_j(t)$.

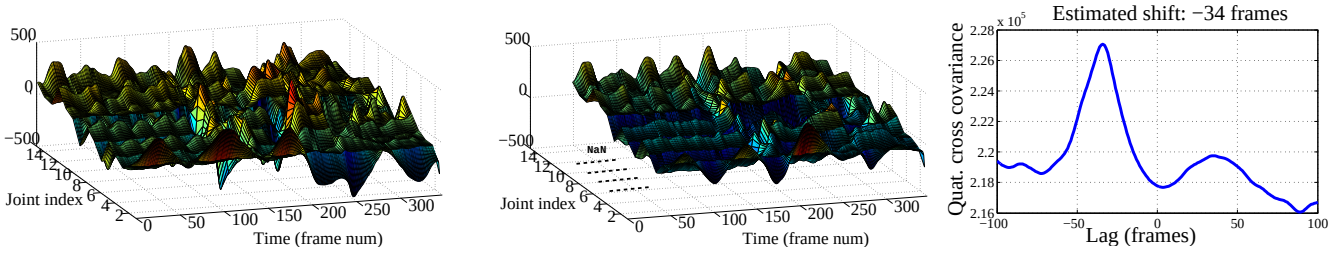
6) Hierarchical representations: A powerful hierarchical angular skeleton data representation is introduced in [17]. Although this representation scheme does not make use of quaternions, we shortly present it here, since it is used in the evaluation section. This scheme maps the skeleton motion data to a set of 19 feature time-series. Three features, the yaw, pitch and roll angles (Tait-Bryan angles) are defined for the whole human's torso, eight angular features are introduced for the "first order joints", i.e. elbows and knees and eight angular features are introduced for the "second order joints", i.e. wrists and feet. Summarizing, the features are calculated as follows: a) The orientation (three orthogonal axes) of the torso frame is found by first estimating the vertical torso axis (principal axis) via Principal Component Analysis (PCA) on the torso's joint positions and secondly finding the Left-to-Right shoulder axis; b) Two angular features (azimuth and elevation) are extracted for each 1st order joint by considering a spherical coordinate system around its parent joint on the torso; c) Similarly, two angular features are extracted for each 2nd order joint by considering a spherical coordinate system around its parent 1st order joint. For further details, the reader is referred to [17].

This is a powerful representation, in the sense that it maps the motion data into a relatively small set of features that retains the salient aspects of motion. Moreover, the 1st and 2nd order joint features are invariant to rigid transformations. The torso frames are not rotation invariant; however, they can be straightforwardly used to estimate the global rotation between two dance motion sequences. It is based on the assumption that the points on the human torso (including neck, shoulders and hips) do not exhibit strong independent motions, which is a plausible assumption most of the times, but probably not always. Additionally, the robustness of the 2nd order features strongly depend on how robustly and accurately 1st order joints were tracked.

V. Temporal Synchronization and Spatial Alignment

A. Global temporal synchronization

The actors perform a specific choreography according to a specific common audio sample. Therefore, ideally two perfect performances (of the same choreography) should perfectly



(a) $\mathbf{P}(j,t)$ (Z component) (b) $\mathbf{Q}(j,t)$ (Z component) (c) $|C_{\text{total}}(\tau)|$

Fig. 5. (a),(b) Relative position signals (Z component) of all joints, for the professional dancer (Betrand c2 t2) and the amateur dancer (Gabi c2 t2). These actually correspond to the Z component of the quaternionic arrays $\mathbf{P}(j,t)$ and $\mathbf{Q}(j,t)$ (subsection V-A). (c) The corresponding "all-joints" quaternionic cross-covariance. The estimated time-shift is $\tau_{\max} = -34$ frames.

match in motion signal shape, as well as in performance speed. However, the Kinect captures were not synchronized and the captured dancing sequences to compare do not have the same length. Furthermore, the time instance at which the tracking module detects the dancer and starts tracking is not common. Thus, when comparing two tracked dances, normally one constitutes a subsequence of the other. The objective is to match and synchronize the two dances by finding the common dance part that is present in both sequences. The steps of the employed algorithm are summarized as follows:

S1) For each sequence, we keep only the frames in which the dancer is tracked, i.e. the frames after the detection of the dancer.

S2) We pad the shortest of the two sequences with NaN (Not-a-Number) values, at the beginning.

An example result is shown in Fig. 4(a), where the relative left-foot position of two dancers is presented, after padding the shortest sequence with NaN values. It is important to highlight that the padded NaN values are not taken into account in the calculations that follow (15). Padding with NaN values to produce dances with common length is only useful for calculating (quaternionic) correlations using circular shifts. We use only the relative joints positions information, although velocities could be also useful. In order to handle jointly the data for all joints, the algorithm continues as follows:

S3) Construct the $J \times T$ pure quaternionic array

$$\mathbf{P}(j,t) = [\mathbf{p}_1^v(t), \mathbf{p}_2^v(t), \ldots, \mathbf{p}_J^v(t)]^{\mathrm{T}}, \tag{14}$$

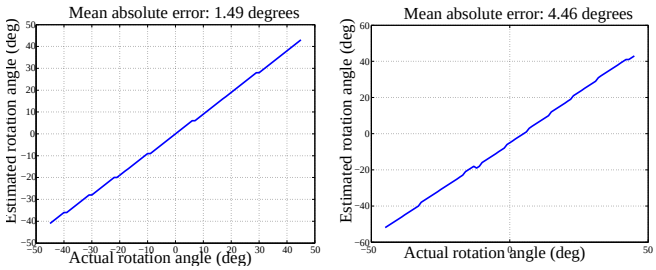
where $\mathbf{p}_j^v(t)$ is the time-varying of $\mathbf{p}_j(t)$, rows correspond to joints and columns to time (see Fig. 5(a),(b)).

S4) Calculate the modulus $|\mathrm{C}_{\text{total}}(\tau)|$ of the "all-joints" quaternionic cross-covariance, as:

$$\mathrm{C}_{\text{total}}(\tau) = \frac{1}{J \cdot T} \sum_{j=1}^{J} \sum_{t=0}^{T-1} \mathbf{P}(j,t)\, \overline{\mathbf{Q}(j,t-\tau)}. \tag{15}$$

S5) The lag $\tau_{\max} = \arg\max\{|\mathrm{C}_{\text{total}}(\tau)|\}$, corresponding to the maximum of the function, constitutes the estimate of the time-shift between the dancing sequences (see Fig. 5(c)).

An example result is depicted in Fig. 4(b). When the temporal shift is estimated and the dances are synchronized, only the common subsequence (dance part that is present in both sequences) is cropped from the sequences and used in the evaluation task. This is illustrated in Fig. 4(b), where one



(a) Rotated Gabi vs Bertand (b) Rotated Thomas vs Bertand

Fig. 6. Estimated rotation angles vs the actual (applied) ones in two relevant experiments - See subsection V-B for details.

sequence is (circularly) shifted based on the estimated time lag and then the common subsequence (red lines) is extracted.

It can be shown that the presented methodology for the global temporal shift estimation $\tau_{\max}$ is approximately invariant to rigid-body transformations (translation, scaling, rotation). This was verified experimentally and it is proved in Appendix B.

B. Fast spatial alignment

Given two dance sequences, which are assumed to be similar except for an unknown rigid transformation (rotation, scaling and translation, RST), their relative joint positions $\mathbf{p}_j(t)$ and $\mathbf{q}_j(t)$ are related through (13) (with the equal sign "$=$" replaced by the approx. equal sign "$\approx$"). Inspired by the ideas of [13], a simple algorithm for the estimation of the transformation between the dances can be summarized as:

S1) The unknown translation $\mathbf{d}$ can be estimated by simply finding and subtracting the centroids $\mathbf{p}^c = \frac{1}{J \cdot T} \cdot \sum_{j=1}^{J} \sum_{t=0}^{T-1} \mathbf{p}_j(t)$ and $\mathbf{q}^c$, respectively.

S2) Construct the $J \times T$ arrays $\mathbf{P}(j,t)$ and $\mathbf{Q}(j,t)$, as in (14).

S3) The unknown scaling factor k is estimated from the ratio:

$$k \leftarrow \sqrt{\frac{\gamma(|\mathbf{P}(j,t)|)}{\gamma(|\mathbf{Q}(j,t)|)}}, \tag{16}$$

where $\gamma(|\mathbf{P}(j,t)|)$ and $\gamma(|\mathbf{Q}(j,t)|)$ are the mean modula of $\mathbf{P}(j,t)$ and $\mathbf{Q}(j,t)$, respectively.

S4) Compute the "all-joints" quaternionic cross-covariance $\mathrm{C}_{\text{total}}(\tau)$ from (15). Given that the sequences were already synchronized, it is adequate to compute $\mathrm{C}_{\text{total}}(\tau)$ only for zero lag $\tau = 0$, since there will be the peak position.

S5) Compute the rotation angle and rotation axis from the "eigen-phase" ϕ_1 and "eigen-axis" $\boldsymbol{\mu}_1$ of $\mathrm{C}_{\text{total}}(0) = |\mathrm{C}_{\text{total}}(0)|e^{\boldsymbol{\mu}_1\phi_1}$. Details for this step are given directly below.

Estimation of rotation angle: Practically, it is expected in our application that the actual rotation is about the Y-axis $(0,1,0)^{\mathrm{T}}$. This corresponds to the quaternion axis **j**. Thus, we actually search for ϕ that minimizes the mean squared distance:

$$D(\phi) = \frac{1}{R}\sum_{r=1}^{R}\left|e^{\boldsymbol{\mu}_1\frac{\phi_1}{2}}\mathbf{p}(r)e^{-\boldsymbol{\mu}_1\frac{\phi_1}{2}} - e^{\mathbf{j}\frac{\phi}{2}}\mathbf{p}(r)e^{-\mathbf{j}\frac{\phi}{2}}\right|^2, \quad (17)$$

where $\mathbf{p}(r)$ are "test" randomly selected unit pure quaternions, $R = 20$ for the experimental results provided below, ϕ_1 and $\boldsymbol{\mu}_1$ are the "eigen-phase" and "eigen-axis" of $\mathrm{C}_{\text{total}}(0)$, respectively. In other words we search for the rotation angle ϕ about the Y-axis that has the same effect (in the MSE sense) with the rotation about $\boldsymbol{\mu}_1$ by ϕ_1. The "test" pure quaternions $\mathbf{p}(r)$ are obtained considering a zero-mean normal distribution for each vector component and then normalization to unit norm. The minimization of $D(\phi)$ is realized via full search in the ϕ space, with a search step equal to 1 degree. Full search does not introduce high computational times, because the search space is 1D.

It is straightforward to show the validity of the steps for translation and scale estimation. The validity of the rotation-estimation step is proved in Appendix B. Here, we give some demonstrating examples. For the results of Fig. 6(a), the dance performance "Gabi c2 t2" was scaled by $k = 1.3$ and rotated about the Y-axis through an angle from -45^o to 45^o. Then, considering the respective "gold" performance of the professional dancer Bertand, we estimate the scaling factor and rotation angle using the above algorithm. The estimated value for the scale factor was $\hat{k} = 1.21$. The difference between $\hat{k}$ and the applied value k can be partially justified by the fact that Gabi is slightly shorter than Bertand (see Fig. 2(b)). The estimated rotation angles vs the applied ones are given in 6(a). Notice that the applied rotation is not strictly equal to the actual one, because the original dances may be slightly rotated each other. A similar example is given in Fig. 6(b), where the scale factor was estimated equal to $\hat{k} = 1.25$. One can verify that in both examples the estimated rotation angles are close to the original ones, verifying the effectiveness of the algorithm. This was the case for all tested dance pairs.

C. Local temporal synchronization

In a strict dance evaluation scenario, where a common audio sample is considered, a perfect dance performance should not deviate locally in phase from the "gold" one. In a looser scenario however, small deviations in the dance speed, i.e. small amounts of local de-synchronization, may be allowed and therefore should not penalize the evaluation scores. On the other hand, large local phase differences should always not be allowed. To add such a flexibility in our approach, we use Dynamic time warping (DTW) [28] to locally synchronize two dance signals. DTW is a robust approach for measuring distance between time series, allowing similar shapes to match even if they are out of phase in the time axis. For details the reader is referred [28]. Here we formulate the DTW approach for our application.

In our case, we consider two pure quaternionic matrices $\mathbf{P}(j,t)$ and $\mathbf{Q}(j,t)$, $j = 1,\ldots,J$, $t = 0,\ldots,T-1$, as in (14). To locally synchronize the sequences, we construct a $T \times T$ matrix, where the element (t_1,t_2) of the matrix contains the distance:

$$d(t_1,t_2) = \frac{1}{J}\sum_{j=1}^{J}|\mathbf{P}(j,t_1) - \mathbf{Q}(j,t_2)|^2. \quad (18)$$

This is a distance metric that takes into account all joints. As can be understood, this metric makes use of the distance between pure quaternions and therefore could be written using vectors notation. However, notice that at this point the dances have been already spatially aligned using the method of subsection V-B.

Consider a warping path $\mathcal{W}$ that introduces a temporal mapping between $\mathbf{P}(j,t)$ and $\mathbf{Q}(j,t)$, with the k-th element of $\mathcal{W}$ notated as $w^k = (t_1^k, t_2^k)$. Each candidate path $\mathcal{W}$ satisfies a set of conditions: a) The boundary condition $w^1 = (0,0)$, b) Continuity, i.e. the allowable steps in the warping path are restricted to adjacent cells and c) Monotonicity, i.e. $t_1^k \geq t_1^{k-1}$ and $t_2^k \geq t_2^{k-1}$. We search for the warping path that introduces the minimum total distance cost. Therefore, we calculate the cumulative distance matrix (see Fig. 7(c)) and search for the optimal path using dynamic programming, i.e. by following the recursion:

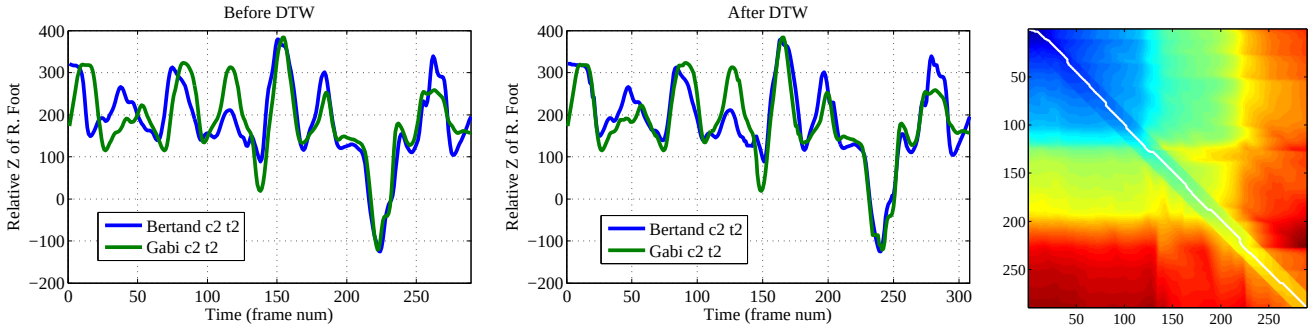
$$\gamma(t_1,t_2) = d(t_1,t_2) + \min\{\gamma(t_1-1,t_2-1)\gamma(t_1-1,t_2),\gamma(t_1,t_2-1)\}. \quad (19)$$

In order to prevent pathological warping paths, where a relatively small time interval in $\mathbf{P}(j,t)$ maps onto a relatively large time interval in $\mathbf{Q}(j,t)$ or vice versa, we use global wrapping constraints [28]. Actually, since we should not penalize only small amounts of local de-synchronization, we allow the wrapping path to lie inside a relatively narrow band along the diagonal (the Sakoe-Chiba band) of width equal to ± 15 frames. The specific wrapping path constraint is realized by setting the distance $d(t_1,t_2) \leftarrow \infty$ for $|t_1 - t_2| > 15$ in (18). This is practically the same to setting the cumulative distance $\gamma(t_1,t_2) \leftarrow \infty$ for $|t_1 - t_2| > 15$ in (19). The applied Sakoe-Chiba band is highlighted in Fig. 7(c). An example of local synchronization is shown in Fig. 7(a),(b).

DTW with hierarchical angular representations:
Since the hierarchical angular skeleton data representation of [17] is also evaluated in this paper, we use DTW with this representation as well. The employed approach is exactly the same, with the difference that the distance metric of (18) is replaced by a robust distance metric, as proposed in [17]. Let $F_1(i,t)$ and $F_2(i,t)$, $i = 1,\ldots,19$ denote the i-th feature at time t for the two dances to be compared, respectively. Then, the distance metric is:

$$d_{\text{HIE}}(t_1,t_2) = \frac{1}{19}\sum_{i=1}^{19} d_{\text{r}}^2\{F_1(i,t_1),F_2(i,t_2)\}, \quad (20)$$

where $d_{\text{r}}\{x,y\} = \min\{|x-y|,\delta\}$. With the use of the threshold δ, the effect of outliers is minimized. Distances



(a) Before DTW (b) After DTW (c) DTW - Cumulative distance matrix and optimal path

Fig. 7. The relative Z position of left foot for Bertand c2 t2 and Gabi c2 t2, (a) before and (b) after local synchronization via DTW. (c) The corresponding cumulative distance matrix and the optimum path. The Sakoe-Chiba band along the diagonal, with width ± 15 frames, is highlighted by multiplying the values outside the band by 1.1.

larger than δ do not introduce high penalties. In [17], the appropriate selection of δ is not discussed. After a lot of trial-and-error experiments in order to obtain meaningful results, we selected $\delta = 35^o$ in the experimental section.

VI. Automatic dancers' evaluation

In this section we address the evaluation of a dancer, compared to the "gold" dance of a professional. It is assumed that the dances to be compared have already been globally synchronized and spatially aligned (if necessary) using the methodologies of subsections V-A and V-B, respectively. Additionally, if small amounts of local de-synchronization are to be allowed, the local temporal synchronization via DTW (subsection V-C) is considered to have been applied.

In this section, three different evaluation scores are proposed. These can be combined to produce an overall score, as explained in paragraphs VI-A4 and VII-C. Two of them are based on quaternionic signal processing, while the third one is inspired by the optical-flow literature [29]. Additionally, a score metric that is used with hierarchical angular representations [17] is discussed.

A. Proposed score metrics

1) Joints positions score: For each joint j, a score is extracted by considering the modulus of the quaternionic Correlation Coefficient (CC) between the relative position signals $\mathbf{p}_j(t)$ and $\mathbf{q}_j(t)$ of the two dancers:

$$S_{1,j} = |\mathrm{CC}_j^{\mathrm{pos}}| = \frac{|\mathrm{C}_j^{\mathrm{pos}}(0)|}{\sqrt{\sigma_j^{\mathbf{p}} \cdot \sigma_j^{\mathbf{q}}}}, \tag{21}$$

where $\mathrm{C}_j(0)$ is the quaternionic cross-covariance (see (6)) at zero lag ($\tau = 0$) and $\sigma_j^{\mathbf{p}}$, $\sigma_j^{\mathbf{q}}$ are the quaternionic auto-covariances at zero lag. Notice that the auto-covariances are scalar (real) numbers. Then, a total score is obtained as the weighted mean of the separate joint scores, i.e.

$$S_1(\mathbf{w}^{\mathrm{pos}}) = \frac{\sum_{j=1}^{J} w_j^{\mathrm{pos}} \cdot S_{1,j}}{\sum_{j=1}^{J} w_j^{\mathrm{pos}}}. \tag{22}$$

The weights $\mathbf{w}^{\mathrm{pos}} = \{w_j^{\mathrm{pos}}\}, j = 1, 2, \ldots, J$ may be selected equal to unity (uniform weights), or may be selected heuristically, based on the significance of each joint in dancing performance and/or the associated signal-to-noise ratio (SNR). However, optimal weights can be found using a small training set and following an optimization procedure. This approach is discussed later in subsection VII-C.

2) Joints velocities score: It is straightforward to define a score based on relative joint velocities, instead of their positions, by taking the quaternionic CC for the relative joint velocity signals. Let the score for each joint be denoted as $S_{2,j} = |\mathrm{CC}_j^{\mathrm{vel}}|$ (calculated similarly to (21)) and the all-joint scores be denoted as $S_2(\mathbf{w}^{\mathrm{vel}})$.

3) 3D flow error-based score: For each frame, the relative velocities of the joints can be seen as 3D motion (flow) vectors. At this point, we drop the time-variable t, for simplicity. Inspired by the 2D optical flow literature [29], we consider the normalized (unit) 3D velocity vectors in homogenous coordinates, i.e.

$$\vec{s}(\vec{v_j}) = \frac{[Vx_j, Vy_j, Vz_j, 1]^{\mathrm{T}}}{\sqrt{|\vec{v_j}|^2 + 1}}. \tag{23}$$

The idea is that apart from the 3D displacement-per-frame representation, the velocity may be written as a unit 4D space-time vector. The unit 4D space-time vector $\vec{s}(\vec{v_j})$ contains information for both the speed and the direction of the 3D motion. Therefore, using such a 4D representation, one can introduce convenient error or similarity measures that handle large and very small speeds without introducing any amplification/bias that would be inherent in 3D vector-based measures (e.g. 3D vector differences or vector inner products).

Let for our application, the superscripts "p" and "a" stand for the "professional" (ground-truth) and the "amateur" dancer, respectively. In the computer vision literature [29], the inner product $\vec{s}(\vec{v_j}^{\,p}) \circ \vec{s}(\vec{v_j}^{\,a})$ is used in the definition of the flow Angular Error (AE) between $\vec{s}(\vec{v_j}^{\,p})$ and $\vec{s}(\vec{v_j}^{\,a})$. The inverse cosine of this gives the flow AE, which is expressed in degrees and is used as an evaluation metric for the difference (error) of an estimated flow vector to the reference one (ground-truth). Using the inverse cosine, the closer the estimated flow to the ground-truth, the smaller the AE is. In our application

however, we want to define a similarity measure, i.e. the closer a 3D flow vector to the reference one, the higher the corresponding (score) measure should be. Therefore, we drop the inverse cosine, which is a non-linear decreasing function, and we adopt a score metric as follows: Since the inner product of two unit vectors is a number in the range $[-1, 1]$, we define

$$S_{3,j} = \frac{1}{2}\Big(\vec{s}(\vec{v_j}^{\,p}) \circ \vec{s}(\vec{v_j}^{\,q}) + 1\Big), \tag{24}$$

in order to have a score metric in the range $[0, 1]$.

For a given frame t, as an all-joints score we consider the median along j, $S_3(t) = \text{median}_j\{S_{3,j}(t)\}$, in order to reject outliers, namely very wrong estimates due to skeleton tracking inaccuracies. This way, we account that significant differences between the professional's (reference) flow field and the amateur's one may arise due to inaccurate skeletal tracking and not due to actual differences in the dancing performance. A total score for the whole choreography can then be calculated as the mean or the median along t. The median-based score metric $S_3 = \text{median}_t\{S_3(t)\}$ was used in the experimental results provided in section VIII.

Notice that the score matric defined previously, does not make use of quaternions and lacks rotation invariance. However, at this point we assume that the dances have been already spatially aligned using the method of subsection V-B.

4) Combined score: Having defined three different score metrics S_1, S_2 and S_3, a combined score can be computed as the weighted mean:

$$S(\mathbf{m}, \mathbf{w}^{\text{pos}}, \mathbf{w}^{\text{vel}}) = \frac{m_1 \cdot S_1(\mathbf{w}^{\text{pos}}) + m_2 \cdot S_2(\mathbf{w}^{\text{vel}}) + m_3 \cdot S_3}{m_1 + m_2 + m_3}, \tag{25}$$

where $\mathbf{m} = \{m_1, m_2, m_3\}$. The estimation of the optimum weights $\mathbf{m}$, as well as the calculation of appropriate weights $\mathbf{w}^{\text{pos}}$ and $\mathbf{w}^{\text{vel}}$ is discussed in section VII.

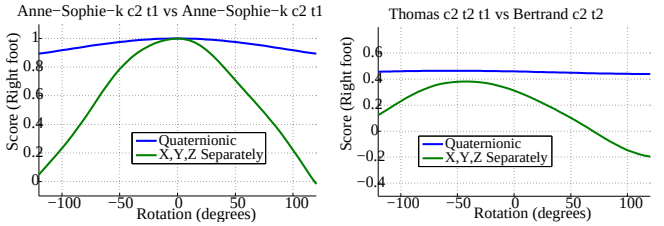
5) Score metrics with hierarchical angular representations: With the hierarchical angular representations, which are also evaluated in section VIII, we use a score metric similar to the one proposed in [17]:

$$S_{\text{HIE}} = \frac{1}{19}\sum_{i=1}^{19} s_i, \quad s_i = \frac{1}{T}\sum_{t=0}^{T-1} e^{-\left(\frac{d_r\{F_1(i,t), F_2(i,t)\}}{\sigma}\right)^4} \tag{26}$$

where σ is a parameter that controls the allowed amount of deviation from the expert's performance and $d_r\{x, y\}$ is the robust distance metric given in (20). The selection of appropriate value for σ is not discussed in [17]. After trial-and-error, we selected $\sigma = 30^o$ for our experiments.

B. Quaternionic scores' invariance to rigid transformations

One of the ideas behind using a quaternionic approach (for scores S_1 and S_2) in the presented application is that one can handle the three coordinate variables X, Y and Z jointly. This has many benefits, such as (approximate) invariance to rigid transformations, i.e. translation, scaling and more importantly rotation. A proof of the quaternionic score's invariance to rigid transformation is provided in Appendix B. Here, a demonstration example is given. For comparison purposes, let us consider the case that the X, Y, Z components of the relative position signals are handled separately, providing three separate scores that can be combined via a mean operation (this would be similar to using a correlation metric that makes use of the 3D vector inner product). It is almost evident that the "X, Y, Z separately" approach is not rotation-invariant, if one simply thinks that a rotation through an angle of 90^o about the Y (vertical) axis results into interchanging the roles of X and Z. On the other hand, the quaternionic evaluation approach is invariant to rigid transformations, as demonstrated by the diagrams of Fig. 8. For simplicity, only the score $S_{1,j}, j = 15$ (right foot) is calculated and presented. Notice that (approximate) invariance of the single joint's score, suggests an even stronger invariance for the "all-joints" score. In Fig. 8(a), a dance performance of Anne-Sophie-k is evaluated against versions of the same performance, that are scaled by a factor equal to 1.4 and globally rotated about the Y (vertical) axis. The calculated score is almost independent to the rotation angle, remaining very close to unity, as expected. On the other hand the "X, Y, Z separately"-score reduces as the rotation angle increases, as also expected. In Fig. 8(b), the performance "Thomas c2 t2" is evaluated against "Bertand c2 t2". The same set of rigid transformations is considered. The quaternionic score remains almost constant, independently to the rotation angle.



(a) (b)

Fig. 8. Approximate invariance to rigid transformations of the quaternions-based score. (a) A dance of Anne-Sophie-k is evaluated against the same dance sequence, rotated about the Y axis and scaled by 1.4. (b) The performance "Thomas c2 t2" is evaluated against "Bertand c2 t2". The same set of rigid transformations is considered. The score $S_{1,j}, j = 15$ (right foot) with respect to the rotation angle is presented.

C. Additional issues

1) Instantaneous scores: The methodologies described throughout the current section can be slightly modified, in order to produce instantaneous scores. In a virtual dance-class scenario, the calculation of instantaneous scores can serve to highlight the time intervals (i.e. choreography parts) in which the dancer's performance is poor and requires improvement. The extension of the methodologies is straightforward: With respect to the total scores S_1 and S_2, instead of considering the relative joint position and velocity signals $\mathbf{p}_i(t)$ or $\mathbf{v}_i(t)$ for the whole time interval $t \in [0, T-1]$, one can use a time-sliding gaussian window around t of length L. The instantaneous scores $S_1(t)$ and $S_2(t)$ around a time instance t_0 are calculated by applying locally the methodology, i.e. by considering the varying parts of $\mathbf{p}_i(t)$ or $\mathbf{v}_i(t)$ in $t \in [t_0 - L/2, t_0 + L/2]$, multiplying them with the gaussian window and calculating the quaternionic correlation coefficient.

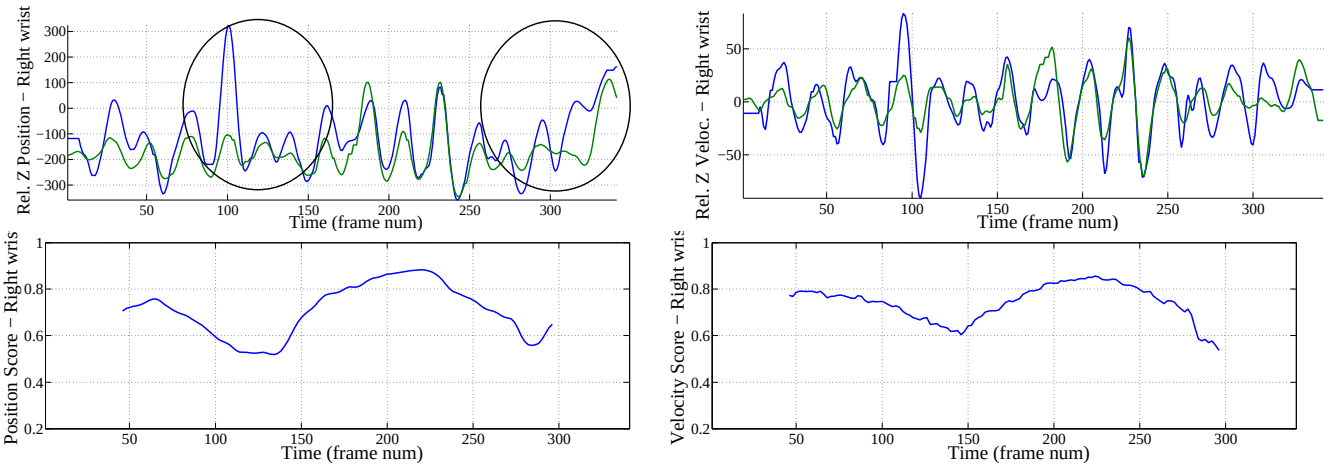


Fig. 9. Instantaneous scores for Jacky c3 t2 vs Bertrand c3 t2 - Right wrist. Left: The Relative positions for right wrist and the corresponding position-based instantaneous score; Right: Relative velocities for right wrist and the corresponding velocities-based instantaneous score. The instantaneous scores remain high almost everywhere, expect for two time intervals where Jacky's performance (with respect to the right wrist) gets worse.

The length L and the parameter σ of the gaussian (defined as in (10), but with t in place of u) have to be adequately large, so that the signals' statistics in a specific time-interval to be adequately rich. In our application, taking into account the Kinect frame rate of 30fps, a reasonable choice for L is 90 frames (3 sec) and for σ is 1. Going into details and presenting extensive experimental results is beyond the scope of this paper. However, experimental results are given in Fig. 9 which demonstrate the effectiveness of the discussed approach. Notice that both position- and velocity-based scores present valleys at time intervals where the dancer'a performance gets worse.

2) Separate scores for the upper and lower body parts: The idea of extracting a total score for the whole dancer's body, in order to identify whether he/she performs well, can be extended in order to provide feedback to the amateur on how he/she can improve the performance. The extension is based on the idea that separate scores can be extracted for different body parts, for example the upper and the lower body parts. This is accomplished by simply considering separately the upper and lower body joints and applying separately the proposed methodologies.

VII. Objective evaluation measures and weights optimization

A. Ground-truth data

The exploited dance dataset (Huawei 3DLife/EMC^2 Grand Challenge 2011) contains "ground-truth" evaluation ratings, provided by Salsa dance experts. For each dance performance, the experts provided ratings as integer scores between 1 and 5 (poor to excellent), over different evaluation axes, such as "Upper-Body Fluidity (UBF)", "Lower-Body Fluidity (LBF)", "Musical Timing (MT)" and "Choreography (CH)". Although mapping to human-based dance evaluation criteria is a difficult task, the proposed evaluation scores are related to all the above criteria: The proposed scores are correlation-based metrics and therefore, they constitute a measure of "similarity" of the "dancing signals" being compared and the degree to which they are synchronized to each other. Assuming that the ground-truth ratings for the reference dance is "excellent", it is essential to consider that the presented automatically extracted scores reflect "CH" (accuracy in executing a specific sequence of dance steps) and "BF" performance, as well as "MT".

In Table I we provide the available "ground-truth" scores for choreographes c1 to c4, which are used in our experimental results. In the last table column, the Average score is given, obtained from the four different scores. The row tables are sorted based on this Average score. For the same Average score, sorting is based primarily on CH and secondly on MT. The ranking of the dance performances in Table I constitutes for us the ground-truth ranking, used throughout the rest of the paper.

B. Objective evaluation measure

At this point we want to define an appropriate objective measure that can be used for the evaluation of the proposed automatically extracted scores. The mean squared difference of the computed scores from the ground-truth Average scores could be selected as such a measure. However, since many factors are involved in the human-based objective evaluation, our will is not to have automatically extracted scores that are as close as possible to the ground-truth scores. Instead, we want the automatic score-based ranking of the dancers to be as close as possible to the ground-truth-based ranking of Table I.

The number of "swapped pairs" [30] can be used as an appropriate ranking-based objective measure. For N dances, the total number of dance pairs is $N_p = \binom{N}{2}$. Given the ground-truth ordering (ranking) $\mathcal{R}^{\text{gt}}$ and the ordering $\mathcal{R}(\mathbf{w})$ for a given selection of parameters $\mathbf{w}$, an ordering inversion for a specific pair of dances occurs when $\mathcal{R}^{\text{gt}}$ and $\mathcal{R}(\mathbf{w})$ disagree about the ordering of these two dances (discordant pair). Let $Q(\mathbf{w})$ be the total number of ordering inversions introduced by $\mathcal{R}(\mathbf{w})$ and $P(\mathbf{w}) = N_p - Q(\mathbf{w})$ be the number of concordant pairs (not inversions). Therefore, an objective evaluation function (to be minimum) can be defined as the

number of ordering inversions $Q(\mathbf{w})$ over the total number of pairs N_p:

$$F(\mathbf{w}) := \frac{Q(\mathbf{w})}{N_p} = 1 - \frac{\tau\{\mathcal{R}(\mathbf{w}); \mathcal{R}^{\text{gt}}\}}{2}, \tag{27}$$

where $\tau\{\mathcal{R}(\mathbf{w}); \mathcal{R}^{\text{gt}}\} := \frac{P(\mathbf{w})-Q(\mathbf{w})}{P(\mathbf{w})+Q(\mathbf{w})}$ is an adapted version of Kendall's τ function [30]. The objective function F lies by definition in $[0, 1]$. It can also be easily verified that F becomes equal to 0.50 for "dummy" scores that produce random ranking.

In Tables II and III, the objective measure F is notated as "Obj." and is given in percentage values %.

C. Weights optimization via PSO

The aim is to find the set of weights $\mathbf{w} = \mathbf{w}^{\text{pos}}$ for the position-based score $S_1(\mathbf{w}^{\text{pos}})$ in (22) that minimizes the objective function in (27), using an appropriate training set with known ground-truth. As a training set, the choreographies in sets c1 and c3 are used.

A variety of intelligent computing algorithms can be adopted to calculate the optimal weights. In this work, the Particle Sworm Optimization (PSO) method [31] was selected. PSO is a global optimization algorithm, which is based on a population-based stochastic-search approach to find good solutions with regard to a given measure of quality (fitness function, which in our case is the function in (27)).

There are some important issues to be addressed here. The fitness function in (27) is a quite complicated function with significant non-linearities, because it involves a sorting operation (ranking). It also contains a large number of local minima. Therefore, a stochastic-search method, such as the PSO, seems to be appropriate, in order to avoid trapping in bad local minima and finding a good one. Secondly, since the objective function involves a sorting operation, its minima correspond to flat regions (subspaces) of the multi-dimensional search space. Namely, two neighbor points in the search space may result into the same value of the objective function. Additionally, the training set is very small to produce a good and robust solution for the optimization problem. Only 9 recordings are available in the training set c1 and c3, a quite small number compared to the total number of weights $J = 15$.

For all these reasons, some simplifications have to be made and heuristic information has to be embedded in the optimization process: It is evident that there is not any specific reason for the weights corresponding to the left-body joints to be different from the weights that correspond to the right-body joints. Additionally, the motion of legs and arms contain much different pieces of dance information than e.g. those of the head or the neck. Finally, joints near the torso (neck, shoulders, hips, as well as neck and head), 1st order joints (knees and elbows) and 2nd order joints (wrists and feet) are assumed to present different SNR characteristics. Therefore, the body joints were separated into six groups: Torso $\mathcal{T}$ (global motion), Elbows $\mathcal{E}$, Knees $\mathcal{K}$, Wrists $\mathcal{W}$, Feet $\mathcal{F}$ and Other $\mathcal{O}$ (joints near the torso, neck and head). The weights within each of these groups were set equal to each other during the optimization process, i.e. $w_i = w_j = w_{\mathcal{O}},\ \forall i, j \in J_{\mathcal{O}}$, etc. In other words, the dimensionality of the search space is reduced to $J_1 = 6$.

In our problem, each PSO particle of the population corresponds to a set of weights $\mathbf{w} = \{w_j\}, j = 1, 2, \ldots, J_1$. We allow the population of particles to take values in the range $\mathbb{S} = [0.2, 0.8]^{J_1}$ (search space). In our experiments we use totally $I = 1000$ particles. We also allow particle velocity to be in the range $[-0.5, 0.5]^{J_1}$. Finally, the maximum number of iterations in the PSO optimization procedure was set equal to $K = 100$.

The PSOT Toolbox[5] for MATLAB was used for weights' optimization. Running multiple times the PSO algorithm, the weights shown in the diagrams of Fig. 10(a) were obtained. As mentioned, a specific minimum of the objective function (even the global one) does not correspond to a single point of the search-space and therefore it is essential that the obtained weights in different trials are not exactly the same. However, they are similar and most of them seem to correspond to the same flat minimum-region of the search-space. Therefore, during the evaluation phase (see experimental results) the mean values of the per-trial weights were used. These are given in the diagram of Fig. 10(a).

The following conclusions can be drawn: For the position-based score $S_1(\mathbf{w}^{\text{pos}})$, the global motion (torso) and 1st and 2nd order joints are more-or-less equally significant and definitely more significant than other joints. The relatively small value for the weight $w_{\mathcal{O}}$ makes sense if one thinks that head and neck are not significant in dance, while joints near the torso (shoulders and hips) do not induce strong motion signals (relatively to torso) and thus they are probably characterised by lower SNR values.

Exactly the same method, with the same parameter values, was used to find the set of weights $\mathbf{w} = \mathbf{w}^{\text{vel}}$ for the velocity-based score $S_2(\mathbf{w}^{\text{vel}})$. The results are presented in Fig. 10(b). Similar conclusions can be drawn, with the difference that the global motion (torso) is not significant here.

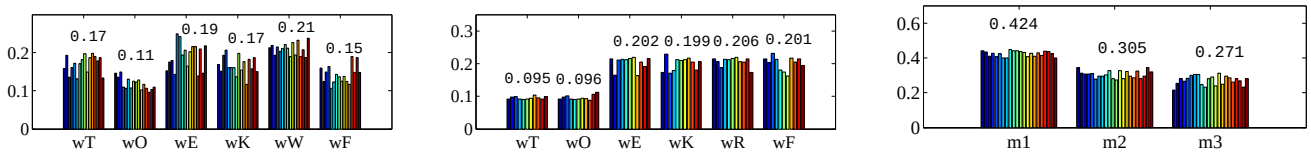
Finally, given the selected set of weights $\mathbf{w}^{\text{pos}}$ and $\mathbf{w}^{\text{vel}}$, the optimum set of weights $\mathbf{m} = \{m_1, m_2, m_3\}$ was found for the combined score of (25). The same methodology was applied, with the same set of parameters. The difference is that the search space here is the 3-dimensional space $[0.2, 0.8]^3$. The results are presented in 10(c).

Notice that the weights $\mathbf{w}^{\text{pos}}$, $\mathbf{w}^{\text{vel}}$ and $\mathbf{m}$ should ideally be found jointly in a single optimization process. However, as explained this seems to be impossible due to a) the non-linearities of the objective function, b) the high dimensionality of the problem and c) the small size of the training set. However, even with the sequential optimization approach that we employed, the combined score is significantly improved compared to the separate scores, as can be verified in the experimental section.

VIII. Experimental results

The automatic evaluation of something as subjective as dancer performance is obviously an extremely difficult task.

[5] http://www.mathworks.com/matlabcentral/fileexchange/7506



(a) (b) (c)

Fig. 10. The sets of weights obtained from multiple PSO optimization trials. For each weight, one bar corresponds to a different trial. The weights after each trial were normalized to sum up to unity. a) The joint-groups weights for position scores $w_{\mathcal{T}}$, $w_{\mathcal{O}}$, $w_{\mathcal{E}}$, $w_{\mathcal{K}}$, $w_{\mathcal{W}}$ and $w_{\mathcal{F}}$; b) The corresponding joint-group weights for velocities scores; (c) The score-metric weights m_1, m_2 and m_3. In each graph, the weights used in the evaluation experiments (obtained from the mean of all trials) are also presented.

TABLE II
AUTOMATICALLY EXTRACTED SCORES (UNIFORM WEIGHTS USED) AND THE CORRESPONDING VALUES OF OBJECTIVE FUNCTIONS.

GT RANK	Without local synchr. (DTW)					With local synchr. (DTW)				
	S_{POS}	S_{VEL}	S_{AE}	S_{COM}	S_{HIE}	S_{POS}	S_{VEL}	S_{AE}	S_{COM}	S_{HIE}
1	0.578	0.482	0.821	0.627	0.804	0.623	0.539	0.850	0.671	0.817
2	0.740	0.538	0.884	0.721	0.865	0.817	0.664	0.917	0.799	0.880
3	0.659	0.284	0.747	0.563	0.793	0.694	0.341	0.815	0.617	0.808
4	0.453	0.293	0.622	0.456	0.677	0.470	0.323	0.679	0.491	0.691
5	0.338	0.121	0.722	0.393	0.869	0.431	0.135	0.745	0.437	0.870
6	0.555	0.354	0.624	0.511	0.772	0.663	0.458	0.750	0.624	0.780
7	0.483	0.376	0.740	0.533	0.863	0.522	0.420	0.766	0.569	0.868
8	0.288	0.250	0.625	0.387	0.806	0.421	0.443	0.765	0.543	0.834
9	0.264	0.116	0.630	0.337	0.739	0.341	0.181	0.662	0.394	0.745
10	0.657	0.349	0.688	0.565	0.727	0.678	0.422	0.761	0.620	0.736
11	0.339	0.173	0.630	0.381	0.716	0.430	0.368	0.739	0.512	0.734
12	0.404	0.204	0.703	0.437	0.687	0.457	0.297	0.695	0.483	0.707
13	0.328	0.167	0.601	0.365	0.722	0.521	0.401	0.750	0.557	0.743
14	0.380	0.246	0.668	0.432	0.721	0.464	0.378	0.738	0.526	0.736
15	0.392	0.235	0.563	0.397	0.733	0.387	0.234	0.600	0.407	0.737
16	0.293	0.201	0.617	0.370	0.729	0.348	0.252	0.664	0.422	0.736
17	0.206	0.133	0.546	0.295	0.691	0.320	0.310	0.713	0.448	0.709
Obj %	25.7	27.9	**24.3**	**22.8**	27.9	**24.3**	33.1	**25.0**	27.2	26.5

TABLE III
AUTOMATICALLY EXTRACTED SCORES USING OPTIMUM WEIGHTS AND THE CORRESPONDING VALUES OF OBJECTIVE FUNCTIONS.

GT RANK	S_{POS}	S_{VEL}	S_{COM}
1	0.601	0.497	0.629
2	0.745	0.549	0.722
3	0.668	0.328	0.586
4	0.446	0.277	0.442
5	0.354	0.145	0.390
6	0.540	0.327	0.498
7	0.492	0.394	0.530
8	0.295	0.253	0.372
9	0.265	0.131	0.323
10	0.637	0.343	0.561
11	0.344	0.169	0.368
12	0.399	0.207	0.423
13	0.334	0.154	0.351
14	0.384	0.241	0.417
15	0.388	0.203	0.379
16	0.288	0.195	0.349
17	0.211	0.128	0.277
Obj. %	24.3	25.0	**20.5**

TABLE I
GROUND-TRUTH RANKING POSITIONS, BASED ON THE SCORES PROVIDED BY EXPERTS.

GT RANK	CHOREOGRAPHY	UBF	LBF	MT	CH	AVERAGE
1	bertrand c3 t1	5	5	5	5	5.00
2	anne-sophie-k c2 t2	5	5	5	5	5.00
3	bertrand c4 t3	4	5	5	5	4.75
4	habib c2 t3	5	5	5	4	4.75
5	habib c1 t2	4	4	5	5	4.50
6	jacky c2 t1	3	4	5	5	4.25
7	jacky c1 t1	3	4	5	5	4.25
8	thomas c1 t2	4	4	4	5	4.25
9	ming-li c1 t2	3	4	4	5	4.00
10	habib c4 t2	4	3	5	4	4.00
11	habib c3 t2	4	3	5	4	4.00
12	thomas c2 t2	3	3	4	5	3.75
13	jacky c3 t2	4	3	4	4	3.75
14	habib c3 t1	5	3	5	2	3.75
15	ming-li c2 t1	3	3	4	4	3.50
16	ming-li c3 t2	3	3	4	2	3.00
17	thomas c3 t1	3	2	3	2	2.50

However, according to our experimental results, the presented methodologies do produce meaningful results, in general. For example, a) When comparing two different captures of a professional dancer for the same choreography, the computed scores are relatively high (see the 1st, 2nd and 3rd rows of Tables I, II and III, which correspond to professional dancers). Obtaining a high score in this case is essential, since a professional dancer is able to perform almost identical dancing movements in two different captures (see Fig. 2(a)); b) The automatic scores-based ranking does not deviate significantly from the one produced using the ground-truth ratings. This is discussed in detail in the next paragraphs; c) As demonstrated in paragraph VI-C1 and Fig. 9, the instantaneous scores $S_1(t)$, $S_2(t)$ show a similar behavior, presenting valleys at time intervals where the dancer's performance is poor.

A. Scores using uniform weights

A first set of experimental results is given in Table II. The rows of the table are shorted according to the ground-truth ranking of Table I. Therefore, the scores should ideally be decreasing along each column. More specifically, Table II presents the obtained relative position-based score (column "S_{POS}"), the velocity-based score ("S_{VEL}"), the AE-based score ("S_{AE}") of paragraph VI-A3, the combined score ("S_{COM}") of paragraph VI-A4, as well as the score of paragraph VI-A5, which is based on hierarchical angular representations ("S_{HIE}"). All scores are given both without and with the local synchronization method (via DTW) employed. For each set of scores, the objective evaluation function F is finally given (row "Obj %"). The experimental results presented in this Table refer to uniform (all equal to unity) weights $\mathbf{w}^{\text{pos}}$, $\mathbf{w}^{\text{vel}}$ and $\mathbf{m}$.

There are a number of conclusions that can be drawn:

C1) In all cases, local synchronization via DTW leads to higher scores. This makes sense, since local phase differences, which otherwise would penalize the scores, are compensated

by DTW. Notice however that in the considered dance evaluation scenario (Huawei 3DLife/EMC^2 grand challenge 2011) the dancers have to remain synchronized with the common audio sample and thus to each other. The ground-truth data were given for that scenario. Therefore, the use of DTW does not improve the performance of the whole automatic evaluation pipeline. This can be verified from the values of the objective function F ("Obj %") with and without DTW. The objective function F goes worse for all kind of scores, except for scores S_{POS} and S_{HIE}, where a slight improvement is observed. In an application scenario where good synchronization of the actors is not a significant issue, DTW would be of great importance. For example consider the scenario, where the objective is to execute a sequence of dance steps without taking into account the speed of execution. The automatic evaluation pipeline should not penalize the user if her/his dance is slower or faster in some intervals. The application of DTW in this case, would compensate the local phase mismatches and the automatically generated scores would be high, as they should be.

C2) Without local synchronization, according to values of the objective functions, the best performance is achieved by S_{AE} and S_{COM}. With local synchronization, the best performance is achieved by S_{POS} and S_{AE}. The relatively good performance of S_{AE} could be partially explained by the median operation-based rejection of outliers (see section VI-A3). Notice however that the weights for all other scores were set equal to unity for the specific set of experiments. When the appropriate set of weights is used (see next paragraph) the performance of these scores is improved.

C3) The hierarchical angular representation-based score ("S_{HIE}") performs quite well according to the values of the objective functions. However, it does not generally outperform the proposed scores.

B. Scores using optimum weights

A second set of experimental results is given in Table III. Specifically, Table III presents the obtained relative position-based score (column "S_{POS}"), the velocity-based score ("S_{VEL}") and the combined score ("S_{COM}"), using the weights found according to the method of subsection VII-C. Additionally, all presented scores were extracted without applying local synchronization via DTW.

The general conclusions that can be drawn are that a) the weighted position- and velocity-based scores outperform all scores that do not use weights (or use uniform weights), presented in Table II; b) The combined score ("S_{COM}"), with the use of appropriate weights outperforms all other scores.

IX. Conclusions

A novel methodology for the automatic evaluation of dance performances was presented. The method is based on motion acquisition via Kinect human skeleton tracking and the application of appropriate quaternionic signal-processing techniques for a) temporally synchronizing, b) spatially aligning and c) "comparing" two vector "dance signals". The proposed quaternionic framework for MoCap dance signal analysis presents some advantages, enabling for example the definition of metrics that are invariant to rigid transformation and/or the formulation of fast algorithms for the estimation of the transformation parameters. The presented dance evaluation results are promising and verify the effectiveness of the proposed approach. We also presented experimental results using hierarchical angular representations of skeleton data, which can be found in the relevant literature [17], but have not been previously extensively tested in a dance evaluation task, such as the one addressed in this paper.

This work is used to support the realization of an online dance studio, where a dance class is provided by an expert and delivered to students via the web. Its adaption/extension to handle other similar physical-activity scenarios in tele-immersive environments is straightforward. Furthermore, although not straightforward, the introduced quaternions-based processing framework, endowed intrinsically with rigid-transformation invariance, could be adapted to the more general problem of human motion analysis.

Acknowledgment

The work presented in this paper was supported by the European Commission under contract FP7-601170 RePlay.

Appendix

A. Additional notes on quaternions

Given a pure quaternion $\mathbf{p}$, any quaternion q can be decomposed into a parallel (to $\mathbf{p}$) part and a perpendicular part, denoted as $\mathrm{q}_{\parallel}$ and $\mathrm{q}_{\perp}$, respectively.

Quaternion multiplication is generally non-commutative. However, parallel quaternions (quaternions with parallel vector parts) commute. Additionally, if $\mathbf{p}$ is a pure quaternion and q is a quaternion with $\mathcal{V}(\mathrm{q}) \perp \mathbf{p}$, then multiplication reordering is possible through: $\mathrm{q}\mathbf{p} = \mathbf{p}\overline{\mathrm{q}}$.

Unit quaternions provide a convenient mathematical notation for expressing rotations in the 3-D space. The rotation of a quaternion q about a 3D unit axis $\boldsymbol{\mu}$ (unit pure quaternion) and through an angle θ, is expressed as $\mathrm{R}\,\mathrm{q}\,\overline{\mathrm{R}}$, where $\mathrm{R} = e^{\boldsymbol{\mu}\frac{\theta}{2}}$.

B. Invariance to rigid transformations

The objectives here are to show that: a) The position $\tau_{\max}$ of the maximum of $|\mathrm{C}_{\text{total}}(\tau)|$ in (15), and thus the methodology in subsection V-A for the estimation of the global temporal shift, is invariant to rigid-body transformations (translation, scaling and rotation); b) The methodology in subsection V-B is valid, i.e. the global rotation (rotation axis and angle) is encoded in the phase of $\mathrm{C}_{\text{total}}(0)$; c) The adopted quaternionic score S_1 (see (21)) is invariant to rigid-body transformations. The same conclusions can then be drawn for score S_2.

Consider the quaternionic relative-position "dance" signal $\mathbf{p}(t)$ of the j-th joint and its transformed versions (we drop j for notational simplicity):

$$\begin{aligned} &\mathbf{p}_{\text{R}}(t) := \mathrm{R}\,\mathbf{p}(t)\,\overline{\mathrm{R}}, \quad \mathbf{p}_{\text{RS}}(t) := k \cdot \mathrm{R}\,\mathbf{p}(t)\,\overline{\mathrm{R}}, \\ &\mathbf{p}_{\text{RST}}(t) := k \cdot \mathrm{R}\,\mathbf{p}(t)\,\overline{\mathrm{R}} + \mathbf{d}, \end{aligned} \tag{28}$$

where $\mathrm{R} = e^{\boldsymbol{\mu}\frac{\theta}{2}}$, k a real-number scaling factor and $\mathbf{d}$ a pure quaternion. These versions correspond to only-rotation, rotation+scaling and rotation+ scaling+translation of $\mathbf{p}(t)$.

Showing that the methodologies are invariant to translation and scaling is almost straightforward, because quaternionic addition and scalar-with-quaternion multiplication do not present any differences from those in complex number theory. More specifically, by decomposing the signals $\mathbf{p}(t)$ and $\mathbf{p}_{\text{RST}}(t)$ into their varying and constant (centroid) parts, from (28) we have:

$$\mathbf{p}^v_{\text{RST}}(t) + \mathbf{p}^c_{\text{RST}} = k \cdot \text{R}\,[\mathbf{p}^v(t) + \mathbf{p}^c]\,\overline{\text{R}} + \mathbf{d} = k \cdot \text{R}\,\mathbf{p}^v(t)\,\overline{\text{R}} + \mathbf{d}_1, \tag{29}$$

where $\mathbf{d}_1 = \text{R}\,\mathbf{p}^c\,\overline{\text{R}} + \mathbf{d}$. Since the centroids of the quaternionic signals are subtracted (the cross-covariance is used in the calculation of score S_1 and the matrix in (14) contain only the varying parts), a spatial shift $\mathbf{d}$ does not affect the proposed methodologies. Therefore, from now on we consider that the quaternionic signals are zero-mean (zero-centroid) and therefore ignore any translation $\mathbf{d}$.

The invariance of the proposed quaternionic score S_1 (see (21)) to scaling can now be shown. Considering the auto-covariance (at zero lag) in the denominator of (21), we have:

$$\begin{aligned}\sigma_{\mathbf{p}_{\text{RS}}} &:= \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{p}_{\text{RS}}(t)\,\overline{\mathbf{p}_{\text{RS}}(t)} = \frac{1}{T}\sum_{t=0}^{T-1} k \cdot \mathbf{p}_{\text{R}}(t)\,\overline{k \cdot \mathbf{p}_{\text{R}}(t)} \\ &= k^2 \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{p}_{\text{R}}(t)\,\overline{\mathbf{p}_{\text{R}}(t)} = k^2\,\sigma_{\mathbf{p}_{\text{R}}}. \end{aligned} \tag{30}$$

Similarly, for the nominator of (21), it holds:

$$\begin{aligned}\text{C}(0) &:= \text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{RS}}(t)\}(0) = \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{p}(t)\,\overline{\mathbf{p}_{\text{RS}}(t)} \\ &= \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{p}(t)\,\overline{k \cdot \mathbf{p}_{\text{R}}(t)} = k \cdot \text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0). \end{aligned} \tag{31}$$

Consequently, using (30) and (31), for the score S_1 in (21) one can conclude:

$$\begin{aligned}S_1\{\mathbf{p}(t), \mathbf{p}_{\text{RS}}(t)\} &:= \frac{\text{C}(0)}{\sqrt{\sigma_{\mathbf{p}} \cdot \sigma_{\mathbf{p}_{\text{RS}}}}} \\ &= \frac{k \cdot \text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0)}{k \cdot \sqrt{\sigma_{\mathbf{p}} \cdot \sigma_{\mathbf{p}_{\text{R}}}}} = S_1\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}. \end{aligned} \tag{32}$$

Therefore, the adopted score S_1 is invariant to scaling.

Similarly to (28), let us denote $\mathbf{P}_{\text{RS}}(j,t) = k \cdot \text{R}\,\mathbf{P}(j,t)\,\overline{\text{R}} = k \cdot \mathbf{P}_{\text{R}}(j,t)$. Now consider $\text{C}_{\text{total}}(\tau)$ in (15). We have:

$$\begin{aligned}\text{C}^{\text{RS}}_{\text{total}}(\tau) &= \frac{1}{J \cdot T}\sum_{j=1}^{J}\sum_{t=0}^{T-1} \mathbf{P}(j,t)\,\overline{\mathbf{P}_{\text{RS}}(j, t-\tau)} \\ &= k \cdot \frac{1}{J \cdot T}\sum_{j=1}^{J}\sum_{t=0}^{T-1} \mathbf{P}(j,t)\,\overline{\mathbf{P}_{\text{R}}(j, t-\tau)} = k \cdot \text{C}^{\text{R}}_{\text{total}}(\tau). \end{aligned} \tag{33}$$

Thus, scaling of a dance just introduces a scaling to $\text{C}_{\text{total}}(\tau)$. Therefore, the position $\tau_{\max}$ of its maximum does not change and the methodology in subsection V-A is invariant to scaling.

Now, we can show that the adopted quaternionic methodologies are approximately rotation-invariant. In order to proceed, we begin with the pathological case, in which the quaternionic signal $\mathbf{p}(t)$ is constant over time, i.e. $\mathbf{p}(t) = \mathbf{p}$. The signal

$$\mathbf{p}_{\text{R}}(t) = \text{R}\,\mathbf{p}\,\overline{\text{R}} = e^{\boldsymbol{\mu}\frac{\theta}{2}}\,\mathbf{p}\,e^{-\boldsymbol{\mu}\frac{\theta}{2}} \tag{34}$$

can be considered as an appropriately rotated version of $\mathbf{p}(t)$, given by a rotation through some angle θ, about an axis $\boldsymbol{\mu} \perp \mathbf{p}$. The quaternionic cross-covariance in this case equals:

$$\text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0) = \frac{1}{T} \cdot T \cdot \mathbf{p}\,\overline{e^{\boldsymbol{\mu}\frac{\theta}{2}}\,\mathbf{p}\,e^{-\boldsymbol{\mu}\frac{\theta}{2}}}, \tag{35}$$

which by removing the conjugate and canceling terms simplifies to:

$$\text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0) = -\mathbf{p}\,e^{\boldsymbol{\mu}\frac{\theta}{2}}\,\mathbf{p}\,e^{-\boldsymbol{\mu}\frac{\theta}{2}}. \tag{36}$$

Since $\boldsymbol{\mu} \perp \mathbf{p}$, it also holds $\mathcal{V}(e^{\boldsymbol{\mu}\frac{\theta}{2}}) \perp \mathbf{p}$. Therefore, the multiplication reordering rule $\text{q}\mathbf{p} = \mathbf{p}\overline{\text{q}}$ (see previous subsection) applies. Additionally, since $\mathbf{p}$ is a pure quaternion, it holds $\mathbf{p}\,\mathbf{p} = -|\mathbf{p}|^2$. Therefore:

$$\text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0) = -\mathbf{p}\,\mathbf{p}\,e^{-\boldsymbol{\mu}\frac{\theta}{2}}\,e^{-\boldsymbol{\mu}\frac{\theta}{2}} = |\mathbf{p}|^2 \cdot e^{-\boldsymbol{\mu}\theta}. \tag{37}$$

In this case $|\text{C}\{\mathbf{p}(t), \mathbf{p}_{\text{R}}(t)\}(0)| = |\mathbf{p}|^2$. Given that the modulus of the cross-covariance is used in the adopted score's definition, the proposed evaluation score S_1 is rotation-invariant. As for the general case, where the signals $\mathbf{p}(t)$ and $\mathbf{p}_{\text{R}}(t)$ are not constant, the rotation (phase) between individual signal samples is summed over the whole time interval $[0, T-1]$, making the score S_1 practically approximately invariant to rotation. This was also demonstrated in Fig. 8.

Now, consider the covariance

$$\text{C}^{\text{R}}_{\text{total}}(\tau) = \frac{1}{J \cdot T}\sum_{j=1}^{J}\sum_{t=0}^{T-1} \mathbf{P}(j,t)\,\overline{\mathbf{P}_{\text{R}}(j, t-\tau)}.$$

As previously, assume the pathodolgical case where $\mathbf{P}(j,t) = \mathbf{P}(j)$ is constant along t and follow exactly the same arguments. One can then conclude (we consider $\tau = 0$, without loss of generality) to:

$$\text{C}^{\text{R}}_{\text{total}}(0) = \left(\frac{1}{J}\sum_{j=1}^{J} |\mathbf{P}(j)|^2\right) \cdot e^{-\boldsymbol{\mu}\theta}, \tag{38}$$

similarly to (37), where $\boldsymbol{\mu}$ and θ are the rotation axis and angle. This means that rotation introduced just the phase factor $e^{-\boldsymbol{\mu}\theta}$ and thus $|\text{C}^{\text{R}}_{\text{total}}(0)| = |\text{C}_{\text{total}}(0)|$, meaning that the methodology in subsection V-A is invariant to rotation.

More importantly now, notice that $\text{C}^{\text{R}}_{\text{total}}(0)$, as any quaternion, can be written in its polar form:

$$\text{C}^{\text{R}}_{\text{total}}(0) = |\text{C}^{\text{R}}_{\text{total}}(0)| \cdot e^{\boldsymbol{\mu}_{\text{q}}\phi_{\text{q}}}, \tag{39}$$

where $\boldsymbol{\mu}_{\text{q}}$ and ϕ_{q} are its "eigen-axis" and "eigen-angle", respectively. Equating the right-hand sides of (38) and (39), we have:

$$\left(\frac{1}{J}\sum_{j=1}^{J} |\mathbf{P}(j)|^2\right) \cdot e^{-\boldsymbol{\mu}\theta} = |\text{C}^{\text{R}}_{\text{total}}(0)| \cdot e^{\boldsymbol{\mu}_{\text{q}}\phi_{\text{q}}}. \tag{40}$$

Consequently, the rotation axis $\boldsymbol{\mu}$ and angle θ can be estimated by calculating $\text{C}^{\text{R}}_{\text{total}}(0)$ and computing its eigen-axis $\boldsymbol{\mu}_{\text{q}}$ and eigen-angle ϕ_{q}, as proposed in subsection V-B. As for the

general case, where the signals $\mathbf{P}(j,t)$ are not constant along t, the rotation between individual signal samples is summed over the whole time interval $[0, T-1]$ and for all joints j. In this case, $\mathrm{C}^{\mathrm{R}}_{\mathrm{total}}(0)$ effectively encodes the global rotation transformation, as demonstrated in Fig. 6. Similar facts have been also been demonstrated in [13] for estimating the global color-space rotation between color images.

REFERENCES

[1] R. Vasudevan, G. Kurillo, E. Lobaton, T. Bernardin, O. Kreylos, R. Bajcsy, and K. Nahrstedt, "High quality visualization for geographically distributed 3D teleimmersive applications," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 573–584, June 2011.

[2] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, March 2003.

[3] K. Altun, B. Barshan, and O. Tuncel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, October 2010.

[4] L. A. Schwarz, D. Mateus, and N. Navab, "Recognizing multiple human activities and tracking full-body pose in unconstrained environments," *Pattern Recognition*, vol. 45, no. 1, pp. 11–23, January 2012.

[5] J. A. Ward, P. Lukowicz, G. Troester, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1553–1567, Oct. 2006.

[6] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, Jan. 2006.

[7] B. Zou, S. Chen, C. Shi, and U. M. Providence, "Automatic reconstruction of 3D human motion pose from uncalibrated monocular video sequences based on markerless human motion tracking," *Pattern Recognition*, vol. 42, no. 7, pp. 1559–1571, July 2009.

[8] I.-C. Chang and S.-Y. Lin, "3D human motion tracking based on a progressive particle filter," *Pattern Recognition*, vol. 43, no. 10, pp. 3621–3635, October 2010.

[9] R. Horaud, M. Niskanen, G. Dewaele, and E. Boyer, "Human motion tracking by registering an articulated surface to 3D points and normals," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 158–164, Jan. 2009.

[10] S. Izadi, R. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, and A. D. A. Fitzgibbon, "KinectFusion: Real-time dynamic 3D surface reconstruction and interaction," in *ACM SIGGRAPH*, Aug. 2011.

[11] M. Zollhoefer, M. Martinek, G. Greiner, M. Stamminger, and J. Suessmuth, "Automatic reconstruction of personalized avatars from 3D face scans," *Comp. Anim. Virtual Worlds*, vol. 22, no. 2-3, pp. 195–202, 2011.

[12] T. A. Ell and S. J. Sangwine, "Hypercomplex Fourier transforms of color images," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 22–35, Jan. 2007.

[13] C. E. Moxey, S. J. Sangwine, and T. A. Ell, "Hypercomplex correlation techniques for vector images," *IEEE Trans. on Signal Processing*, vol. 51, no. 7, pp. 1941–1953, 2003.

[14] D. S. Alexiadis and G. Sergiadis, "Estimation of motions in color image sequences using hypercomplex fourier transforms," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 168–187, Jan. 2009.

[15] ——, "Motion estimation, segmentation and separation, using hypercomplex phase correlation, clustering techniques and graph-based optimization," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 212–234, Feb. 2009.

[16] L.-Q. Guo and M. Zhu, "Quaternion Fourier–Mellin moments for color images," *Pattern Recognition*, vol. 44, no. 2, pp. 187–195, February 2011.

[17] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *ACM SIGGRAPH Eurographics Symposium on Computer Animation*, 2011, pp. 147–156.

[18] M. P. Johnson, "Exploiting quaternions to support expressive interactive character motion," PhD dissertation, Massachusetts Institute of Technology, 2003.

[19] E. Pennestri and P. Valentini, "Dual quaternions as a tool for rigid body motion analysis: A tutorial with an application to biomechanics," *Archive of Mechanical Engineering*, vol. LVII (2), pp. 187–205, Oct. 2010.

[20] J. Barbic, J. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proc. of Graphics Interface GI '04*, 2004, pp. 185–194.

[21] C. Li, S. Q. Zheng, and B. Prabhakaran, "Segmentation and recognition of motion streams by similarity search," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 3, pp. 1–26, 2007.

[22] J. K. T. Tang, J. C. P. Chan, and H. Leung, "Interactive dancing game with real-time recognition of continuous dance moves from 3D human motion capture," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication (ICUIMC '11)*, 2011.

[23] J. C. P. Chan, H. Leung, J. K. T. Tang, and T. Komura, "A virtual reality dance training system using motion capture technology," *IEEE Transactions on Learning Technologies*, vol. 4, pp. 187–195, 2011.

[24] L. Naveda and M. Leman, "Representation of samba dance gestures, using a multi-modal analysis approach," in *5th Int. Conf. on Enactive Interfaces*, 2008, pp. 68–74.

[25] S. Takaaki, N. Atsushi, and I. Katsushi, "Detecting dance motion structure through music analysis," in *6th IEEE Intern. Conf. on Automatic Face and Gesture Recognition*, 2004, pp. 857–862.

[26] A. Camurri, S. Hashimoto, M. Ricchetti, K. Suzuki, R. Trocca, and G. Volpe, "KANSEI analysis of movement in dance/music interactive systems," in *Proc. of International Conference of HUmanoid and RObot (HURO99), Japan*, Oct. 1999, pp. 9–14.

[27] W. R. Hamilton, *Elements of Quaternions, 2nd ed.* Longmans, Green and CO, 1901.

[28] E. Keogh, "Exact indexing of dynamic time warping," in *Proc. of the 28th International Conference on Very Large Data Bases*, Hong Kong, 2002, pp. 406–417.

[29] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. Journ. of Comp. Vision*, vol. 12, pp. 43–77, 1994.

[30] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

Dimitrios S. Alexiadis was born in Kozani, Greece, in 1978. He received the Diploma degree and the Ph.D. degree in Electrical and Computer Engineering from the Aristotle University of Thessaloniki (AUTh), Thessaloniki, Greece, in 2002 and 2009, respectively. Since 2002, he has been involved in several research projects and has been a Research and Teaching Assistant with the Telecommunications Laboratory, Department of Electrical and Computer Engineering. During 2010, he was a Research Assistant with the School of Science and Technology, International Hellenic University (IHU). During 2010 and 2011, Dr. Alexiadis served as a Full-Time Adjunct Lecturer at the Department of Electronics, Technical Education Institute, Thessaloniki. Since March 2011, he is a Postdoctoral Research Fellow at the Information Technologies Institute, CERTH, Greece. His main research interests include still and moving image processing, face recognition, 2-D and 3-D motion estimation, structure-from-motion and stereo/multi-view image processing.

Petros Daras (M07) was born in Athens, Greece, in 1974. He received the Diploma degree in electrical and computer engineering, the M.Sc. degree in medical informatics, and the Ph.D. degree in electrical and computer engineering, all from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1999, 2002, and 2005, respectively. He is a Researcher Grade B, at the Information Technologies Institute (ITI) of the Centre for Research and Technology-Hellas (CERTH). His main research interests include search, retrieval and recognition of 3-D objects, 3-D object processing, medical informatics applications, medical image processing, 3-D object watermarking, and bioinformatics. He serves as a reviewer/evaluator of European projects. Dr. Daras is a key member of the IEEE MMTC 3DRPC IG.