ChoreoGraphics: An Authoring Environment for Dance Shows

Adriana Schulz Luiz Velho
VISGRAF – IMPA, Rio de Janeiro, Brazil

***Abstract*—This work[1] describes an authoring environment for dance shows that can both be used as a mechanism to synthesize virtual performances and as a tool to assist the planning of choreographies, allowing design and visualization of full motion sequences. To this end, we explore adaptations of techniques for leveraging motion capture data to an application that makes extensive use of musical references. We also discuss methods for controlling the way dancers move on stage as a group, creating formations and following trajectories. Finally, we propose an integrated platform, which suggests a new form of collaboration between artists, allowing the show to naturally evolve from iterative contributions of dancers, musicians and choreographers. In this way, we indicate how graphics technologies can be used not only to facilitate creation but also to suggest new forms of artistic expressions.**

***Index Terms*—authoring; animation; motion capture.**

I. Introduction

A. Motivation

The widespread of Motion Capture systems has had a very significant impact on the area of computer animation, as technologies allow the combination of real actors with virtual ones that are synthesized from captured performances. We observe, however, that most applications are still very naive, demanding a lot of work from the actors, whose performances are directly used to drive the movements of the virtual characters, and from the artists, who manually manipulate the data.

On the other hand, many recent papers have proposed efficient techniques for manipulating MoCap data, allowing a large number of different motions to be generated from a relatively small database and with very accurate control. Related work includes not only movement editing [1], but also synthesis of new streams of motion from previously acquired data, allowing creation of new and more complex motions. Motion synthesis strategies include constructing models of human motion [2], interpolating motion to create new sequences [3] and reordering motion clips employing a motion graph [4].

Methods for data driven animation suggest a new paradigm for content creation, allowing the synthesis of sophisticated and realistic human motion with very little effort from artists. However, in order for this revolution to in fact take place it is essential to adapt these techniques to real scenarios and develop authoring environments that serve as a bridge between artists and the emerging technologies.

Although applications of these resources are mostly observed in movies and have so far been little explored in other media, recent trends indicate that artists will also begin to use them creatively to produce new visual effects in dance shows, concerts, musicals and other spectacles. In this context, we propose an adaptation of the existing methods for leveraging MoCap data to the context of creating dance content from artistic input and propose an authoring environment for dance shows. With this, we advocate how graphics technologies can be used not only to facilitate creation but also to suggest new forms of artistic expressions.

[1]This is a short version of a Masters' thesis presented to IMPA in July 2011 (http://www.impa.br/∼aschulz/ChoreoGraphics).

B. Contributions

The contributions of our work are related to both the structuring of the authoring platform and to the development and adaptation of techniques for motion synthesis.

1) The Authoring Environment: To the best of our knowledge, we are the first to propose an authoring and collaboration platform for dance shows that integrates the creative elements that compose a choreography: music, dance and group motion. This platform can both be used as a mechanism to synthesize virtual performances and as a tool to assist the planning of dance shows, allowing design and visualization of full motion sequences. Though it is highly applicable in conventional shows, guiding the artists through conception, production and execution, it also suggests a new paradigm for content creation, allowing the show to naturally evolve from iterative specifications of different artists.

2) Dance Motion Synthesis: Dance is a very interesting type of motion because its structure is very hard to describe and, therefore, to reproduce using physical models which optimize, goal, balance and energy. Hence, we have chosen to use a data driven system both because we believe that the nature of the dance movements make this technique more efficient and because it permits a strong interaction between dancers and choreographers.

One of our most significant contributions is that we approach the problem of analyzing dance motion by exploring its strong relation and synchronism with music, which is a very structured signal. We take advantage of this for segmenting the motion and creating a structure that not only facilitates dance design, but also makes editing tools simpler and more efficient.

Another aspect that we explore in this work is how we can take advantage of the platform for efficient data acquisition. Since we suggest iterative contribution of artists, the database constructed by the dancers is guided by the choreographers' specifications. Hence, we are able to have a very broad application (that can be used for practically any dance style) without demanding large amounts of data.

3) Group Motion Synthesis: Another interesting aspect of dance shows is the visual effects generated by the combination of the individual performances, i.e., the way the dancers move

on stage as a group, interacting, creating formations and following trajectories.

Group motion can be regarded as the movement of particles on the 2D plane and, therefore, is by itself an interesting animation problem, which has some very interesting related work. Thus, in this work, we also explore not only declarative ways of positioning the dancers on stage, but also behavioral animation methods for controlling group motions.

II. Overview

Dance shows are becoming evermore collaborative and the responsibilities of each artist evermore fuzzy. In this context, we propose a platform for designing content of dance shows which covers all elements of the creative process and promotes the communication between different artists, allowing shows to naturally evolve from iterative contributions. Our creative tool permits planning, editing and visualizing dances, thus guiding the artists through conception, production and execution. This improves the quality of the results significantly, since, equipped with a system that allows easy input and instant feedback, the artists can concentrate uniquely on their craft.

Still, once we transmit the creation process to the computer, we are offered a whole new range of tools that can not only make dance composition easier and more efficient, but also suggest innovative approaches. In this work we explore, for example, methods for simulating behaviors of autonomous agents in order to control the locomotion of the performers on the stage. These contributions are significant in advancing the state-of-the-art in dance design, but they are still restricted to the same conventional steps involved in its production.

There are, however, some very unordinary ways with which computational tools can be applied, suggesting a whole new paradigm in dance composition. An interesting application of our framework and its extensions are "on stage" productions and improvisations, which would join efforts from choreographers, dancers and musicians in real time. An example is a performance that combines live and virtual dancers projected on stage, whose movements are guided by the combination of different artistic inputs. In such scenarios, the artists would be able to influence not only the virtual dancers' movements, but also one another through a framework of instant feedback.

Ultimately, we can use this system to mediate between performers and spectators. By allowing real time collaborations, we bridge the gap between conception and execution, making it possible for the performance to be produced and presented at the same time. Hence, we can also allow the public to contribute to the performance.

A. Artistic Input

In this work, we concentrated in exploring the contributions of dancers, musicians and choreographers. We assume that the role of the dancers is to determine the steps that will be executed during the performance, while the role of the choreographer is to plan how these steps will be combined or sequenced, and how the multiple dancers will interact on stage (determining what we refer to as *group motions*). The role of the musician is, of course, to compose the show's soundtrack.

A very important aspect that must be analyzed is how each of these artists should input their creative abstractions into the computer. The simplest input is perhaps the musicians', whose music can be represented by an audio signal with several annotations. These annotations consist of the music's segmentation into measures and also control signals that indicate musical events that can be synchronized with dance movements.

Dance specifications is somewhat more complex, since movements have many degrees of freedom which cannot be easily determined by a standard notation. In this work, however, we used a MoCap setup to capture the movement of the dancers, therefore allowing dance steps to be fully determined by example. We also propose methods for storing the captured clips based on annotations made by the dancers. This results in a structured data set that can be used by the choreographer during dance design and also by the system's motion synthesis algorithms that allow visualization of animated dancers.

Finally, it is necessary to determine the choreographers' input. Analogously to dance steps, this is also a very complex information with no existent standard notation that describes it. In this case, however, there is no available set up that allows us to easily capture the choreographer's specifications and transform it into a signal, as is done with MoCap. We also observe that the choreographers' role is much more central to the dance, since they are the ones who usually coordinate how the input of the other artists should be combined.

Hence, we decided to develop an interface for choreographers. To develop such a tool, it is very important to understand the choreographer's process of creation in order to translate it to a language the computer understands. In short, we need to specify and classify the relevant components of a group dance and find a way of making them available for use in our authoring environment. This is somewhat challenging, first because there is no established list of all motion elements that may be present in a group dance and, second, because there is no standard notation that choreographers use to determine the dance steps and group motions. For these reasons, we have decided to approach the interface design problem by first making a careful study of dance elements.

B. Dance Analysis

To propose a new vocabulary for describing dance, we drew information from conversations with dancers and choreographers, analysis of dance performances, as well as previous work on dance descriptors and motion synthesis. It is our intention to validate, in the future, our approach with feedback from other artists.

We observed that authoring group motions involves not only modeling, but also visualization, i.e., determining projection and staging. Both of these aspects should be designed for the different elements that compose the show: dance steps, group motions, music and set design. All of these elements should be specified in parallel and, preferably, in collaborative environment.

Because our main interest is analyzing group motions for designing the choreographers' interface, we propose a classification for group motion modeling illustrated in Figure 1. We base our classification on the nature of specification, which can be: spatial, temporal or categorical.

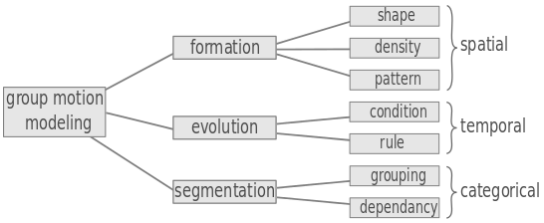


Fig. 1. Modeling elements.

The spatial elements of group motions refer to the way the 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.

III. The Authoring Environment

In this section we will discuss the main functionalities of our framework, that integrates: music, dance and choreography.

A. Music

Songs are mostly very structured (organized in measures)and dance steps usually have the same duration as an exact number of measures. In accordance with this, we make extensive use of the music's rhythmic structure in several aspects of our work. We use a discretized timeline as the foundation of our interface, as shown in Figure 2. This timeline allows choreographers to plan the elements of the dance show using the musical structure described above.

Fig. 2. Musical reference. From top to bottom: representation of the music segmented by the measures; control signals; the authoring timeline (i.e., a sequence of small boxes appear that correspond to musical measures and are filled in with elements that indicate to the choreographers' specifications).

We also use the music's rhythm to guide the segmentation of the MoCap data into segments that correspond to motion 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 IV-A.

In addition to the rhythm, we also explore the music's melody, which is used to determine control signals that should be synchronized with motion events (see Figure 2). These signals can either be extracted from a selected music and used as a reference to guide the dance, or can be iteratively edited by both the musician and the choreographer in a scenario where the show is designed in a collaborative effort of both artists.

B. Dance

As previously discussed, we use a motion capture system to acquire the dance motions. In order to be able to synchronize the dance with the musical rhythm, we capture the movements of the dancers while they perform to the music or 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.

In addition to this segmentation, we also have the dancer annotate the data, specifying which motion step the sequence refers to. The final structure of our motion database is, therefore, a set of annotated clips, where each clip has an integer number of measures and corresponds to a specific dance step. It is also important to point out that each dance step may be captured several times if the dancers want to perform different variations (e.g. steps that perform trajectories, may need to be performed facing several directions). These variations are also annotated and will be used in many different ways during motion synthesis.

C. Choreography

As discussed above, choreography involves both designating which steps each dancer will perform at any given time and the way the group will move along the stage creating formations and following trajectories. By exploring the classification described in Section II-B, we propose an interface in which choreographers can specify both of these elements.

We observed that group motions are usually specified with images of an upper view of the stage, where the dancers are represented as balls (often color coded) and lines or arrows indicate trajectories. Commonly, these images are staked together creating a storyboard that indicates the sequence of motions. In view of this, we created an interface for specifying group motion that has a window that represents the stage and where the choreographers can position the dancers and determine trajectories. Right bellow this window we place the musical reference with the authoring timeline. By clicking at any segment of the timeline, the choreographers can view the group motions specified for this segment.

In terms of representing individual motions, we have found that there is no notation, except full written descriptions or videos that would be efficient for all dance forms. For this reason, many choreographers have chosen to film the performance of the dance steps in order to be able to reproduce them later on. Based on this idea, we have suggested presenting the choreographer a list of the names of the captured dance steps. By clicking on any one of these names, the choreographer can preview an animation of the motion (see Figure 3) or see information such as size, available variations, etc.

Hence, our interface has two modes: group motion specification and dance movement specifications (see Figures 3 and 4). In both of these modes, specifications are made using the

authoring timeline shown in Figure 2. At the dance motion mode, 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 the steps. At the group motion mode, choreographers also have to fill in these boxes, but with elements that correspond to evolutions. Notice that evolutions often depend on formations and, therefore, instead of a list of steps, the interface's mode for group motion specification presents a list of formations, which choreographers can create by determining shapes, densities and patterns.

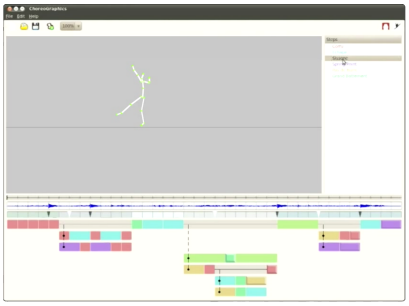


Fig. 3. Interface for dance steps mode

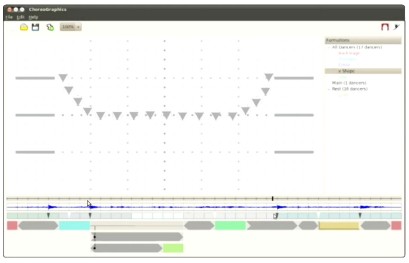


Fig. 4. Interface for group motions mode

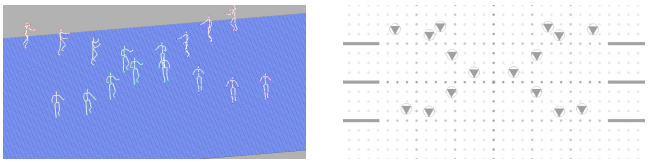
We explore the categorical aspects of group motions by allowing the dancers to be grouped and regrouped in different ways throughout the dance. We will also discuss these techniques in greater detail in Section V.

It is important to point out that choreographers should be able to specify dance steps to groups, subgroups or individual dancers. Hence, we use the same hierarchical structure to allow them to divide and subdivide the group on the dance movement specification mode. Notice, however, that these groupings 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).

We also allow users to visualize the resulting 3D animation in a separate window and to preview the group motion by playing an animation of the resulting 2D movement of the balls (see Figure 5).

IV. Dance Motion Synthesis

In this work, we chose to explore the motion graph technique [4] for creating new motions from a collection of MoCap data. 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.



(a) 3D motion (b) 2D motion

Fig. 5. Motion previews

For this reason, we propose, an extension of a motion graph, which is *structured* and *measure-synchronous*. As described in Section III-B, we organize the captured data as a group of clips that consist of sequences of measures. Hence, we create a *measure-synchronous* motion graph by, instead of creating nodes at any frame, only allowing nodes to be placed 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 clip with the name of the dance step and its variations. With this structure, we can represent the motion graph as a set of clusters, where each cluster represents a set of instances of the same step. The variations between the different instances may be: stylistic, structural 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.

Notice that multiple examples of the same step allows us to create more complex movements, as for example making the dancers follow different trajectories on stage while dancing. Also, the more instances of the same step we have, the greater the connection between the graph clusters.

It is noteworthy that the effectiveness of our motion synthesis method is highly dependent on the captured data. Instead of trying to get around this problem, by allowing some kind of 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.

A. Motion Editing Tools

One of the advantages of the structure that we developed is that it allows us to use very simple motion editing tools based on signal processing techniques and get very realistic results. In this work, we have explored such methods for the following motion modifications:

1) Interpolations and Combination: The greatest challenge with interpolating and combining upper and lower body motions is that the results will not be seamless unless there is a reasonable alignment between the two motion segments and that may require nontrivial warpings. In this work, we argue that since we have clips of the same dance step which have equal durations, we guarantee that they are trivially aligned by construction.

2) Rotations: Although dancers usually face the public during the show, it is often necessary to allow them to choose different orientations. In this work, we sugested a method for making the characters 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 was to analyze the foot contact, select the segments in which there is only one foot touching the floor and apply a rotation and translation transform to the motion sequence that changes the orientation of the dancer while maintaining the foot contact.

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 on the data to make the group dance look natural. In this work, we have explored time warping mechanism for desynchronizing the data and amplitude variation methods for varying the upper body motion. We chose to apply the latter tools only to the upper body part because they are less susceptible to undesirable artifacts, since there are no floor contact restrictions.

B. Combining Motion Segments

In this section, we will describe the mechanisms for searches in the motion graph.

1) Sequencing Dance Steps: To discuss the sequencing of dance steps, we will interpret the data as clusters of dance components, in which each cluster corresponds to a specific dance step and each component is a structural variation this dance step. The only variations that are relevant to this discussion are the structural changes. Notice that stylistic variations can be described as new clusters and that locomotion variations can be interpreted as iterations of a single dance component. Since each dance component is associated to a different structural variation, we can determine connections between them and represent them as a graph, as shown in Figure 6. Notice that this is a simplification of the actual motion graph, since each component may correspond to several dance clips.

The choreographer determines the sequence of dance steps, $d(n)$, for $0 < n \leq N$, where N is the number of dance steps. Our goal is to create a path in the graph, $p(n)$, also of size N, that matches this description (i.e., the order of the visited clusters should correspond exactly to the specified order of dance steps). Our approach is to use an A* with a tree search algorithm, a widely known method for best choice search which is optimal when used with an admissible heuristic [5].

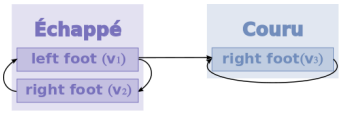


Fig. 6. Example of connections between the dance components.

We, therefore, specify the cost of a path as the total number of nodes n, such that $p(n) \neq d(n)$. Since we are using a tree search, at any given iteration, n is equal to the depth in the tree. Hence, given that n is known at any point in the algorithm we can specify the heuristic $h(x)$ (which is an estimate of the cost from node x to the objective) as $h(x) = 0$ if $p(n) = d(n)$ and $h(x) = 1$, otherwise.

Notice that this heuristic is admissible, because $h(x)$ is always less or equal to the actual cost of any path that contains that node at that stage. Of course, since the cost is the total number of unmatched steps, when we have at least one unmatched step we can guarantee that the total cost is greater or equal to one. Hence, we can guarantee that this algorithm is optimal (i.e., it will find the optimal solution) and is optimally efficient.

2) Following Trajectories: One of the essential aspects of this work is to synthesize group motions, i.e., to synthesize the way the dancers move on stage. Notice that the 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. Hence 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 information, and knowing the dance component each measure corresponds to from the above discussion, we are able to synthesize each motion measure. We start by enlarging the database significantly by interpolating clips that correspond to different locomotion variation, using the editing tools discussed in the previous section. 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.

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. Finally, we also use a feedback system to inform the dancers if new locomotion variations are needed during motion synthesis.

V. Group Motion Synthesis

When viewed together, a group of dancers moving on stage creates the impression of a dynamic, complex object.

Therefore, an important aspect of dance motion synthesis is determining how the dancers move as a group, creating formations, following trajectories, interacting without colliding, etc.

In this work, we have proposed both declarative methods, that take into account the standard specification methods used by choreographers, and procedural methods, which explore previous work on group control and simulation, to create tools for group motion design.

A. Declarative Methods

Declarative methods make extensive use of the characterization described in section II-B.

1) Formations: We specify formations by determining shape, density, and pattern. Shape refers to the geometry that the group of dancers assume 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 the dancers are distributed on the designed shapes. We also allow the choreographer to specify symmetries by drawing a symmetry line and reflecting over it the elements that are drawn on either side.

2) Evolutions: Evolutions are created by determining conditions and rules. We consider three types of group motions: the ones 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 of positions, synthesizing a trajectory, and determining the orientations. In this work, we use a bipartite matching algorithm [6] for matching the initial and final positions. We create trajectories by connecting the two positions with a straight line and then correcting the paths using a collision avoidance method. Finally, we specify orientations by guaranteeing that the characters face the path during the locomotion.

Movements based on initial conditions and evolution rules allow choreographers to plan very 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.

3) Segmentation: To allow choreographers to easily specify these groupings we developed a hierarchical structure for arranging and rearranging dancers into subgroups that are specified either based on individual identities or positions on stage.

B. Procedural Methods

In this work we explored behavioral animation methods [7] for specifying evolution rules. These methods regard each dancer as an autonomous agent that travels along the 2D manifold represented by the stage according to combined steering forces. Hence, new behaviors result from creating several types of forces and by combining them in different ways in order to allow agents to reach higher level goals.

There are, of course, innumerable evolution rules that could be designed based on these behaviors. For the purpose of illustrating the applications of this method, we have implemented the following group motions: following attraction/repulsion forces, spreading out on the stage, and crossing over.

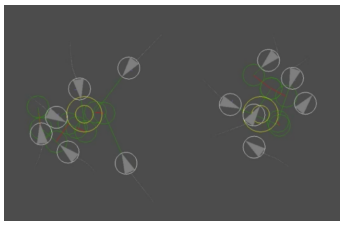


Fig. 7. Example of attractors.

Therefore, the choreographer positions structural elements on stage (e.g., an attractor) and determines the number of measures they should be active. Figure 7 illustrates the motion that results from two attraction forces.

VI. Conclusion

In this work, we suggested an authoring and collaboration platform for dance shows, discussed the relevance of this research to dance design; studied the technical aspects related to both dance and choreography specification and synthesis, developed tools that were sufficient to demonstrate the proposed concepts, and illustrated the applications of our method with experiments performed with dancers and musicians. The videos that illustrate these experiments are available at www.impa.br/∼aschulz/ChoreoGraphics/videos.

Future works should take proposed ideas to the next level, applying then to live scenarios in which conception and execution are done simultaneously, in a framework of instant feedback that fully integrates the collaborations of dancers, choreographers and musicians. We believe that this environment is much more than a tool to facilitate creation, but in fact, suggests new forms of artistic expressions, bridging the gap between performers and spectators. Expansions of the developed tools, additional experimentations, and contributions from artists can made these ideas evolve in directions that we can only start to imagine.

References

[1] A. Bruderlin and L. Williams, "Motion signal processing," in *SIGGRAPH '95*. New York, NY, USA: ACM, 1995, pp. 97–104.

[2] M. Brand and A. Hertzmann, "Style machines," in *SIGGRAPH '00*. New York, NY, USA: ACM, 2000, pp. 183–192.

[3] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 559–568, 2004.

[4] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *SIGGRAPH '02*. New York, NY, USA: ACM, 2002, pp. 473–482.

[5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2003.

[6] R. Diestel, *Graph Theory*. Fourth electronic edition, 2010.

[7] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," in *Computer Graphics*, 1987, pp. 25–34.