Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

ChoreoGraphics:
An Authoring Environment for Dance Shows

Adriana Schulz, Luiz Velho and Wojciech Matusik

Technical Report TR-2012-07 Relatório Técnico

November - 2012 - Novembro

ChoreoGraphics: An Authoring Environment for Dance Shows

Adriana Schulz* MIT
Luiz Velho† IMPA
Wojciech Matusik‡ MIT

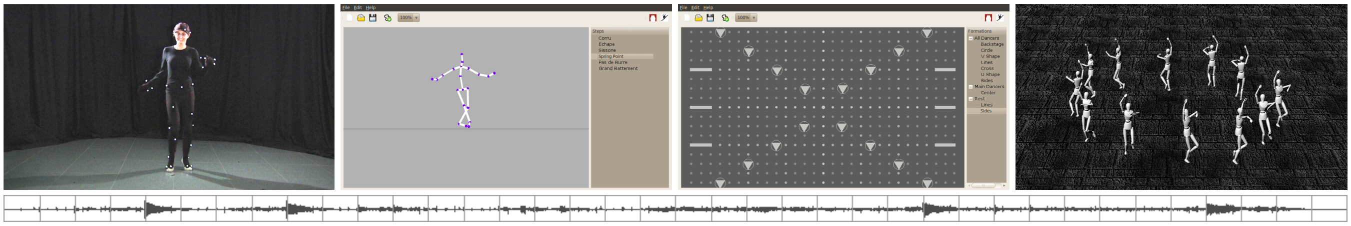


Figure 1: *The creative tool for dance design stimulates interactions between dancers, choreographers and musicians. The musicians are responsible for composing the soundtrack (bottom) and we use a motion capture system to capture the performance of the dancers, creating a list of steps (left). Choreographers use a designed interface to determine the sequence of dance steps and the way the dancers move on stage as a group (middle). The system synthesizes the specified motion to allow visualizations of the designed performance (right).*

Abstract

This work describes a computer system for authoring dance shows. The system can be used both as a mechanism to synthesize virtual group performances and as a tool for interactive design and visualization of real choreographies. To this end, we use as an input motion capture of dancers synchronized with an underlying musical composition. We adapt motion synthesis methods for use in combination with a musical track. The authoring system provides methods for controlling group movement of dancers on stage, such as creating formations and following trajectories. Our integrated platform suggests a new form of collaboration between artists, allowing the show to naturally evolve from iterative contributions from dancers, musicians and choreographers. In this way, we demonstrate how computer graphics research can be used to create new tools for artists to express their creativity.

Keywords: choreography, authoring, motion graphs

1 Introduction

Choreographic design has thus far had little exposure to the digital technologies. Although some choreographers have started to take advantage of computational techniques to allow additional visual output and special effects (e.g., the use of projectors during performances), it is fair to say that the tools used during the creative process of dance shows are still predominantly based on pen and paper. One of the major challenges choreographers face is that dance, unlike music, does not have a simple standard notation. Typically, dancers have to memorize the whole step sequences or use video to record their compositions. The representation of dance also presents a challenge when it comes to positioning the dancers on stage and determining the group dynamics. In addition to notation, these *group motions* also involve interactions and collision avoidance, which are difficult to plan without visualization tools. The solution is usually to try out many different trajectories during rehearsals. This can be stressful and tiresome depending on the group of dancers (consider, for example, children's dance recitals).

In this context, we propose a creative tool which allows planning, editing, and visualizing dance shows. This tool guides the artists through conception, production and execution. The system is fully integrated to promote the interaction of dancers, musicians and choreographers. We use a motion capture system to acquire dancers' performances and create a database of dance steps and corresponding musical tracks. To allow choreographic input, we propose an interface for specifying the combination of dance steps and group movement of the dancers on stage. The music is used both to guide the segmentation of the motion data into rhythmic structures and to set the timeline for the sequencing of events.

The output of our system is a full description of the designed performance in a computer representation that includes music, a sequence of steps performed by each dancer, a dynamic distribution of the dancers on stage, and an articulative motion of each dancer. This representation is used within an authoring tool for designing both real and virtual dance shows. Ultimately, the stored representation allows the show to be reproduced by other groups of artists.

In summary, the contributions of this paper are:

- an authoring system that guides artists through conception, production and execution of a dance show and integrates creative elements that compose a choreography: music, dance and group motion;
- a novel mechanism for synthesizing dance motion from motion capture data that explores the relationship between dance and music;
- methods for specifying group motions based on declarative specifications and procedural techniques; and
- a representation for specifying and storing choreographic information.

The paper is structured as follows. First, we discuss previous work (Section 2) and present an overview of the authoring platform (Section 3). Then, we describe the proposed interface for choreographers that allows specifications of step sequences and group motions (Section 4). We also introduce a language for representing group motions and propose declarative and procedural tools for design. Next, we describe the methods for motion synthesis (Section 5), that include techniques combining dance steps and motion editing tools. We validate the approach with examples and user evaluations, and discuss limitations (Section 6) before we conclude.

---

*e-mail: adschulz@mit.edu
†e-mail:lvelho@impa.br
‡e-mail:wojciech@mit.edu

2 Related Work

Traditional Dance Authoring. The creative process for designing dance shows involves several abstract decisions such as intention, theme, style, artistic impact and aesthetics [Blom and Chaplin 1982]. As with many other creative processes, it is highly subjective and, therefore, there is no universal sequence of decisions that choreographers follow to design a dance performance. However, when it comes to the actual specification of form (i.e., the plan for patterning movement or the motion sequence [Ellfeldt 1988] it is easier to find a common composition structure.

In general, choreographers start by selecting a piece of music and extracting its rhythmic information, e.g., they find the number of measures in the piece and segment it into different parts. Then, they design step sequences that match these rhythmic patterns. They often use storyboards to annotate their specifications. Storyboards are, in general, top-down views of the stage. Dancers are positioned using different shapes or color codes to distinguish different groups. Lines or arrows are used to specify motions. It is important to emphasize that these annotations are not at all standard and they usually can be only understood by the choreographer who designed them and his close collaborators. For specifying each step, choreographers sometimes can use an existent nomenclature (e.g., for classic ballet compositions, where each step has a given name). However, for most dance styles, this nomenclature is not available and therefore choreographers usually have to know steps from memory or use video recordings.

Synthesis and Editing of Motion Capture Data. Some of the earliest approaches to editing motion capture data involve signal processing techniques [Witkin and Popovic 1995; Bruderlin and Williams 1995]. The drawback of these methods is that they only analyze low level information and are unable to deal with more structural motion aspects. In this work, however, we were able to suggest simple and yet effective editing tools based on signal processing, by exploring the rhythmic structure of dance motion data.

In addition to editing, there are many approaches that synthesize new and more complex streams of motion from previously acquired data. Motion synthesis strategies include constructing models of human motion [Brand and Hertzmann 2000], interpolating motion to create new sequences [Kovar and Gleicher 2004] and reordering motion clips employing a motion graph [Kovar et al. 2002a]. Motion graphs create a directed graph from a set of captured motion clips by selecting similar frames (nodes) and creating connections by interpolating sequences of frames. A walk along this graph generates a new motion by re-assembling the captured data. In this work, we use a motion graph to sequence the set of captured steps. We augment this method to take into account the rhythmic structure of the dance that needs to be preserved during motion synthesis.

Declarative Group Motion. In dance literature there is no universal notation for describing group motions or a classification of the essential elements that should be used in order to specify them. The most important reference in dance analysis is the work by Laban [Laban and Ullmann 1960], which introduces a language for describing and annotating all forms of movement. We do not use Laban's representation in our system because it concentrates on analyzing movements of a single dancer, while our main interest is specifying group motions. In addition, Laban's language was developed to allow motions to be recorded and repeated in the future and therefore the movements descriptors (e.g., effort, shape) do not correspond to intuitive design elements.

Procedural Group Motion.

By transferring the creative process to the computer, we make multiple computational techniques available for use in our system. To exemplify how these can be used to suggests to new forms of artistic expression, we explored previous research in behavior animation, use procedural models to control motions of autonomous characters [Reynolds 1987; Tu and Terzopoulos 1994; Blumberg and Galyean 1995].

We adapted to the context of dance the work on *steering behaviors* [Reynolds 1999], that have been used to synthesize motions for animal herds and flocks with little user input. In this procedure, each character is represented as a vehicle that moves on a 2D manifold (in our case, the stage) translating and rotating. Control is given by a combination of steering forces that allows certain goals to be reached. Marshall and Leonard [Marshall and Leonard 2010] explore these techniques for dance. They experiment with using steering commands for real dancers.

Synchronizing Motion and Music The rhythmic structure of dance movements is an essential aspect of choreography. Hence, music is a natural guiding method for motion composition that has been explored in multiple ways [Shiratori et al. 2006; Kim et al. 2003]. Shiratori et al. [2006] perform a music and motion feature analysis (the latter being based on human emotional aspects inspired by Laban) and synthesize a new dance to match an input music evaluating similarities. Kim et al. [Kim et al. 2003] extract rhythmic patterns from a motion capture database and use them to generate a *movement transition graph* to synthesize new motions that match a musical input.

These approaches differ from ours because they use unlabeled motions whose rhythmic patterns are unknown. In our work, however, we propose a system that not only synthesizes motions from the specifications of a choreographer, but also incorporates the input of the dancers and musicians. It is well known that dancers' performances are naturally guided by rhythmic audio signals [Blom and Chaplin 1982]. Hence, we capture dancers' performance while they are dancing to a known piece of music and use the synchronization of the two signals to segment the data according to the musical phrases. We also allow dancers to annotate the dance steps, creating a labeled data set that allows the choreographers to design the dance based on a dancer's input.

Computer Authoring Tools for Choreography. Although historically choreographic design has little relationship with technology, it is not uncommon for choreographers to explore tools such as video recording and video editing software to help store design information. To the best of our knowledge the only computational tool that has been designed specifically for choreographer is the Dance Designer [ChoreoPro 2010]. This software was created by a group of choreographers with the intention of assisting planning and documentation. Similarly to our approach, it uses a 2D visualization of the stage for group motion specification and bases the design on the music timeline. It uses video recording and notes to specify the dance steps and prints a storyboard as an output.

Our system differs from this approach in several qualitative ways. First, our tools are integrated, meant to be used not only by a choreographer, but also by dancers and musicians, promoting a collaborative creation process. Second, we explore intuitive resources that have been endorsed by professional choreographers. For example, we suggest a description for group motions that allows choreographers to specify formations and trajectories from rules, instead of having to manually specify the path of each dancer. We also suggest new forms of choreographic specification drawing ideas from crowd control. Finally, we render a 3D visualization of the motion instead of using notes and video recording. This is useful not only for notation purposes but also for better visualization.

3 System Overview

Our authoring tool involves dancers, musicians and choreographers to create a virtual dance performance. We present the pipeline of the system in Figure 2.

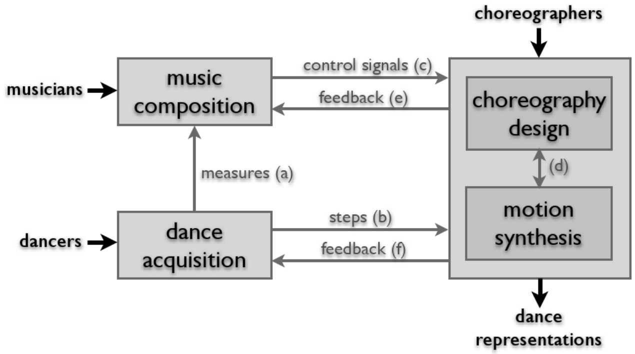


Figure 2: *System pipeline. The musicians are responsible for providing an annotated soundtrack. We use the musical measures to guide the dance (a), and have the dancers create a data set of steps using motion capture and annotations. Both this data set (b) and the music (c) are used as input to the choreographer's interface. The choreographers design the dance and the group motions. We synthesize the resulting motion, generating visualizations (d) and a data representation (output). The system also includes two feedback loops: one for specifying steps that should be performed by the dancers (e), and another allowing choreographers to suggest modifications in the music composition (f).*

The process typically starts with a musician composing a soundtrack for the show. He also creates a series of annotations that mark musical segments (e.g., the chorus) and events (e.g., significant notes or moments in the music). These control signals are important because they serve as a bridge between musicians and choreographers, allowing melodic information to be synchronized with movements. These control signals are displayed in the choreographer's interface, as a guiding timeline. Choreographers are also allowed to make modifications to these control points (e.g., suggesting that parts of the music should be longer or shorter). This is used as a feedback to the musician, who can then iteratively modify the song to match the dance.

We explore the rhythm of the music in several ways. We observe that songs are structured (organized in measures) and dance steps usually have the same duration as an exact number of measures. In accordance with this, we use a discretized timeline as the foundation of our interface, as shown in Figure 3. This timeline allows choreographers to plan the elements of the dance show using the musical structure described above.

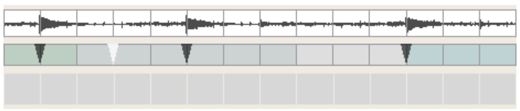


Figure 3: *Musical reference. From top to bottom: representation of the music segmented by the measures; control signals; the authoring timeline (i.e., blank area that choreographers fill in with specifications).*

We also use the rhythm of the music to guide the acquisition of dance steps. As previously discussed, we use a motion capture system for acquiring dancers' movements. In order to synchronize the dance with the musical rhythm, we capture the movements of the dancers while they perform to the music or to a "tack-tack" audio signal that counts the musical beats. With an accurate synchronization, we can segment the data into motion sequences that correspond to musical measures. Although the purpose of this segmentation is to facilitate the combination of motion sequences in an application that makes extensive use of musical references, this structure can also be explored for motion editing, as will be discussed in Section 5.2.

In addition to this segmentation, we also have the dancer annotate the data, specifying to which motion step a given sequence refers. The final structure of our motion database is a set of annotated clips, in which each clip has an integer number of measures and corresponds to a specific dance step. It is important to point out that steps may be recorded several times in order to capture variations (e.g., the same step my be performed while the dancer stays in the same spot or moves along the stage). These variations are also annotated and will be used during motion synthesis.

Figure 4: *Dance annotations: the motion sequence is segmented into blocks that correspond to musical measures and the dancers annotate them, specifying which step they correspond to.*

The choreographers coordinate how the input of the other artists should be combined and therefore assume a central role in dance show creation. They design how dance steps are sequenced and how multiple dancers interact on stage (determining what we refer to as *group motions*). Describing this design is difficult since there is no existing standard notation for it. Hence, we developed an appropriate representation and an authoring tool for designing and editing choreographic input. We discuss these in further detail in Section 4. We use these specifications to synthesize the resulting motion, generating 3D visualizations of the dancing characters (see Section 5). The data representation includes high level specifications of the dance that permits iterative editing and also full descriptions of the final synthesized motion (bvh file format) that can be exported to other animation software packages, such as Maya. We also allow instructions to be given to the dancers, suggesting specific dance steps that should be performed. This feedback is given whenever the synthesis algorithm is unable to generate the motions specified by the choreographers due to the absence of data.

4 Interface for Choreographers

In this section we will explain in greater detail the tools we developed for choreographic input. We will first give a general overview of the components of the interface. Then, we will give further detail on how to specify group motions, by discussing representation methods and specification tools.

4.1 Overview of the interface

As previously mentioned, the choreographers should be able to specify both the sequences of dance steps and the group motions. Although these two elements are closely related, the design tools used for each of them are quite different. Hence, our interface has two modes: one for step sequence specification and one for group motion specification. Each mode has a different set of authoring tools. We stress, however, that both elements are synchronized with the music and therefore, using the music as a guideline for both interface modes, we allow the two elements to be naturally matched.

Step Sequence Specification. In terms of representing individual motions, it is a common practice to video record the performance of the dance steps in order to be able to reproduce them later on. Based on this idea, we have suggested presenting a choreographer with a list of names of the captured dance steps, which are determined by the dancers' annotations. By clicking on any one of these names, the choreographer can preview an animation of the corresponding motion (see Figure 5).

Figure 5: *Interface for dance steps mode. Center: preview of the captured step. Right side: list of steps in the data set. Bottom: timeline with music information and sequence of steps.*

Hence, choreographers can specify dance movement at each measure by filling the boxes on the authoring timeline using a drag-and-drop motion from the list of steps. It is important to point out that we should be able to specify dance steps to groups, subgroups or individual dancers. For this reason, we developed a hierarchical structure that allows the group to be divided and subdivided based on individual identities (such as gender, dance expertise) or positions on stage. We make these segmentations visible in the interface by subdividing the timeline.

Group Motion Specification. The group motion mode of the interface has a similar layout, as shown in Figure 6. We observe that group motions are traditionally specified with images of a top–down view of the stage in which dancers are represented as circles (often color coded) and their trajectories are indicated with lines or arrows. Commonly, these images are stacked together creating a storyboard that indicates a sequence of motions. Based on these principles, we have created a similar computer interface for specifying group motions. Our visual interface has a component that represents the stage and where choreographers can position dancers and determine their trajectories.

Analogous to the dance motion mode of the interface, we place in the bottom a component with a musical reference and its authoring timeline. However, instead of dance steps, the choreographers should fill in the timeline with elements that show how the positions of the dancers evolve in time. Also, instead of a list of steps, we have a list of *formations*, i.e, specifications of how the dancers are distributed on stage at a given time. We will discuss tools that allow choreographers to make these specifications in detail in Section 4.3.

While the sequence of dance steps can be described and stored as a list, classification and representation of group motion elements is somewhat more complex. Hence, in Section 4.2, we will describe in greater detail the language we propose for group motions.

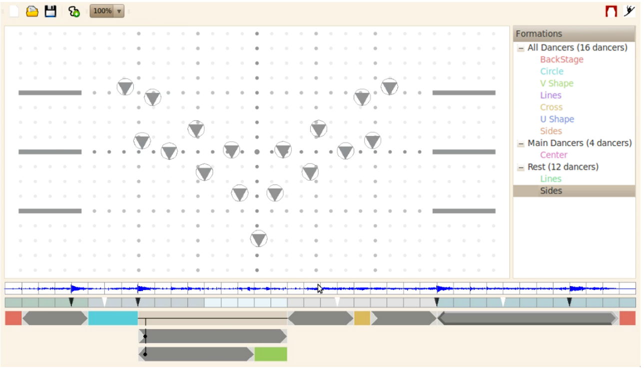


Figure 6: *Interface for group motions mode. Center: 2D view of the stage, where the dancers are represented by circles and triangles that indicate positions and facing directions, respectively. The grey rectangles represent the entrances to the backstage. Right side: list of formations that were designed by the user. Bottom: timeline with music information and motion group evolutions.*

Visualization. An important aspect of the authoring tool is to allow choreographers to pre–visualize the designed dance. Hence the interface includes a preview the group motion by playing an animation of the resulting 2D movement of the dancers (see Figure 8) and a visualization of the resulting 3D animation (see Figure 7)

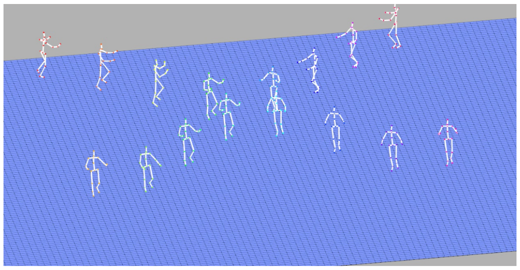


Figure 7: *Visualization of the 3D motion used for previewing the design.*

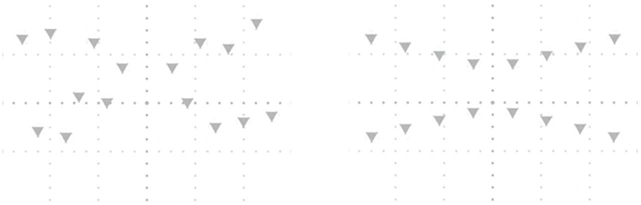


Figure 8: *Visualization of the 2D group motions at two different frames. Dancers are represented by circles and triangles that indicate positions and facing directions, respectively.*

4.2 Representation of Group Motions

Due to the lack of an existent representation for group motion, we decided to propose a new vocabulary for describing dance. We approached this problem by drawing information from conversations with dancers and choreographers, analysis of dance performances, as well as previous work on dance descriptors and motion synthesis.

4.2.1 Analysis

The proposed a classification for group motion modeling is illustrated in Figure 9. We base our classification on the nature of the specification. This specification can be: spatial, temporal or categorical. The spatial elements of group motions refer to the way dancers are positioned and distributed in space at a given time, creating formations on the stage. Temporal elements indicate ways of specifying how the dancers' positions evolve in time. This is done by designating the evolution conditions and rules. Finally, categorical elements, refer to different ways of rearranging the group of dancers into smaller subgroups and creating dependencies between them.

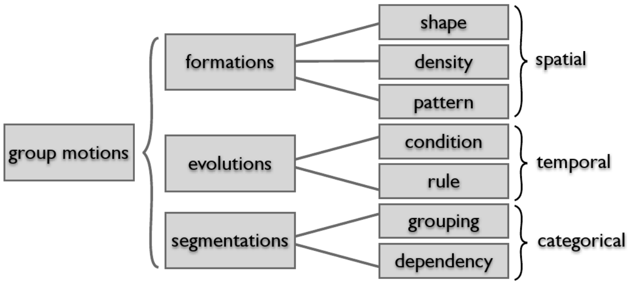


Figure 9: *Classification of group motion elements used for design specifications.*

We interpret group motion in three levels of abstraction: *motion intention*, *motion segment*, and *locomotion*. *Motion intention* refers to the essential identity or objective of the motion (e.g., assuming a given formation or following a given trajectory). Though it contains the essential specifications of the motion, it lacks the contextualization as part of the overall dance. We create a *motion segment* by taking *motion intention* specification and determining how it fits in the dance timeline (e.g, for how long the dancers assume a given position or trajectory, to which subgroup of dancers these specifications are related to, etc.). Finally, after a sequence of *motion segments* is combined, we can determine the actual *locomotion*, i.e., the 2D stage position and orientation of each character at each frame.

4.2.2 Data Structure

The *locomotion* information is stored for each dancer as the list of 2D positions and orientations, and it guides the synthesis of the articulated motion. Although this level contains all necessary information for motion synthesis we have chosen to create a data structure that aggregates all three abstraction levels. This allows us to distinguish between different inputs and therefore permits the data to be reused and edited.

The most essential element of our data structure is a list of *motion segments*. Each motion segment refers to an *evolution* (see characterization in Figure 9) and can, therefore, be of one of the three types: fixed formations, boundary conditions, and initial conditions. These motion segments appear in the interface on the authoring timeline for group motion specifications, as shown in Figure 10. The data structure of each motion segment includes the following informations: type, reference to a group of dancers, set of attributes (conditions, rules, etc.) and a reference to its position in the timeline.

In most cases, it is simpler to view *motion intentions* as attributes of motion segments (e.g., a motion segments that describes a trajectory has the path as one of its attributes). However, since formations are one of the essential elements in choreographic design, we store them separately, allowing them to be repeated effortlessly during the dance. Hence we also include in the data structure a list of formations. As previously discussed, the formations designed by the choreographers are displayed as a list on the user interface (see Figure 6).

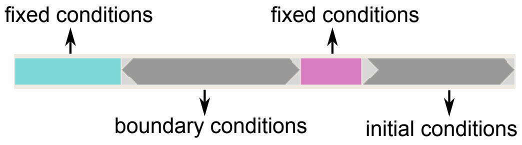


Figure 10: *Example of how evolutions are specified using the authoring timeline.*

The final element of our data structure is the hierarchical representation of the different groupings. We allow the group of dancers to be divided and subdivided in different ways, and store for each group a list of IDs of the dancers. Notice that we allow such grouping to be made for both group motions and dance step specifications. However, the groups do not have to be the same (i.e., choreographers can specify the same dance step but different group motions to different parts of the group and vice-versa).

4.3 Tools for Authoring Group Motion

In oder to propose a natural environment for dance authoring, we developed tools that take into account traditional design methods and permit *declarative* specifications. In addition, we argue that an advantage of transferring the creative processes to the computer is that this allows us to explore different computational techniques that suggest novel creative expressions. To illustrate this, we draw ideas from related work in behavioral animation to suggest *procedural* tools for dance design. In what follows, we will discuss both the declarative and procedural specification methods in greater detail.

4.3.1 Declarative Tools

As previously discussed, group motion declarations involve specifications of formations and evolutions (i.e, *motion segments*). Next, we will describe the tools provided in the interface that allow choreographers to make such specifications.

Formations. We specify formations by determining its shape, density, and pattern. Shape refers to the geometry that a group of dancers assumes on the floor that can be designed with a sketch input; density refers to the number of dancers that will occupy a given stage area; and pattern refers to the way dancers are distributed on designed shapes (see Figure 11). We also allow a choreographer to specify symmetries by drawing a symmetry line and reflecting elements that are drawn on either side.

Evolutions. Evolutions are created by determining conditions and rules. We consider three types of group motions: motions based on fixed conditions, boundary conditions, and initial conditions.

We determine a *fixed condition* by referencing a formation and determining how it fits within the timeline.

Movements based on *boundary conditions* are set by determining initial and final positions and creating a group motion that allows them to naturally evolve from one to the other. There are three basic steps in creating such group motions: matching the two sets

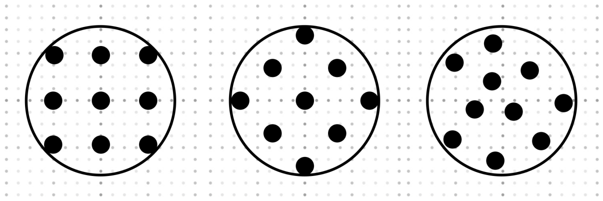


Figure 11: *Examples of different patterns on a circle. From left to right: uniform grid, diagonal grid, and Poisson sampling.*

of positions, synthesizing a trajectory, and determining orientations of dancers (i.e., the directions they face at each step).

In our system, we use a bipartite matching algorithm [Diestel 2010] for matching the initial and final positions. Each vertex in the graph refers to an individual position on stage, the vertices in the first set referring to the initial positions and the vertices in the second set referring to the final positions. The cost function being optimized is determined by the weights of each edge. We can set these weights to be equivalent to the distance between the two positions and have the algorithm return the matching that guarantees that, if the dancers travel in a straight line, the sum of all the trajectories will be minimized:

$$W_{\text{overall minimal distance}} = \|P_i - P_j\|\,,$$

However, depending on the scenario, it may be more important to guarantee that the distance traveled by the individual dancers are optimally equivalent. To implement this, we should specify the weight of each edge as the norm of the difference between the distance traveled with this matching and the average distance traveled by the dancers:

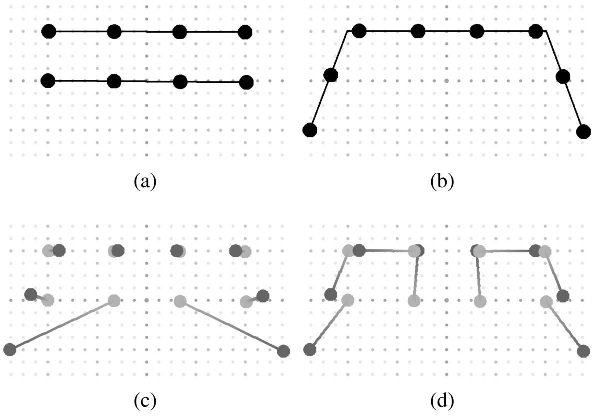
$$W_{\text{equal distances}} = \left\|W_{\text{overall minimal distance}} - \bar{D}\right\|\,,$$

where $\bar{D}$ is the average traveled distance. Calculating the actual average distance is challenging since it naturally depends on the chosen matching. However, we can approximate it by taking the average of the distances traveled when we choose the matching that results from optimizing the overall minimal distance. Empirically, we have observed that this yields good results, as can be seen in Figure 12. We create trajectories by connecting the two positions with a straight line and then correcting the paths using a collision avoidance method based on [Perlin 2004]. Finally, we specify orientations by guaranteeing that characters face the path during the locomotion or by linearly interpolating the initial and final specified orientations.

Movements based on *initial conditions* and evolution rules allow choreographers to plan complex movements based not on the final positions, but on the actual motion effect that they desire. An example of an evolution rule is a sketch of a trajectory on stage (see Figure 13). More complex evolution rules include ones that determine parameters and simulations algorithms, as will be discussed in Section 4.3.2.

4.3.2 Procedural Tools

Our authoring tool also provides behavioral animation methods [Reynolds 1987] for specifying evolution rules. These methods regard each dancer as an autonomous agent that travels along a 2D manifold represented by the stage according to combined steering forces. Choreographers can simulate new behaviors that allow agents to reach higher level goals by creating several types of forces and by combining them in different ways. Innumerable evolution rules could be designed based on these behaviors. For the purpose of illustrating the applications of this method, we have implemented (using the OpenSteer library) the following procedural methods: following attraction/repulsion forces, spreading out on the stage, and crossing over, as shown in Figure 14. In this case, choreographers position corresponding *structural elements* (e.g., an attractor) on stage in order to create evolution rules, as shown on Figure 15. Choreographers also create evolutions specifying for how long each of these rules is active in the authoring timeline.



(a) (b)

(c) (d)

Figure 12: *Examples of evolutions based on boundary conditions. The choreographer specifies an initial formation (a) and a final formation (b). Each ball indicates the position of a character on stage. Evolutions are synthesized based on different optimization specifications: overall minimal distance (c) and equal distances (d). The lines indicate the trajectories followed by each dancer. Notice how different rules result in different motions.*

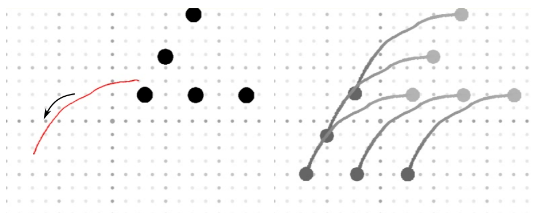


Figure 13: *Example of an evolution based on initial conditions: evolution rule defined by the sketch of a trajectory on stage (right) and resulting group motion (left).*

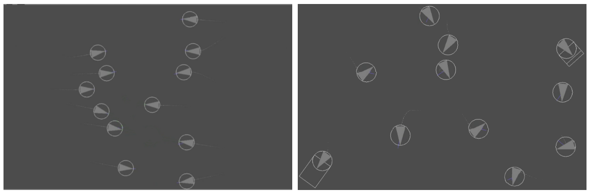


Figure 14: *Example of procedural evolution rules: crossing over (left) and spreading out on the stage (right).*

5 Motion Synthesis

Dance is an interesting type of motion because its structure is difficult to describe. It is different from other common types of motion,

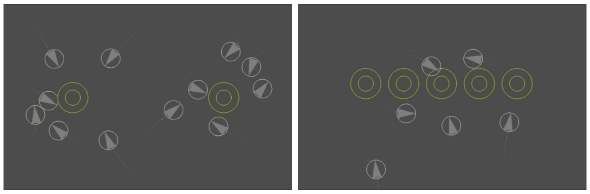


Figure 15: *Example of multiple procedural elements: two attractors (left) and multiple attractors (right).*

such as walking, lifting, or sports movements, which can be associated with specific intentions. These motions have deterministic objectives and are, therefore, simpler to reproduce using physical models that optimize goal, balance, and energy. Dance, on the other hand, cannot be interpreted as an optimization of any kind, but is at the same time not at all random, since each nuance of the movement is important to the resulting *expression*.

We have chosen to use a data driven system both because this allows strong interaction between dancers and choreographers and because we believe that the nature of the dance movements makes this technique more efficient than physically based animation methods. We have chosen to explore a motion graph method [Kovar et al. 2002b] for creating new motions from a collection of MoCap data. We have extended the approach proposed in [Kovar et al. 2002b], in order to incorporate the rhythmic structure into our model for motion synthesis. Although a straightforward use of this technique to our problem would create a plausible motion, it would not create a plausible dance, since it would combine pieces for motion of random duration which would not likely fit into any rhythmic structure.

We propose a *structured* and *measure-synchronous* motion graph. As shown in Figure 4, we segment the captured data according to musical measures. Hence, we create a *measure-synchronous* motion graph by, searching and connecting similar sequences of frames at the end of each measure. With this, we guarantee that even a random walk on this graph will result in a motion that is coherent with the musical metric. We create a *structured* motion graph by annotating each set of measures with the name of the dance step it corresponds to, as shown in Figure 4. With this structure, we can represent the motion graph as a set of clusters, where each cluster indicates a group of instances of the same step, as shown in Figure 16.

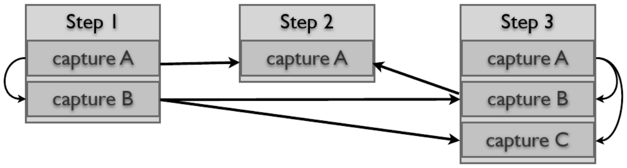


Figure 16: *Structured motion graph created by clustering example of the same step.*

Notice that since the same step may be performed in many different ways, different motion clips within the same cluster may be connected in different ways to different steps. Therefore, the greater the number of instances of the same step we capture, the greater the connection between the graph clusters. As previously mentioned, dancers annotate motion capture segments not only with the step names, but also with information about the step variations. These variations may be in: style, structure and locomotion. Stylistic variations refer to variations that significantly alter the motion in a way that it may be appreciated by untrained spectators as a different dance step (e.g., changes in the upper body motion). Structural changes correspond to modifications that are essential for graph connectivity (e.g., when the same step is performed with the left or right foot or when the beginning and finishing poses are changed to allow different step sequences). Finally, locomotion variations refer to changes in the path that the dancer follows while performing a given motion.

We take advantage of this structure both for synthesizing new motion sequences by combining the captured data (Section 5.1), and for developing motion editing tools based on signal processing (Section 5.2).

5.1 Combining Motion Segments

As previously discussed, choreographers determine a sequence of steps that each dancer should perform and the trajectory that they should follow on stage. Hence, to synthesize the resulting motion, we need to develop methods for seamlessly combining steps in the proper order while guaranteeing that the specified trajectories are followed. In what follows, we will describe these techniques.

5.1.1 Sequencing Dance Steps

When it comes to sequencing dance steps, structural variations are the only ones that are relevant. Since stylistic variations significantly alter dance design, we assume that they should be specified by the choreographers. Therefore, we simply interpret them as different steps and create new clusters to represent them. Locomotion variations, on the other hand, do not alter the connectivity properties of the motion sequences. As a result, we can group different locomotion variations that correspond to the same structure and create *structural components*, as shown in Figure 17. We can therefore create a new graph (which is a simplification of the original motion graph), where each node represents a different structural variation. Sequencing the dance steps is equivalent to finding an optimal path on this graph that matches the choreographers' description (i.e., the order of the visited clusters should correspond exactly to the specified order of dance steps).

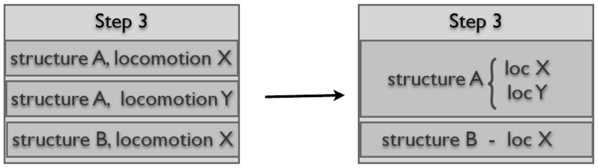


Figure 17: *Structural components within a step cluster: created by grouping examples with the same structural variations but different locomotions.*

We use an exhaustive search algorithm that traverses the entire graph by expanding and examining each node in the graph, producing a *spanning tree* of the nodes reached during the search. We set the cost for each path to be equal to the number of nodes that do not match the choreographers' specifications, and we use the spanning tree structure to find the sequence of visited nodes that has zero cost. We stop expanding a node when its depth in the spanning tree is equal to the size of the specifications. We can also prune the spanning tree whenever we reach a node that corresponds to an unmatched step. We can guarantee that the cost of any subsequent path will be greater then zero.

The effectiveness of our motion synthesis method is highly dependent on the captured data. For instance, the choreographer can specify that step A is followed by step B and this may not be possible if we have not captured the necessary variations of the steps that allow the two clusters to be connected. Instead of allowing suboptimal synthesis, we have explored the fact that our platform is integrated and that the choreographer can give feedback to the dancer while they are designing the choreography. Therefore, we have suggested a system for informing the artists when a given specification cannot be accurately synthesized and what kind of input they should provide to resolve the problem. In this way, we argue that our synthesis technique will always find a solution and that this solution will be optimal in the sense that it will fit the exact specification.

5.1.2 Following Trajectories

After we find a sequence of structural components, we should choose a locomotion variation within each component to permit dancers to follow desired steps. Notice that choreographers' specifications conform to a timeline that is discretized according to musical measures. Therefore, the specified durations of each path are also measure-synchronous and we can create a list of positions and orientations for each individual dancer. The locomotion can be represented as a discrete list $S(n) = (x_n, y_n, \theta_n)$, for $n = 1, \ldots, N$, where (x_n, y_n) and θ_n are, respectively, the position and the orientation at the end of segment n. With this discretization, we can search for the optimal locomotion at each measure. It is important to emphasize that we use the desired final position at the end of each measure and not the distance that the character should travel at each measure. This is done to guarantee that small errors resulting from restrictions in the database do not add up, creating a trajectory that diverges from the specification.

To allow characters to follow trajectories, we make use of two editing mechanisms that will be discussed in the next section in further detail. The first tool involves combining through interpolation different locomotions variations within the same component to allow, for example, a diagonal path to be synthesized from a forwards and a sideways motion. We also allow the dancers to rotate while performing a single step in order to satisfy the orientation specifications.

Hence, we start by significantly expanding the database, interpolating clips that correspond to different locomotion variations. Then, we take each of these clips and calculate their total stage locomotion when the desired rotation is inserted and choose the one that best matches our solution. Finally, we also use the feedback system to inform the dancers if new locomotion variations are needed during motion synthesis.

5.2 Motion Editing Tools

One of the advantages of the structure that we have developed is that it allows us to use simple motion editing tools based on signal processing techniques to obtain realistic results. We have explored these methods for the following motion modifications.

5.2.1 Interpolations and Combination

As previously mentioned, we interpolate locomotion variations to allow dancers to move in different lengths and directions while performing a single step. We also explore the recombination of upper and lower body motions that come from different stylistic variations.

The greatest challenge with interpolating and combining upper and lower body motions is that the results are not seamless unless there is a reasonable alignment between the two motion segments, and this alignment may require nontrivial warpings. We argue that since we have clips of the same dance step with equal durations, we guarantee that they are trivially aligned by construction. Our experiments show that this is, in fact, the case.

5.2.2 Rotations

Part of the trajectory specification is the orientation of the dancers at each measure. We propose a method for allowing the characters to rotate while performing a step that tries to minimize foot sliding artifacts. We observe that, in a walking step, rotations occur when only one foot is on the ground. Therefore, our approach analyzes the foot contact, selects the segments in which there is only one foot touching the floor and applies a rotation and translation transform to the motion sequence that changes the orientation of the dancer while maintaining foot contact. Figure 18 illustrates this procedure. Notice that we are editing the motions at each measure. This is important because if we had a large motion clip we would have to select a certain moment or a certain movement to apply the transformations. However, since we know that a whole sequence corresponds to a specific dance step, we can distribute the rotation angle uniformly between all selected frames.

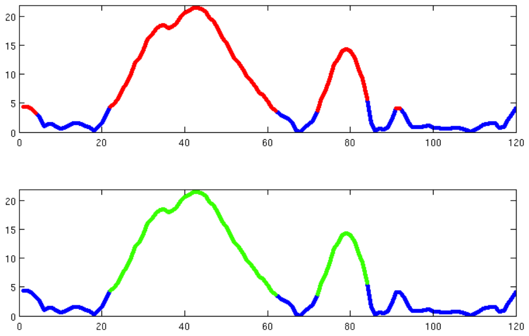


Figure 18: *We calculate the distance between the two feet and select the regions where this value is greater than* 20% *of the maximum value (top). Then, we explore the fact that steps tend to be smooth to ignore all the segments that we consider too small (we used 10 frames as a threshold) and therefore are able to eliminate the small errors (bottom).*

5.2.3 Variations

When replicating the movements of a single dancer to synthesize multiple characters performing on the stage, it is essential to make some modifications of the data to make the group dance look natural. We have explored time warping mechanism for desynchronizing the data and amplitude variation methods for varying the upper body motion. We chose to apply these tools only to the upper body part because they are less susceptible to undesirable artifacts, since there are no floor contact restrictions.

6 Results and Discussion

In this section we validate the proposed system discussing examples and user evaluation results. We also discuss the system's limitations and point directions for future work.

6.1 Experiments

We have conducted a number of experiments with the help of a dancer, a choreographer, and a musician. These experiments gave us the opportunity to test each component of the authoring pipeline.

The resulting animations are available in the supplementary material.

To explore the synthesis of a combination of dance steps, we have asked the choreographer to design an individual dance based on a music of her choice. She selected four Ballet steps and performed them using the musical measures for timing. Each step has been captured multiple times with different structural and locomotion variations. The structural variations have included the performance of the same step starting with the right or left foot, and also altering the beginning and finishing poses of the steps from a *sous-sur* (rising up) to a *plié* (bending of the knees). Finally, she has proposed a choreography and the corresponding motions have been synthesized by sequencing the steps accordingly. The *structured* and *measure-synchronous* motion graph guaranties that the combinations are seamless and synchronized with the music.

To validate our system's ability to allow standard group motion specification, we have chosen a segment of an existing dance show and used the interface to replicate the corresponding group dynamics. For simplification, we have used a single dance step that has been captured with many locomotion variations. With just a couple of formation specifications, and calculating the intermediary motions based of the boundary condition specifications, we have managed to synthesize a new dance that very closely resembles the original piece.

Finally, we have experimented with procedural techniques. First, we have defined four procedural elements that the choreographer can use to define the group motions: attraction forces, repulsion forces, commands for spreading out on the stage and directions for crossing-over. The choreographer then designed a first version of the choreography defining a how these elements should be sequenced in the timeline. The musician used this information to create the first version of the song, which was used to guide dancer's performance. With the feedback from the simulations, iterative editing was made to insert new procedural elements to the timeline and also modify their duration. This process altered the control signals that were then used by the musician to recompose the music in order to match the resulting dance.

6.2 User Evaluation

To validate the proposed platform, we invited a group of eight dancers and choreographers to analyze our system. We had them discuss every aspect of the proposed pipeline and give input on the design decisions. In addition, we had a group brainstorming session where we discussed the possible applications, extensions and limitations of the authoring environment.

The artists validated both the representative language we proposed and the system design. In general they found the interface intuitive, and approved the separation between dance movements specifications and group motions. They also validated the completeness of the movement analysis described in Figure 9. They considered this description of group motions intuitive for creative specifications. With regards to the interface, they acknowledged that for most standard show, representing the stage as a 2D plan is a valid simplification. However, the choreographers that work with contemporary dance argued that they like to explore the 3D space, e.g., allowing dancers to be suspended in the air on strings. They also stressed the importance of allowing staging design and specification of interaction with objects.

In general, the artists gave positive feedback on the system and they were enthusiastic about the applications, which would range from small amateur recitals to large professional performances. They confirmed that documentation takes a lot of manual work and that a better alternative would make them significantly more productive. They were specially excited about the visualization tools. One of the reasons for this is that, in general, dancers only get to practice a few times in the actual theater, while most rehearsals are done in small dance classrooms. This makes planing group motions more difficult because choreographers have to somehow guess how the movements will look like on the actual stage and whether the dancers will have enough time to go from one end of the stage to the next.

The choreographers were also particularly excited about how the system suggests greater collaborations among different artists. They commented that, while choreographers in the past were responsible for the design of every aspect of the dance show, their role is progressively changing. In recent works, choreographers tend to assume the position of editors instead of composers, and the dancers' role in the design has become much more fundamental. In this context, a system that stimulates interaction between artist would be most welcome.

6.3 Limitations

One important aspect of dance shows that this work does not address is physical interaction between the dancers. These interactions are quite common and can be very complex, specially in contemporary dance. In order to allow such interactions to be designed in this authoring environment it is necessary to develop a descriptive language for interactions as well as corresponding specification tools. Furthermore, in order to synthesize motions with interactions it is necessary to develop new tools for leveraging motion capture data.

Our system only considers group motions in 2D. However, some staging designs include object that dancers can climb on top of or crawl under. To allow design of such interactions, it should be necessary to expand the group motion specification to allow positions to be determined on the 3D stage. In addition to expanding the representation, new tools for designing group motions would have to be developed.

7 Conclusions and Future Work

In this work, we have designed an authoring and collaboration platform for dance shows and discussed the relevance of this research to dance design. We also studied the technical aspects related to both dance and choreography specification and synthesis, and developed tools that were sufficient to demonstrate the proposed concepts. Finally, we illustrated the applications of our method with experiments performed with dancers and musicians and validated the approach with user evaluations.

The authoring platform is complete enough to be used by dance professionals. However, since it has a simple design and includes default specifications, it can also be used by non-experts. One of the interesting applications for this system is learning, since it allows students to create full dance shows and have a visual output of their design without the need to hire actual dancers.

As a future work it would be valuable to address the current limitations of the system as described in the previous section. It would also interesting to apply the system to live scenarios in which conception and execution are done simultaneously. This would require an instant feedback that fully integrates the collaborations of dancers, choreographers and musicians. An example of this would be a performance that combines live and virtual dancers projected on stage, whose movements are guided by the combination of inputs from different artists and possibly spectators.

References

BLOM, L. A., AND CHAPLIN, L. T. 1982. *The Intimate Act Of Choreography*. University of Pittsburgh Press, Pittsburgh, USA.

BLUMBERG, B. M., AND GALYEAN, T. A. 1995. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 47–54.

BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 183–192.

BRUDERLIN, A., AND WILLIAMS, L. 1995. Motion signal processing. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 97–104.

CHOREOPRO, 2010. Dance designer. `http://www.choreopro.com`.

DIESTEL, R. 2010. *Graph Theory*. Fourth electronic edition.

ELLFELDT, L. 1988. *A Primer for Choreographers*. Waveland Press, Inc., USA.

KIM, T.-H., PARK, S. I., AND SHIN, S. Y. 2003. Rhythmic-motion synthesis based on motion-beat analysis. In *ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, SIGGRAPH '03, 392–401.

KOVAR, L., AND GLEICHER, M. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3, 559–568.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 473–482.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In *SIGGRAPH '02*, ACM, New York, NY, USA, 473–482.

LABAN, R. V., AND ULLMANN, L. 1960. *Mastery of Movement*. Princeton Book Company Publishers., USA.

MARSHALL, S., AND LEONARD, N., 2010. Flocklogic. `http://www.princeton.edu/~flocklogic/`.

PERLIN, K., 2004. Path planning. `http://mrl.nyu.edu/~perlin/experiments/path/`.

REYNOLDS, C. W. 1987. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, 25–34.

REYNOLDS, C., 1999. Steering behaviors for autonomous characters.

SHIRATORI, T., NAKAZAWA, A., AND IKEUCHI, K. 2006. Dancing-to-music character animation. *Computer Graphics Forum 25*.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, 43–50.

WITKIN, A., AND POPOVIC, Z. 1995. Motion warping. In *SIGGRAPH 95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 105–108.