“Topos” toolkit for Pure Data: exploring the spatial features of dance gestures for interactive musical applications

Luiz Naveda
State University of Minas Gerais - UEMG
luiznaveda@gmail.com

Ivani Santana
Federal University of Bahia - UFBA
ivanisantana.mapad2@gmail.com

ABSTRACT

The dissemination of multimedia technologies in the information societies created an interesting scenario: the unprecedented access to a diverse combination of music, image, video and other media streams raised demands for more interactive and expressive multimodal experiences. How to support the demands for richer music-movement interactions? How to translate spatiotemporal qualities of human movement into relevant features for music making and sound design? In this paper we study the realtime interaction between choreographic movement in space and music, implemented by means of a collection of tools called *Topos*. The tools were developed in the Pure Data platform and provide a number of feature descriptions that help to map the quality of dance gestures in space to music and other media. The features are based concepts found in the literature of cognition and dance, which improves the computational representation of dance gestures in space. The concepts and techniques presented in the study introduce new problems and new possibilities for multimedia applications involving dance and music interaction.

1. INTRODUCTION

If *music* is in anyway related to movements and sense of movement in our bodies, *dance* is the frontside of such relationships in the culture. Evidences of this close relationship have been extensively described in the literature (e.g.: [1,2]) and reinforced by the parallelism of music and dance practices in Western and non-Western cultures. Modern linguistic, artistic and labor divisions also reinforce the disciplinary divisions of music and dance, which shape the cultures, technologies, media and information we consume.

However, the recent technological revolutions created an interesting momentum: The access to networks, sensors, motion capturing technologies, processing power and all sort of technological devices produced unprecedented access to music and dance media. In this scenario, the encounters of people and music take place in multimodal contexts that provide mixed and diverse experiences with sound, imagery and movement interactions (e.g.: media streaming services, games, movies, dance clubs). Consequently, these societies start demanding deeper interactional experiences that take into account the potential expressiveness of its basic modalities and the expressive relationships that can be placed in-between the modalities. At this point, information societies seem to share the same characteristics of other non-Western cultures where divisions between listening, seeing and moving are somewhat diffuse (or not relevant). How to develop more comprehensive and expressive relationships between the dance gestures and the musical ideas in the rise of a society that becomes intrinsically multimodal?

In this paper we study the interaction between choreographic movement in space and music, which is implemented by means of realtime tools called **Topos**, programmed in the platform Pure Data [3] (aka PD). These tools inherit from developments proposed in recent studies and provide a number of techniques that contribute to description of human movement in space. By enriching the computational representation of dance gestures we expect to improve the concepts and techniques that involve dance and music interactions.

In the next sections, we describe the objectives and limitations of this study (Section 2), previous work (Section 3) and theories (Section 4) that inspired the present approach. In the Section 5 we explain the design and implementation of the features, which are illustrated in Section 6 by means of examples and application scenarios. In the last section (Section 7) we discuss critical aspects and problems of the library, possible solutions and future studies.

2. OBJECTIVES

The aim of this study is to present and explore realtime features of choreographic gesture in space for computer music applications. Although the work focuses on dance-music interactions, it also makes use of generalized representations that might be useful to other analytical (non-realtime) and realtime approaches. The features are developed from low-level three-dimensional position data [1], such as position of a point in the Euclidean space (e.g.: x, y, and z position). Our problem is to develop strategies that can provide more comprehensive feature descriptions of human movement at the top of low-level motion descriptors. The implementation as Pure Data objects aims at providing an non-exhaustive collection of realtime tools (open-source)

[1] Motion capturing techniques and devices are not approached in this study.

that can be available for artists and researchers. In the next sections, we describe the background that supported this work.

3. REVIEW OF LITERATURE

3.1 Previous work

It is possible that great part of the efforts dedicated to build musical instruments were concentrated on mapping the sophisticated capabilities of human movement to the fine control of musical sounds. Although representations of music and dance seem have been always present in the development of music and dance forms (e.g.: XVIII century dance-music representations in [4]), the idea of a gestural movement serving as a control or descriptor for music performance or analysis is relatively new. For example, in 1920's Gustav Becking analyzed gesture curves in order to identify characteristics of the music styles [5] while in the work of Truslit [6] the music parameters are coordinated in the form of drawings. In first the half of XX century, the work of Rudolf Laban [7–9] on the theory and notation of dance expressed the necessity to combine a systematic view on expressive dance features by means of notational technology. In the 1970's, Clynes attempted to derive emotional states from movement curves [10]. In the last decades of the XX century, the rise of computational technology promoted several approaches in the notation and analysis of dance, such as the works presented in [11–13].

In the last years, the availability of computer power and a number of motion capturing methods created a diverse panorama of representations, notational approaches and interactive experiences for dance and music. For example, the Eyesweb [14] and Isadora [15] platforms provided a number of dance features derived from video analysis and sensors. Similar techniques based on video were further implemented in platforms such as Pure Data [3], VVVV [16], Max/Msp [17] and other graphic programming languages (for a survey of tools, see [18]). The availability of microelectromechanical systems (MEMS) was responsible for the dissemination of accelerometer and compass data in consumer devices, which facilitated studies on human movement. More recently, the access of motion capture data from low-cost optical motion capture systems and gaming devices (e.g.: Kinect) started a revolution in approaches based on position tracking, which also influenced software developments for research and analysis of body movement, such as the MocapToolbox [19].

So far, the analysis of the gesture involved in music and dance still remains a demanding and unexplored area. Challenges in the field include the inherent variability and parallelism of human movement [20] and its connections with the cognition, relationships with other domains, emotion, expressivity among others. The use of human movement data in art and in realtime interactive systems inherited many of these problems and posed many other questions. A brief look at the theories of the cognition of dance and music in the next section may help to tackle some of these problems.

4. THEORETICAL BACKGROUND

4.1 Space

The human action in space is one of the core elements of performatic display of dance [21] and was subjected to various approaches in the literature. For example, Hall [22], investigated how individuals organized the different classes of space (e.g. personal, intimate, public) and territories across different cultures. Previc and colleagues [23] reviewed behavioral and neurological literature concerning the neuropsychology of 3D space. For these authors, our interactions with the spatial environment are "*the single most important influence on the forging of the structural and functional architecture of the human brain*". Other evidences show that individuals encode patterns of movement in space by enacting both "external" and "internal" frames of gesture representation [24], which demonstrates the plurality [25] of gesture representation in human cognition. Paillard [26] proposes that the body schema would be composed of *morphokinetic* and *topokinetic* components. The *morphokinetic* component relates to the shape or the form of a gesture in the space. The *topokinetic* component pertains to the location of the gesture in space [26]. The topokinetic component of the gesture and its dynamics in time define the main scope of this study.

4.2 Spatiotemporal representations

Like music, dance develops in time, which is reportedly to be an inherent dimension of dance experience [21, 27] and probably the dimension that better interacts with the musical domain. The timing features involved in dance include familiar musical characteristics such as rhythm, synchronization, periodicity, metrical order and organization among others reported in the literature. Although dance encompass many other possibilities of analysis and representation, the *space* – considered as the medium for the deployment of movement – and the *time*, – seen as medium for segmentation and synchronization of movement [28] – seem to encompass a significant level of information needed to represent gestures. A spatiotemporal representation that manages both time and space would then provide an explorative field in which we look for higher representations of dance movement.

4.3 Data and imposed metrics

Low-level motion descriptors such as the kind of data delivered by motion capture devices are rooted in a metrical system that describes a three-dimensional Euclidean space. No matter how precise theses systems may be, this kind of unprocessed information if often difficult to use for artistic applications since it reflects an artificial axis. How space and time representations can be transformed in descriptors that reflect the qualities, intensities and dynamics that we observe in dance?

Carlsson, in [29], formalized a similar problem in four elements: (1) it is necessary to produce more qualitative information about the data, its (2) metrics are not theoretically justified for all systems (such as art), (3) coordinates are not natural do not carry intrinsic meaning and (4) summaries of information are more valuable than isolated parameters. We propose that a more comprehensive modeling of dance-music would involve representation of human variability, time and space. It must provide relationships independent from metrics and be able to attach qualities in the choreographic space. Topological methods may provide appropriate tools to deal with these problems [29].

4.4 Topology and gestures

Topology — or the study of *topos*, "place"— deals with qualitative and flexible geometric information such as proximity, connectivity and envelopment, ignoring information about shape, distances, sizes, and angles [29, 30]. This flexibility has provided a tool for mathematical abstraction, in which one can infer inherent connectivity of objects while ignoring their detailed form [31] and subjacente metrical system. Applications inspired in these concepts may make use of quantitative information (such as points measured in space, distances, angles) or abstract quantities to derive topological relationships. Examples of these applications can be found in fields such as qualitative reasoning [32], geographical information systems (GIS) [33], and spatial cognition [34, 35]. The **Topos** library derives specifically from the topological gesture analysis (TGA) proposed in [28] and further developments such as the Windowed-Topological Gesture analysis (wTGA), presented in [36] and briefly describe in the following sections.

4.4.1 Topological Gesture Analysis (TGA)

The Topological Gesture analysis (TGA) is a method introduced in [28] that consists of projections of time-based qualitative information – or cues – onto the gesture trajectories. The projected cues form regions populated by clusters of cues – points in a space called *point clouds* – whose volume confines the trajectories of the gesture within the time section. It can be said that the whole dimension of time populated by cues and qualities is transformed into points in space. The points become regions that inherit the qualities of the cues. Figure 1 illustrates how musical cues are (a,b) projected and (d) discriminated in the dance space from the perspective of the TGA approach.

4.4.2 Windowed Topological Gesture Analysis (wTGA)

The windowed topological Analysis (wTGA, proposed in [36]) is a variation of the TGA analysis. Instead of using external information (cues) to project "qualities" onto the space, the trajectories itself shape the structure of point clouds in a time window, which provides geometrical and temporal characteristics. The analysis uses a running window of n time points sampled from the recorded trajectories that produces a point cloud that morphs across time. In our first proposition in [36] a time window of 1 second (100 samples, at 100 fps) was used to compare the evolution of the use of space in 3 excerpts of dance improvisation. The Figure 2 illustrates rationale of this process.

The methods implemented in the **Topos** library inherit from the wTGA the idea of a (time) windowed analysis

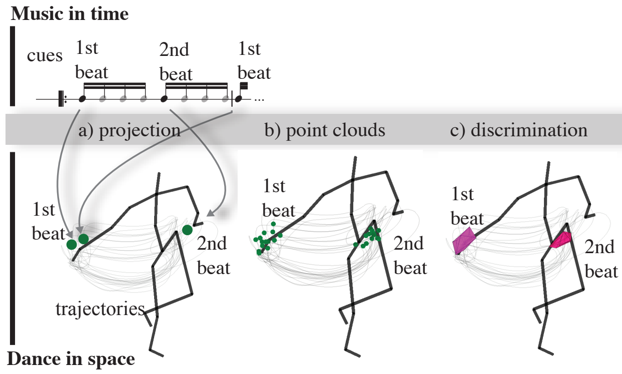

Figure 1. Topological Gesture Analysis applied to a sequence of dance gestures. The cues used in this example represent the musical metrical qualities (first and second beat levels) annotated in the time dimension. The accumulation of cues in time generate point clouds in (b) that are further discriminated by its qualities.

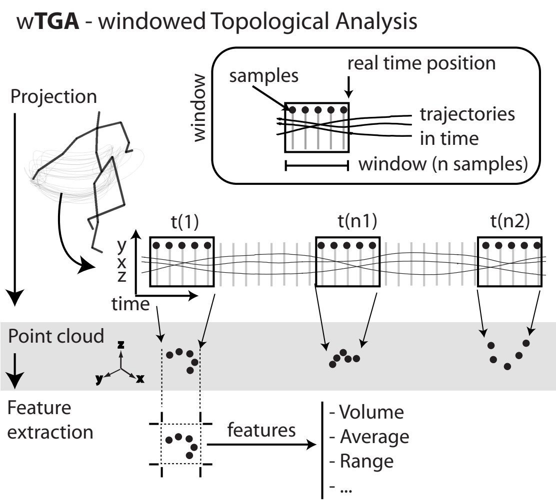

Figure 2. Schematic representation of the wTGA method: projection of sample window, point clouds and feature extraction

that provides descriptors of the space. By transforming the low-level data (e.g.: x, y and z coordinates) into point clouds it generates new levels of information that are not entirely dependent from metrical scales and coordinates. In this study, we implement the basis of the wTGA in a realtime system, as explained in the following sections.

5. METHODOLOGY

5.1 Implementation

The Topos library is implemented a set of programs in Pure Data, entirely written in Pure Data graphical language. The elements of the library are composed of Pure Data's high-level "functions" called *abstractions* or simply objects. Abstractions are built in the form of graphical objects that interact with each other by means of graphic or symbolic links (lines, send and receive objects/messages) [37]. Consequently, all code operations and algorithms are availa-

ble for the programmer/user directly from the abstractions. The abstractions contain help files that indicate the usage and examples. The library uses code and 3^{rd} part objects from other pd-extended libraries such as the GEM, Mtl, iem_tab and zexy. The "Topos" library is available for free download at the website
`http://naveda.info`.

5.2 Scope

The basic workflow comprises a given three-dimensional stream of floats (e.g., x, y and z) that is transformed by abstractions and produces other feature streams, datasets or messages. Within our application framing, the three-dimensional stream is expected to be the output of a motion capture device and gesture movements in the euclidean space. However, experiences using other kinds of data streams (velocity, sensor data, etc.) and hybrid high-dimensional data spaces (e.g.: spatial displacement and musical pitch, velocity and sound intensity) may provide interesting possibilities not examined in this paper.

5.3 System design

The design of the **Topos** library involves objects that transform, extract features and describe interactions between data structures, movements and external information. The Figure 3 shows a diagram of the system. Each element of the diagram represents one or more Pure Data abstractions. The types of objects present in the library are briefly described below:

- **Topological operations** involve transformation or qualification of 3D points or point clouds.
- **Topological features** involve the extraction of features from point clouds and three-dimensional data.
- **Geometrical features** involve the extraction of features from the geometry of the 3D points, which are expected to be a representation of human body movement.
- **Relational features** produce information about the interaction between points and regions in space.
- **Tools** are complementary objects that facilitate programming and data flow routines.

These classes and transformations will be specified in the following sections.

5.4 Topological operations and features

In the next subsections we describe the algorithms and proposals implemented in the library. Although the relevant contribution of the study lies on the descriptions of the algorithms we will provide minimal code examples as a practical description of the data flow in the system.

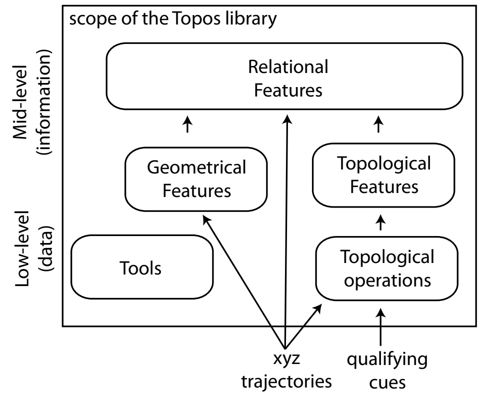

Figure 3. Schema of the design and data flow of the system.

5.4.1 *The object [topos]*

Algorithm: The main algorithm in library is implemented in the abstraction [`topos`][2]. The abstraction performs the transformation of three-dimensional streams of floats to a morphing point cloud that can be also visualized as a polygon in 3D space (GEM/OpenGL). The rationale behind the implementation of the object starts with the accumulation of a n number of points from a realtime stream of floats (x, y and z) in a cache. The actual (realtime) state of the cache is projected in the data space or a polygon in a visual 3D space. This process creates a running spatio-temporal shape in space that mirrors the internal cache that stores the samples in the time window. The internal and circular clock of the object addresses the actual (realtime) sample point to the next vertex of the polygon (or an index in the cache). The collection of points in this window forms the *point cloud* topology. In short, the point cloud represents occupation of the space used by the dancer in the last time periods (defined by the window). Figure 4 illustrates the algorithm and the minimal code example necessary to run it.

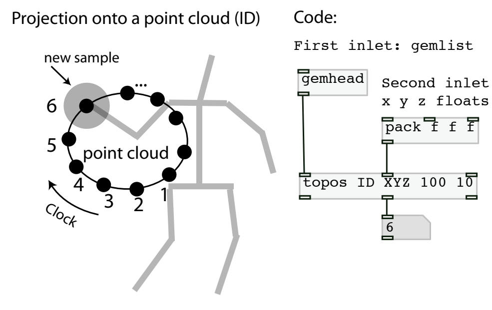

Figure 4. Example of usage of the object [topos]. The object takes as input a list of floats (x y z) that can be sent either through the second inlet or through sends (2^{nd} argument). Sampling in ms (3^{rd} argument) and number of points in the point cloud (4^{th} argument) are set in creation of the object or through messages to the 3^{rd} inlet.

The object [`topos`] process the visualization and stores

[2] References to Pure Data objects and their code operations will be enclosed between squared brackets (ex.: [`topos`]).

the point cloud in a cache. It also broadcasts the actual point cloud references by means of symbolic connections (see the method in Figure 6). In the first proposition of the wTGA method (implemented in Matlab platform) the visualization of the point cloud was produced using Delaunay tree methods [38]. In the present realtime implementation, we opted to visualize it through a GEM polygon (openGL) [39] composed of $n + 1$ vertices that visualize the points provided by the time window cache, composed of a set of 3 arrays (x, y and z).

Operation: There are 4 arguments in the [`topos`] object, namely: [1] symbolic ID, [2] a receive list (receive), [3] sampling period (in ms) and [4] number of samples per window. For example, an object created with the following arguments [`topos lefthand lh/pos 100 30`] broadcasts its output to other objects through the ID (identification) "lefthand" and receives a list of 3 floats (x y z) from the symbolic link "lh/pos". These operations are sampled at each 100 ms (3rd argument) and store 30 data samples (4rd argument). As expected, 100 ms * 30 samples will produce a point cloud that represents the space used in the last 3 seconds in samples (window or memory of 30 samples or 3 seconds).

The visualization of the point cloud is implemented as a normal GEM object, which inherits the properties of the GEM object [`polygon`]. More details about these operations are available in the help patch. Three types of visualization are exemplified in the Figure 5. Other functionalities are also implemented as messages to the 3rd inlet of the [`topos`] object, for example:

- `RandomLevel`: adds random noise to the set of points in the point cloud.
- `setPoint`: forces a point to be in a given position specified in the message.
- `freeze`: freezes the point cloud in the space

Other functionalities include controls of the visualization and internal clocks of the object, explained in the help patch.

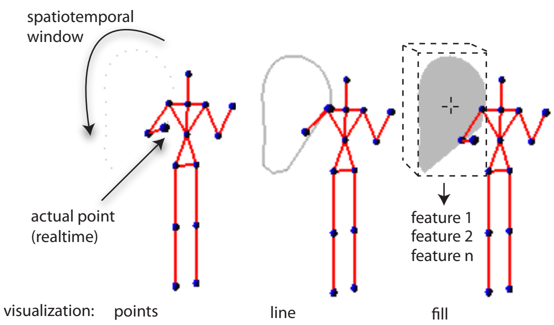

Figure 5. Visualization of the point cloud in three different graphic GEM methods available in the library.

5.4.2 *[topos.features]*

The features of the moving point cloud stored in the object [`topos`] are retrieved through the object [`topos.features`]. This object extracts and distributes the characteristics of the point cloud (see Section 4.4.2, for the wTGA explanation). The only argument of the object is the symbolic ID of the starting [`topos`] object, which operates the original projection of the point cloud. Figure 6 demonstrates creation arguments and the symbolic link that enables data flow between the two objects. Table 1 describes the name, output format and description of the features.

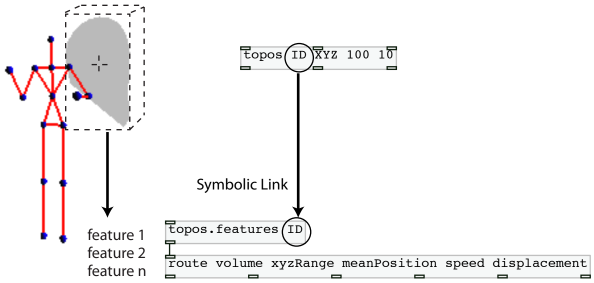

Figure 6. Basic setup of the [topos.features] object and the symbolic link set as "ID" in both objects.

Feature	Format	Description
Volume	f (float)	Volume of the bounding cube containing the point cloud
3D range	f f f	size of the point cloud (max-min) as a projection of the maximal and minimum onto each x y and z dimension
Average position	f f f	average 3D position of the point cloud
Displacement	f	displacement of the average position of the point cloud
Speed	f	speed of the average position of the point cloud

Table 1. Description of topological features available in the object [`topos.features`]. See help file for the method of routing these features

A number of higher-level qualities of the point clouds can be deduced from set of features available in the object [`topos.features`]. For example, the volume indicates the evolution of the magnitude of the space being occupied. If one ranges of x y z components is higher than the other it indicates that the geometry of the gesture is larger in one axis. This feature is easily accessible from the feature `3D range`. Many other qualitative information can be extracted using the relational and qualitative features presented in other objects of the library.

5.4.3 [topos.quality]

The object [`topos.quality`] (no arguments) projects and stores classes of cues onto each element of the point cloud [3]. Classes can be a numeric or symbolic representation of cues. They can refer to any data structure represented in a Pure Data list (e.g.: symbols, floats). The object uses the internal clock of a [`topos`] object – that indicates which vertex of the polygon is receiving new data points in the realtime – and stores incoming classes in the index provided at the moment of projection. The message *get* x sent to the first inlet retrieves the list of the recorded classes associated to an index x. In the example in Figure 7, a [`notein`] object provides midi messages that are used to project pitch and velocity qualities onto the gestural space. See Section 6.1 for a complete example.

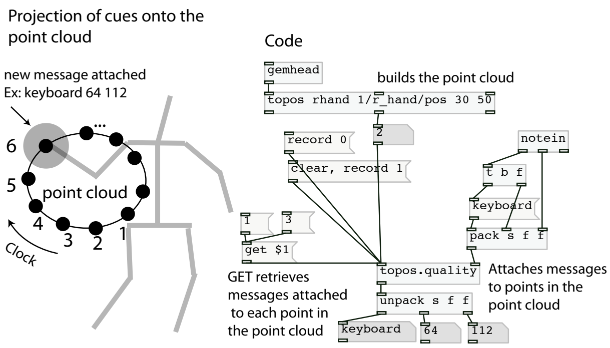

Figure 7. Example of the usage of the object [`topos.quality`]. In this example the object attaches a list (a symbol and two floats: 'keyboard', pitch and velocity) to the points in the cloud. The message "get [float]" retrieves the information attached to each point.

5.5 Geometric features

Geometric feature objects process geometric information constructed from three-dimensional points. Although the logic behind these objects is based on simple geometric operations, it is expected that the input signals come from body movement data. The Figure 8 illustrates some features implemented in these objects.

5.5.1 [topos.geo.2]

The [`topos.geo.2`] extracts features generated from two 3D points. The features include sum of speed, 3D angle and distance between the two points. Additionally, it also includes a specific measure of *instability*, which takes into account the angle and distance between the two points, specified in the following equation:

$$instability = |\frac{|\measuredangle ab| - 90}{180} - 0.5| * d^{ab} \quad (1)$$

where $\measuredangle ab$ is the 2d angle between the two points in grads (a and b) and d^{ab} is the Euclidean distance between the two points.

[3] From the algorithmic viewpoint, the object only receives the indices of the internal clock of a [topos] object and attach to them any kind of Pure Data list. The list can be further retrieved using messages to the [`topos.quality`] object itself

This simple relationship between angle and length may be meaningful for applications dealing with higher level features. For example, if these two points are taken from trunk and head, the instability level will increase as the angle decrease in both sides (see Figure 8), which means that the trunk is changing from a less unstable position to a more unstable pose.

The [`topos.geo.2`] is initiated with 3 arguments: symbolic ID, point A (receive or 2^{nd} inlet) and point B (receive or 3rd inlet). Example: [`topos.geo.2 hands lhand/pos rhand/pos`].

5.5.2 [topos.geo.4]

The [`topos.geo.4`] is designed to process rectangular geometries, which are normally attached to 4 extremities of the body (e.g.: hands and feet). It is expected that the first two points relate to upper body parts (e.g.: hands) and the last two relate to lower parts (e.g.: feet). The features include (1) a sum of the instability measurements, (2) an average position between points and (3) the volume of the bounding rectangle that encloses the 4 points. The last feature (demonstrated in Figure 8) is specifically connected to the expansion/retraction quality often mentioned in dance theory and other applications (e.g.: [40]).

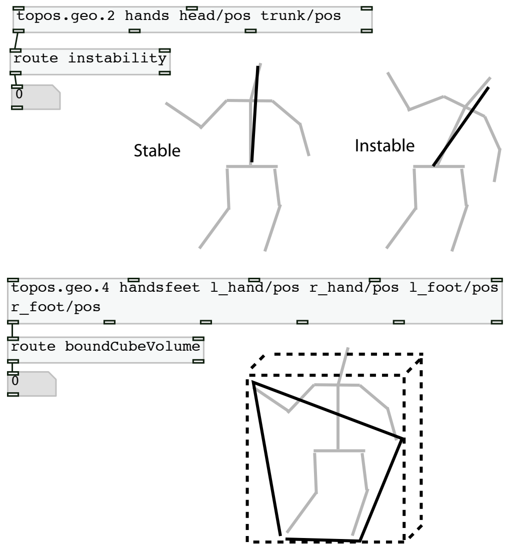

Figure 8. Usage and demonstration of two features present in the geometric objects.

5.6 Relational features

Relational objects operate and broadcast higher-level feature descriptions such as collision, distances, connections and envelopment (see [32]). These objects are specially interesting for applications involving interaction since they signalize important topological relationships between points and point clouds. Most of the objects contain thresholds that transform metrical information in qualitative information (e.g.: if a distance between two positions is below x mm, then broadcast the topological relationship "contact").

5.6.1 *[topos.rel.pointRegion]*

The object [topos.rel.pointRegion] detects relationships between a 3D point and a point cloud region. Relationships include collision, envelopment and others. For example, if a 3D point approximates to a point cloud (defined as the half of the maximum 3D range component) the feature routed as “hitRegion” outputs [1]. If a 3D point reaches the threshold of collision to any point in the point cloud it outputs the element (routed as element) and velocity (routed as velocity) before the collision. Other features include the minimum actual distance from the point cloud (minDistance) and the position at the collision (posCollision). The Figure 9 demonstrates the implementation. Like the other objects the arguments of the [topos.rel.pointRegion] are created with a [1] symbolic ID, [2] a 3D Point, [3] a reference to a [topos] ID and [4] a threshold.

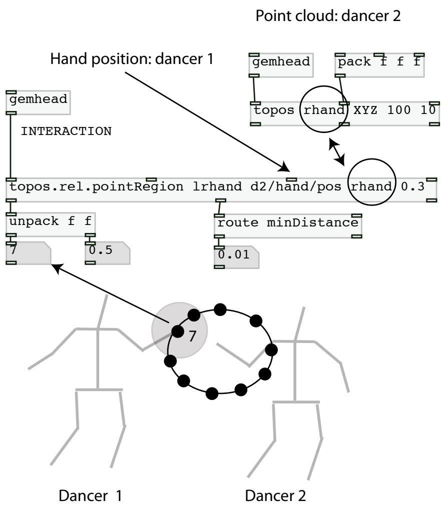

Figure 9. Diagram of the patch that illustrates the relational features implemented in the object [topos.rel.pointRegion]. In this patch the point cloud designed by the dancer 2 interacts with the position of the hand of the dancer 1. The object provides the references to the position and velocity at the moment of collision.

5.6.2 *[topos.rel.points] and [topos.rel.regions]*

The relational objects defined as [topos.rel.points] and [topos.rel.regions] extract features from the relationships between points and between regions. Like the other objects the arguments of the include a [1] symbolic ID, two elements (either a 3D Point receive symbol or point cloud region created and referenced by a [topos] ID) and a threshold. Likewise, if distance between elements reach a given threshold the objects outputs the index of the point in the point cloud (routed as “element”) and velocity (routed as “velocity”) of collision. Other features include the distance between points and average position.

5.7 Other tools

The library includes other tools such as the object [topos.skeleton] that organizes OSC messages between motion capturing applications and the Topos library. Until now only the applications Osceleton [4] and Synapse [5] are implemented. Other tools like [topos.rec] and [topos.rec.slave] assist the recording, exporting and inspection body motion capture data.

6. APPLICATIONS

In this section we describe three scenarios of applications. The library and some of the applications mentioned here were used in the development of three dance performances with movement, sound, image and network interactions [41–43]. The Figure 13 shows the dancer and the representations of point clouds projected against a translucent screen in one of these performances [41]. Other videos and images of other performances are available at the website http://naveda.info.

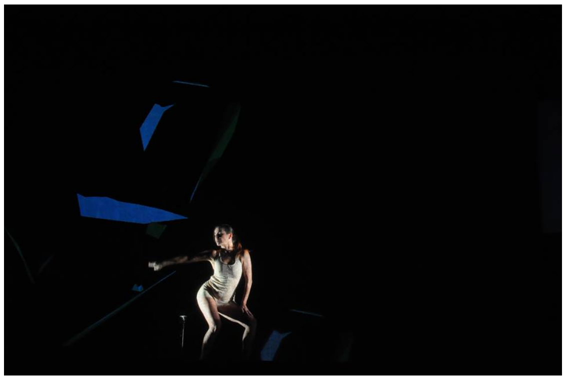

Figure 10. Dancer and visualization of point clouds of the local and remote dancer (networked performance). Performance: “Embodied Varios Darmstadt 58” (2013). Dancer: Ivani Santana. Photography: Shai Andrade

6.1 Realtime mappings of music in the dance space

The availability of computer power provides great possibilities of parallel processing. However, mapping strategies are often very direct, linear and static due to complexities of routing and control. In the Figure 11, a midi instrument is played while the dancer builds a point cloud from his/her gesture movements. The patch attaches the musical pitch and velocities coming from a musician to the indices of points in space at the moment the point cloud is created (A). Using methods such freeze 1, it is possible to freeze the transformation of the point cloud, which allows the dancer to play the pitches and velocities by (B) interacting (colliding) with the regions in space where the musical information was recorded. Explorations may include variations of a the original pitch sequence (*i.e.* theme variation) and interactive *duetos* with the musicians. The point cloud can be reconstructed in other positions. Other midi

[4] Available online at https://github.com/Sensebloom/OSCeleton
[5] Available online at http://synapsekinect.tumblr.com

parameters, filters or transformations can be applied using other movement features.

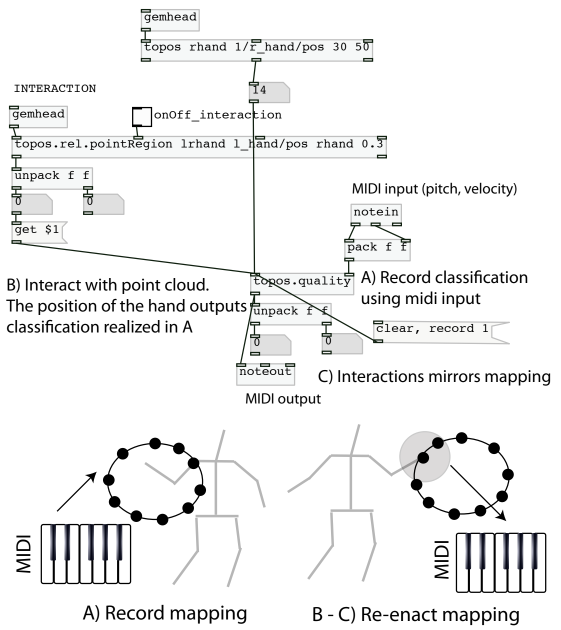

Figure 11. Diagram of the patch and resulting data visualization. In this example, in (A) a musical midi keyboard is played while the point cloud is generated. The dancer can mirror the musical information in (B) by touching the regions that will output information recorded from the keyboard.

6.2 Control of parameters for synthesis using high level features

One of the main problems in computational synthesis is the control of the synthesis parameters. The possibilities of exploration, control and storage of parameters using the massive information from body movements or key-poses can be used as a exploratory tool for sound synthesis. In the example displayed in Figure 12 the distance between the left hand to each point of the right hand point cloud is used to control the magnitude of the harmonics in a addictive synthesis. The dynamic flow of parameters possible in the change of positions of the hands and the points generated in the point cloud can improve the exploration of timbres and engage the dancer in choreographic explorations across the timbre space. These are easily implemented using the feature `distanceFrom` in the [`topos.rel.pointRegion`], which outputs the distances of every point in the point cloud to a given 3D point.

6.3 Integration of controls and dancers in networked/telematic art

In networked or telematic art, the integration of the performances and creative forces among displaced nodes is a challenge. There is a risk of developing parallel and disconnected creative work due to the distance between dancers and creation space. In the example displayed in the

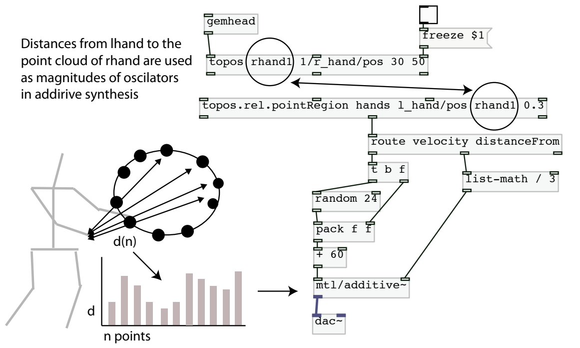

Figure 12. Diagram of the patch and resulting data visualization. In this example the distance from a hand to n points is used to control the magnitude of a simple additive synthesis.

Figure 13 the point cloud of the local dancer is disrupted by the position of the hand of a remote dancer, whose data arrives from the network. While the local dancer is still influencing the shape of the point cloud, the result of the interaction is a single geometrical element and dataset composed of the gestures of both dancers. The use of features resulted from these interferences in sound synthesis or even in the graphical output can help to find conceptual solutions for the integration in these scenarios.

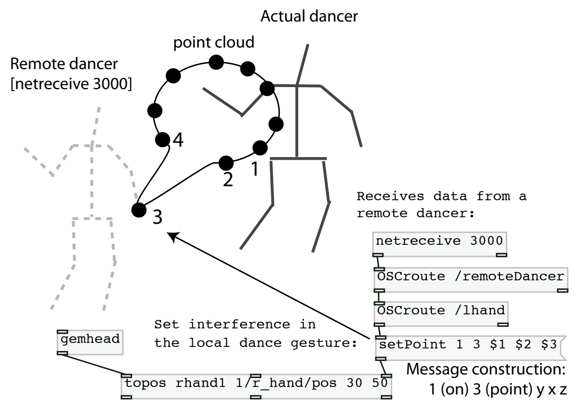

Figure 13. Diagram of the patch and resulting data visualization. In this example the hand of a remote dancer interferes in the point cloud of the local dancer. It not only changes visualization but the whole data structure and features.

7. DISCUSSION AND FINAL REMARKS

In this paper we described the implementation of novel realtime features that provide gestures descriptors for interactive applications in music. The exploration was based on features extracted from the quality of dance gestures in space and applied to realtime implementation of strategies based on previous analytical work in the field. Although a part of this paper deals with the technical demands of programming and developing strategies for music and dance interactions, the conceptual questions and new ideas proposed here may open perspectives that go beyond the implementations.

So far, the work presented here does not attempt to be a generally applicable strategy. The description of a topological perspective of the dance movement involves specific characteristics that may not fit to all conceptual and performatic contexts. The occupation of space is contextual and needs time to be shaped. Since we cannot predict the direction of creative movement patterns, the evolution of its topological features in time is gradual and highly affected by the time window. This means that some features will not be so responsive to some gestures. They need time develop across the contextual frame in the same way human perception needs time to gather contextual information in humans. Some proposed features may not be as reactive as other low-level features, such as acceleration profiles, for example.

In applications where the principle of strict synchronization in time is not relevant, the topological features extracted from movement may provide interesting solutions for the control demands of multimedia applications. Interaction ideas such as the one proposed in Section 6.1 or 6.2 may help to develop more reactive interactions with the choreographic space. Above all, the qualitative level of feature description in this work may provide better or at least different accounts of the gesture for multimedia performances. We understand that this contributes to the development of more comprehensive applications in fields such as gesture-sound interaction, dance, digital luthery, networked performance, among others.

7.1 Future work

There are many limitations in the implementation point cloud operations and visualizations. Many other possibilities can be realized by means of development of the core objects in the library, possibly in C++ or Lua objects. Further developments using Kalman Filters or simple statistical techniques may help to develop a movement predictor that helps to generate probabilistic point clouds before they appear. This could have a great impact on features related to expectancy. Other future ideas include better visualizations and implementations of computer techniques developed in the original work on Topological Gesture Analysis [28]. Many other features can be derived from the actual processes and will be added to the tools in the next releases of the library.

Acknowledgments

We are thankful for the suggestions of the anonymous reviewers and for the dancers and professionals who contributed to the developments of the ideas presented in this paper. We are also thankful for the support of a grant from FAPEMIG (Fundação de Amparo à Pesquisa do estado de Minas Gerais).

8. REFERENCES

[1] A. Grau, "Sing a Dance. Dance a Song: The Relationship between Two Types of Formalised Movements and Music among the Tiwi of Melville and Bathurst Islands, North Australia," *Dance Research: The Journal of the Society for Dance Research*, vol. 1, no. 2, pp. 32–44, 1983. [Online]. Available: http://www.jstor.org/stable/1290759

[2] J. Hanna, "Toward a cross-cultural conceptualization of dance and some correlate considerations," *The Performing arts: music and dance*, p. 17, 1979.

[3] M. Puckette, "Pure data: another integrated computer music environment," pp. 37–41, 1996.

[4] K. Tomlinson, *The Art of Dancing Explained by Reading and Figures, repr*, London, 1735.

[5] G. Becking, *Der musikalische Rhythmus als Erkenntnisquelle*. Benno Filser, 1928.

[6] A. Truslit, *Gestaltung und Bewegung in der Musik*. Berlin: Chr. Friedrich Vieweg, 1938.

[7] R. Laban, *Schrifttanz [Writing dance]*. Vienna, Austria: Universal Edition, 1928.

[8] R. Laban and F. C. Lawrence, *Effort*. London: Macdonald and Evans, 1947.

[9] R. Laban and L. Ullmann, *Choreutics*. London: MacDonald and Evans, 1966.

[10] M. Clynes, "Sentic cycles: The seven passions at your fingertips," *Psychology Today*, vol. 5, pp. 58–60, 1972.

[11] A. Camurri, P. Morasso, V. Tagliasco, and R. Zaccaria, "Dance and Movement Notation," *Human Movement Understanding: From Computational Geometry to Artificial Intelligence*, 1986.

[12] D. Herbison-Evans, "Dance, Video, Notation, and Computers," *Leonardo*, vol. 21, no. 1, pp. 45–50, 1988.

[13] T. Ungvary, S. Waters, and P. Rajka, "NUNTIUS: A Computer System for the Interactive Composition and Analysis of Music and Dance," *Leonardo*, vol. 25, no. 1, pp. 59–68, 1992.

[14] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems," *Computer Music Journal*, vol. 24, no. 1, pp. 57–69, 2000.

[15] S. DeLahunta, "Isadora almost out of beta: tracing the development of a new software tool for performing artists," *International Journal of Performance Arts & Digital Media*, vol. 1, no. 1, pp. 31–46, 2005.

[16] "vvvv - a multipurpose toolkit," 2014. [Online]. Available: http://vvvv.org/

[17] "Max." [Online]. Available: http://cycling74.com/products/max/

[18] M. Zadel and G. Scavone, "Laptop performance: Techniques, tools, and a new interface design," in *Proceedings of the International Computer Music Conference*, 2006, pp. 643–648.

[19] P. Toiviainen and B. Burger, *MoCap Toolbox Manual*. Jyvskyl, Finland: University of Jyvskyl, 2011. [Online]. Available: http://www.jyu.fi/music/coe/materials/mocaptoolbox/MCTmanual

[20] N. Stergiou and L. M. Decker, "Human movement variability, nonlinear dynamics, and pathology: is there a connection?" *Human movement science*, vol. 30, no. 5, pp. 869–888, 2011.

[21] J. L. Hanna, *To Dance is Human: A Theory of Nonverbal Communication*. University Of Chicago Press, 1987.

[22] E. T. Hall, "Proxemics," *Current Anthropology*, vol. 9, no. 2/3, p. 83, 1968.

[23] F. H. Previc, "The neuropsychology of 3-D space," *Psychological Bulletin*, vol. 124, pp. 123–164, 1998.

[24] M. Ghafouri and F. G. Lestienne, "Contribution of reference frames for movement planning in peripersonal space representation," *Experimental Brain Research*, vol. 169, no. 1, pp. 24–36, 2006.

[25] J. Paillard, "Motor and representational framing of space," *Brain and space*, pp. 163–182, 1991.

[26] E. Mullis, "The image of the performing body," *The Journal of Aesthetic Education*, vol. 42, no. 4, pp. 62–77, 2008.

[27] S. Brown, M. J. Martinez, and L. M. Parsons, "The Neural Basis of Human Dance," *Cerebral Cortex*, vol. 16, no. 8, pp. 1157–1167, 2006.

[28] L. Naveda and M. Leman, "The spatiotemporal representation of dance and music gestures using Topological Gesture Analysis (TGA)," *Music Perception*, vol. 28, no. 1, pp. 93–111, 2010.

[29] G. Carlsson, "Topology and data," *Journal: Bull. Amer. Math. Soc*, vol. 46, pp. 255–308, 2009.

[30] L. C. Kinsey and T. E. Moore, *Symmetry, shape, and space: an introduction to mathematics through geometry*. Emeryville: Key College Pub, 2001.

[31] E. Weisstein, "Topology," 2010. [Online]. Available: http://mathworld.wolfram.com/Topology.html

[32] A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts, "Qualitative spatial representation and reasoning with the region connection calculus," *GeoInformatica*, vol. 1, no. 3, pp. 275–316, 1997.

[33] P. Bogaert, N. Van de Weghe, and P. De Maeyer, "Description, definition and proof of a qualitative state change of moving objects along a road network," M. Raubal, A. Sliwinski, and W. Kuhn, Eds., Münster, Germany, 2004, pp. 239–248.

[34] C. Freksa, *Qualitative spatial reasoning*. Springer, 1991.

[35] M. Knauff, R. Rauh, and C. Schlieder, "Preferred mental models in qualitative spatial reasoning: A cognitive assessment of Allen's calculus." Mahwah: Lawrence Erlbaum Associates, 1995, p. 200.

[36] L. Naveda and I. Santana, "Space, music and body dynamics in three excerpts of dance improvisation," in *DRHA2010*. London, UK: Brunel University, 2010.

[37] A. Farnell, *Designing sound*. MIT Press Cambridge, 2010.

[38] J.-D. Boissonnat and M. Teillaud, "The hierarchical representation of objects: the delaunay tree," in *Proceedings of the second annual symposium on Computational geometry*. ACM, 1986, pp. 260–268.

[39] M. Danks, "Real-time image and video processing in gem," in *Proceedings of the International Computer Music Conference*, 1997, pp. 220–223.

[40] A. Camurri, B. Mazzarino, and G. Volpe, *Analysis of Expressive Gesture: The EyesWeb Expressive Gesture Processing Library*. Heidelberg: Springer Verlag, 2004, vol. 2915, pp. 460–467.

[41] I. Santana, L. Naveda, R. Sanchz, and A. Baumann, *Embodied Varios Darmstadt 58*, ser. Artistic project. Salvador, Barcelona, Ciudade de Mexico: Iberescena, 2013, dance performance - Networked performance.

[42] I. Santana, L. Naveda, and F. Silva, *Embodied Varios Darmstadt 58*, ser. Artistic project. Belo Horizonte, Brazil: SESC, 2014, dance performance.

[43] F. Santos, D. Chamone, Paulo adn Herrmann, J. Villas, and L. Naveda, *Miradas do Caos II (Chaos Looks II)*, ser. Artistic project. Belo Horizonte, Brazil: PBH/Oi Fuguro, 2014, dance performance. [Online]. Available: http://miradasdocaos.com/