Sketch-Based Dance Choreography

Elahe R. Moghaddam
Department of Computational Media Design
University of Calgary
Calgary, Alberta, Canada
erajabza@ucalgary.ca

Javad Sadeghi
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
jsadeghi@ucalgary.ca

Faramarz Samavati
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
samavati@ucalgary.ca

***Abstract*—Sketching and doodling are two techniques commonly used by choreographers to design a dance sequence. These sketches usually represent the trajectory of the dancer in the scene. A set of annotations can be used to differentiate the various dance movements. In order to have more control over the choreographed dance, a 3D animation is preferable. This paper presents a novel sketch-based approach to assist dance choreographers authoring dance motions in a 3D environment. The proposed approach allows a choreographer to story board a dance using stick figure sketches of a dancer. Inspired by traditional choreography, a set of simple annotations is introduced for ballet. These annotations help to retrieve and blend ballet 'mini-motions' in order to create a synthesized dance. To build the mini-motions, we have analyzed and processed several ballet movements available in a MoCap database.**

***Keywords*-Dance; Choreography; Ballet; Sketch Recognition; Animation Synthesis;**

I. Introduction

Dance is an art form that involves rhythmic movements of the body. These movements are usually synchronized to a musical score and sometimes used to express an idea or story. Dance is used in many cultures as a form of emotional expression, social interaction, or exercise [1]. Many dances were developed as a feature of court festivities during the Middle Ages. Ballet, which originated from Italy and France between 15^{th} and 16^{th} centuries was one such dance [2]. With the popularity of ballet in France, special dance instructors or "Ballet Masters" began to appear [3]. They developed choreographic techniques to design dance patterns to be used throughout Europe. Dance choreography (also known as dance composition) is the act of composing dance patterns, usually intended to be performed in a concert [4]. Historically, choreographers have used drawings with a set of annotations to compose different dance patterns. The initial patterns are then observed and rehearsed several times to find any problems and correct them. Annotations play an essential part in this process, as they help document and preserve this traditional art form.

Over the decades, more than twenty annotation systems have evolved for ballet choreography, which can be divided into five categories: word abbreviations, track drawings, stick figure (visual) systems, music note systems and abstract symbol systems [5]. The classic word based annotations [6] and track drawing systems [7] are simple to use but not rich enough for the full control of 3D dance movements. On the other hand, rich but complicated 2D annotations (e.g. [8], [9]) are difficult to understand for novice dancers (see section III). It seems that stick figure systems [10], [11] provide a good balance between richness and simplicity.

Living in today's digital world that has computerized many traditional processes, one might ask: is it possible to use simple 2D annotations to unambiguously identify 3D dance movements? With this, the traditional approach of ballet choreography can be accompanied by a dancing avatar. This helps choreographers view the resulting dance from an arbitrary angle that reduces the ambiguities related to the dance patterns.

Recently, there has been increasing interest towards sketch-based interfaces for creating and controlling animation. These interfaces have caused a shift in the field of 3D animation interfaces by providing a more natural means of interaction. While controlling the full range of human motions with a small set of simple annotations is a challenging problem, ballet consists of limited and constrained motions. Therefore, in this paper we propose a sketch-based approach to create ballet dance animations in a 3D environment.

Our interface is more closely related to the traditional media (pen and paper) used for choreography than the standard mouse-and-keyboard interface. This makes it possible to use our approach on tactile devices and interact with choreography similar to pen and paper. As illustrated in Figure 1, doodling a rough sketch in our prototype can trigger a ballet *mini-motion*. These mini-motions cover the basic movements of ballet, hence are strongly constrained. We introduce a simple set of stick figure annotations for representing these mini-motions. Creating realistic human motions over the full spectrum of natural movements is challenging because human motion features many subtle details that should be considered in animation. Using Motion Capture (MoCap) techniques [12] will address this issue and provide 3D animation clips with all the characteristics of a live performance.

To construct mini-motions, we segment the ballet motion clips from a MoCap database according to the poses defined

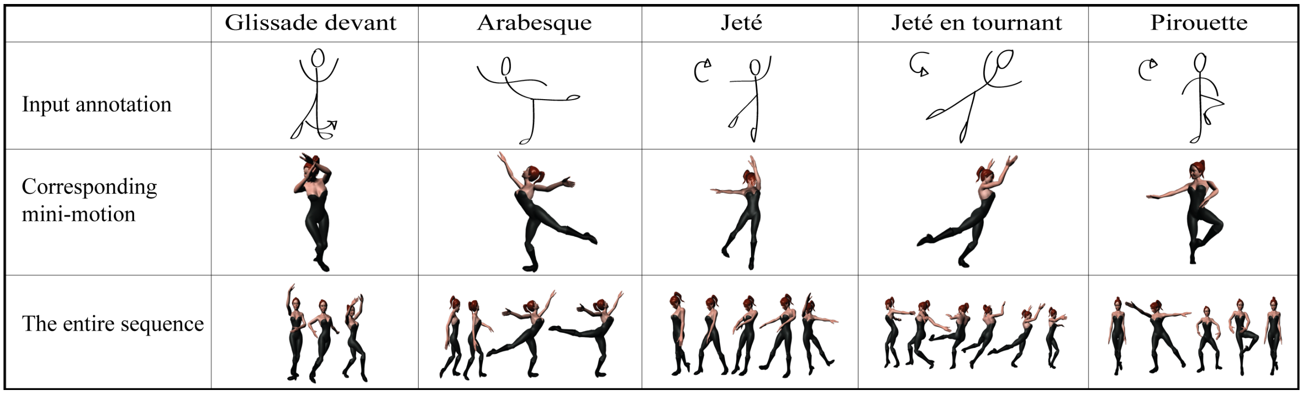

Figure 1. The breakdown of our approach for creating a dance sequence. Each input sketch is associated with a mini-motion. By blending these mini-motions the entire sequence is synthesized.

in a standard stick figure annotation system. We then use a state of the art sketch recognition algorithm to associate the input sketch to the corresponding mini-motion. The retrieved mini-motions are added to a list that can be re-ordered for the best combination. We allow choreographers to sketch the trace of the dancer in the scene. The mini-motions are then smoothly blended together along this path to synthesize a complete dance.

The contribution of this paper is three-fold:

- The concept of sketch-based dance choreography: a rapid-prototyping approach for ballet composition.
- Using a dynamic motion blending mechanism to smoothly blend the captured mini-motions while following a trajectory.
- Analyzing the dance notations used in choreography in order to define a set of simple annotations.

The rest of the paper is organized as follows. In section II, we review the related work to sketch-based dance choreography. In section III, different dance annotations are analyzed. In section IV, we outline our system's framework. The results of our approach are discussed in section V. Finally, the conclusions and future work are presented in section VI.

II. Related work

In dance choreography, different annotation systems have evolved to illustrate three dimensional movements on the two dimensional surface of a piece of paper. However, as discussed in section I, only the stick figure systems [10], [11] are easy to read and have less ambiguity. Also, compared to the other annotation systems, they have fewer symbols. While a good annotation system helps one to document the dance in a more readable format, it is not enough to document all of the subtle movements of a performer in 3D space. Watching a performance is the typical means of learning a dance, nevertheless Guest [13] argues that filming a dance is not sufficient to communicate what is envisioned by the choreographer. Video-assisted systems such as Dance Designer [14] enhance the learning process of dance but only support limited viewing angles. This is problematic as, during the recording of a dance, some movements may be obscured by the dancer from certain angles. James *et al.* [15] introduce a video-based choreography system that uses a visual stick figure interface to retrieve archival dance footages for synthesizing new videos. Their approach is useful when all videos are taken from a single viewpoint which is not enough to capture all of the spatial information. DanceForms [16] tackles the lack of spatial information in 2D choreography by providing a 3D layer on top of an abstract notation [8]. While it helps in avoiding some of the ambiguities during dance composition and editing, the way it visualizes the dance steps is not very natural. This results mainly from the limitations of DanceForms' dynamic animation generation. Connecting different motion clips from a MoCap database provides sufficient flexibility to create new motions on the fly. Web3D Dance Composer [17] uses this technique to animate realistic virtual ballet performances on the Web. However, it does not provide an easy to use interface for choreographers that resembles the way they usually author and edit the dance patterns on a piece of paper.

In recent years, sketch-based interfaces for doodling and controlling animation have gained popularity. Lee *et al.* [18] present the idea of real-time control of a three-dimensional character's motion in a 3D environment using a sketched path. Davis et al. [19] introduce a new sketching interface for rapidly creating animations for an articulated 3D figure from 2D sketches of the character in the desired key frame poses. Thorne *et al.* [20] use a sketch-based interface to sketch a character's motion. They parse user inputs and map them to a parametrized set of output motions that reflect the location and timing of the input sketches. Oshita [21] presents a pen-based intuitive interface to control human motion interactively. He employs the pen as a metaphor of sketch,

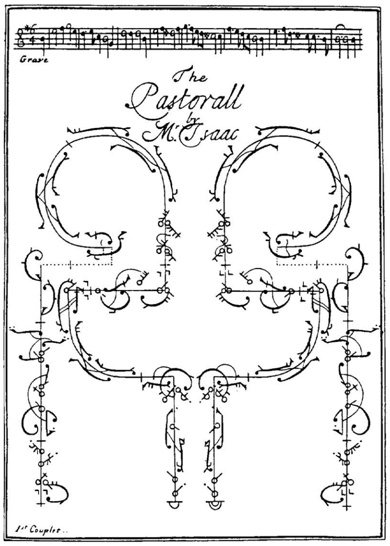

Figure 2. Feuillet's annotation system based on track drawing. Image from Britannica.

and directly maps pen movements to the figure motions. Figueroa *et al.* [22] introduce a sketch-based interface for creating an animation with sound from a drawing on paper. They use a combination of object symbols, actions and arrows for animation generation. MotionMaster [23] provides a sketch-based approach for choreographing Kungfu motions at the early stage of animation creation. It approximates realistic character motions that can be iteratively optimized and refined by animators to be used later in production. These techniques are useful for generating expressive motion in real-time; however, they support a limited subset of gesture vocabulary in 3D such as direction, orientation and speed, which is not enough to create detailed dance motions.

III. Annotation analysis

The symbolic representation of human movement is called dance annotation [13]. The process of dance annotation requires reducing the four-dimensional movement of a dance (time being the fourth dimension) to a two-dimensional surface of paper [24]. Therefore, in order to create a sketch-based system for choreography, we need a simple and comprehensive annotation system. Our preference is to use annotations that can be drawn as a single line. Also, it is vital that there exists no ambiguity when a 3D motion is created out of these annotations. There are more than twenty annotation systems for ballet choreography [5]. Here, we analyze some of the most relevant systems to our work that can visually represent dance movements.

The first known attempt to gather a full set of dance annotations was Feuillet's book: "Choreography; or, the Art of Describing the Dance" [7]. His notation was based on adding specific symbols to the trace of the dancer on the floor (see Figure 2). Our sketch-based approach is partly inspired by this system, because it provides a visual clue as to the location of the dancer in the scene. After Feuillet, many other notations were developed by choreographers. Arthur Saint-Léon in France introduced a choreography system based on stick figures. Friedrich Albert Zorn, a successful German dancer, choreographer, and dance theorist, extended this notation in Ukraine. He published his new annotation system in a book entitled: "Grammar of the Art of Dancing" in 1905 [10]. Figure 3 represents a sample choreography using Zorn's system. The simplicity of Zorn's annotations motivated us to use stick figure-based annotations in our approach.

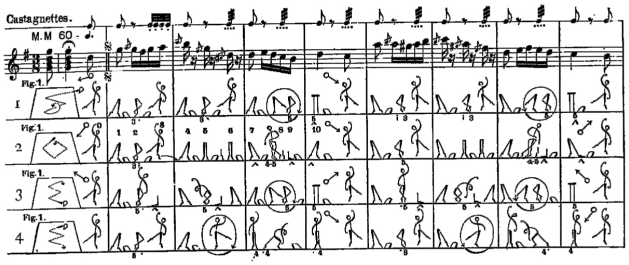

Figure 3. Zorn's annotation system. Extracts from the score of Fanny Elssler's *Cachucha*. Image from Wikipedia.

Another important ballet notation, that mostly focuses on the adaptation of motions to music notes, was developed by Vladimir Stepanov [25] in his book "Alphabet of Movements of the Human Body" (see Figure 13). Mapping movement annotations to music notes makes the input mechanism for sketch-based dance creation more complex, although the problem of mixing these two aspects remains an interesting future work. In the 1920s, Rudolf von Laban [8] introduced labanotation, which is a system for analyzing and recording human movements. His notation uses three staves that run vertically from bottom to top (see Figure 13). These abstract block symbols are used in some digital choreography techniques, e.g. DanceForms [16]. In 1940, Rudolph Benesh [11] developed Benesh movement notation, which is still used in many companies including the Royal English Ballet. Benesh's system uses five horizontal staves, similar to the staves of a musical score (see Figure 13). These lines are read from left to right, where each line refers to one part of the body. More recently, Noa Eshkol and Abraham Wachmann [9] introduced a new movement notation called Eshkol-Wachman notation. They use numbers and a small set of symbols to record each possible movement (see Figure 13).

In short, based on our observations and analysis of the pros and cons of each annotation system, it seems that annotations that use the trace of the foot or body of the performer on the floor e.g. [7] aid in understanding the dance better. However, they project the 3D movements onto a 2D plane, which is not enough to represent all spatial information.

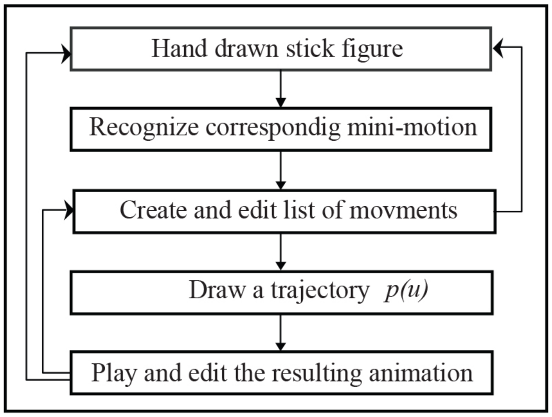
Hand drawn stick figure
Recognize correspondig mini-motion
Create and edit list of movments
Draw a trajectory $p(u)$
Play and edit the resulting animation

Figure 4. Our pipeline for sketch-based dance choreography.

On the other hand, the music note annotations e.g. [25] are helpful for another important aspect of dance. However, they can add the complexity of the input annotations. Also, regardless of the richness of the abstract systems e.g. [8], [9] they are difficult to be used as a simple doodling/sketching interface in compare with the stick figure systems [10], [11] that are easy to read and have less ambiguity.

As part of our analysis, we engaged in some informal discussions with a few ballet masters, choreographers and university instructors. These field studies highlighted their tendency towards using simpler notations, preferring stick figure notations over abstract symbols because it was more aligned with their artistic direction in dance. This motivated us to use a simplified version of Zorn's annotation system [10]. We do not require the full range of Zorn's detailed annotations, as only one annotation is needed per mini-motion.

IV. Methodology

In this section, we provide a detailed description of our sketch-based approach to dance choreography. Figure 4 illustrates our proposed steps to synthesize a dance from a set of mini-motions. First, we introduce the new stick figure annotations developed for this project. Then, we discuss how the corresponding mini-motions were acquired and prepared. Next, the sketch recognition approach that assigns annotations to mini-motions is explained. Finally, we describe how these recognized mini-motions are synthesized into a dance animation.

A. *Stick figure annotation*

As described in section III, we have elected to use a simplified version of Zorn's annotation system [10]. Hence, our annotations can be simple and also closer to the common shape of the stick figures. This is due to the fact that we only need them in order to retrieve a mini-motion, i.e. a sequence of keyframes. Figures 5(a), 6(a), and 7(a) show Zorn's notations for the Pirouette, Jeté and Rond de jambe movements respectively. To create one of these ballet motions, choreographer must draw every single pose, such as bending the left leg, stretching it, transferring to the other leg etc. For instance, in Figure 6(a) a combination of 5 different annotations (plus a repeat sign) are used to describe poses and movements. Figures 5(b), 6(b), and 7(b) illustrate our stick figure annotations that describe the entire sequence of comprising keyframes using only a single representative pose. Instead of showing the details of every movement, this pose captures the main characteristics of the move (according to the vocabulary of ballet). This pose may be accompanied by one or two movement arrows. The curved arrow shows which way the character turns while the straight arrow is used to indicate the direction of movement for the corresponding part of the body. For example, Figure 6(b) shows our alternative to Figure 6(a) using one representative pose and one movement arrow. These new annotations help choreographers focus on the main features of the motions rather than annotating every detail of the performance.

B. *Using captured motions*

To create a 3D dance with all the details of a real performance, we gathered a set of MoCap data for ballet. Capturing ballet motions is not easy for MoCap studios perhaps because ballet shoes usually do not work on their floors. Therefore, we were able to use only a small subset of the motions in the Motion Capture Database of Carnegie Mellon University [26]. These motions are first cleaned, adjusted and skinned using a motion editing tool [27]. The quality of the motions is acceptable, however, these motions contain full ballet dance sequences and need to be segmented into mini-motions. Therefore, we use a dynamic animation generation tool [28] to segment these motions into smaller

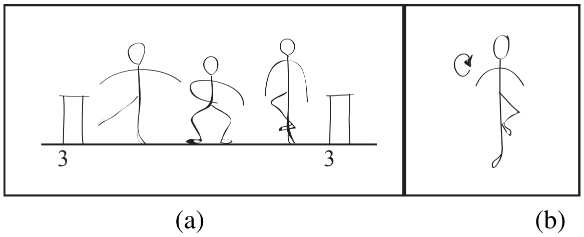

Figure 5. Pirouette (a) in Zorn Annotation and (b) in our annotation.

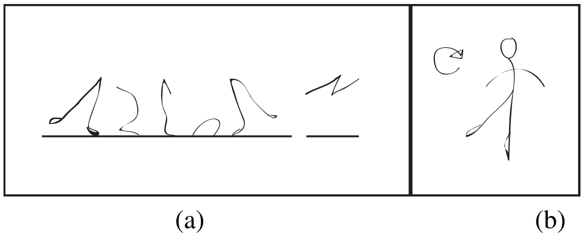

Figure 6. Jeté (a) in Zorn Annotation and (b) in our annotation.

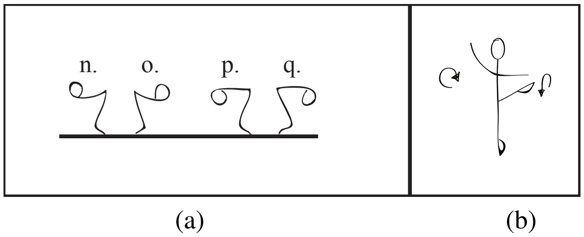

Figure 7. Rond de jambe (a) in Zorn Annotation and (b) in our annotation.

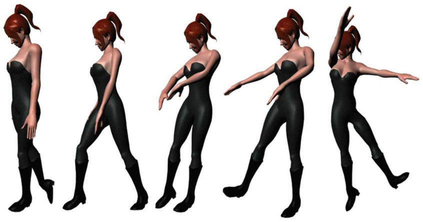

Figure 8. The Jeté mini-motion prepared according to the annotations provided in Figure 6(a).

components. Figure 8 shows five frames of the Jeté mini-motion carefully extracted from a larger clip. We visually matched the start and end frames of this mini-motion with the corresponding Zorn annotation shown in Figure 6(a). This process provides us with a basic set of ballet mini-motions $m_1, m_2, \ldots, m_k$ which are stored in our mini-motion repository.

Currently, in our prototype we have 30 mini-motions that covers only a fraction of the vocabulary of ballet. These mini-motions can be visually inspected and selected from a list. However, in practical scenarios we may have hundreds of mini-motions that makes the visual or text-based selection difficult. We address this difficulty using a stick figure sketch recognition approach.

C. Annotation assignment and sketch recognition

The next step, after preparing a set of mini-motions, is assigning their corresponding ballet annotations. As we use sketch-based input, we want to be able to retrieve mini-motions by sketching the corresponding annotations. We aim to support freeform strokes, without constraining the input to follow a specific order. Also, to complete the annotations, multiple strokes are usually needed. Finally, because of the branching structure of the stick figures, they do not fall into the category of simply recognizable shapes and need special treatment.

We experimented with a number of state-of-the-art gesture recognition algorithms to see which one better detects the drawn annotations with the discussed characteristics. We found that the $P Point-Cloud Recognizer [29] provides acceptable result for our approach, because it recognizes unordered point-clouds in a low-cost, accurate and fast way. It uses the nearest-neighbor approach to match the point cloud of the candidate gesture (C) to the point cloud of a template (T) in the training set with the smallest distance. To compute the matching distances, both C and T should be resampled to the same number of the points. To reduce the error rate, we store 10 different templates for each annotation in the training set. There templates follow no constraints on the number of the strokes, stroke ordering and stroke direction. Based on the recognized annotation, one of the mini-motions $m_1, m_2, \ldots, m_k$ is retrieved.

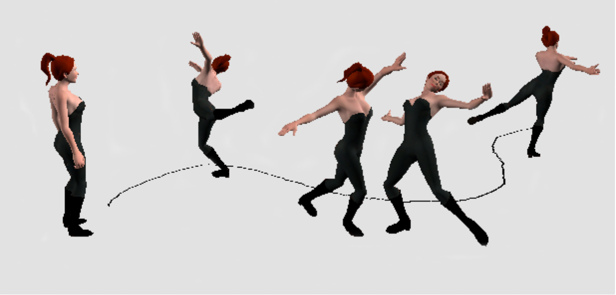

Figure 9. A sequence of dance following the sketched trajectory.

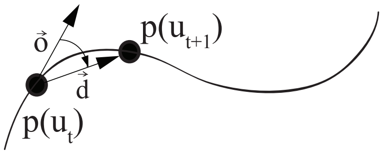

Figure 10. To align the direction of the dancer on the path's current direction, the dancer is tilted towards the approximate tangent.

D. Sketch-based dance choreography

In our method, we start with a sequence of sketched annotations (see Figure 1), then retrieve a sequence of mini-motions $m_{i_1}, \ldots, m_{i_l}$ where $i_1, \ldots, i_l \in \{1, \ldots, k\}$. As shown in Figure 4, after the mini-motions are retrieved, they can be re-ordered or re-arranged if necessary. To play the dance in the scene, the choreographer needs to specify the main trajectory of the dancer by drawing a path $p(u)$ on the floor of the 3D space, similar to the Feuillet system [7]. Mini-motions are then blended together and played along this path in the same order of their appearance as in the list of movements. This allows one to preview and evaluate the synthesized dance from different angles. Figure 9 shows a dance sequence created using this approach with 4 mini-motions, which follows the given path.

To create the trajectory $p(u)$, we trace the intersection points between the floor plane and the ray shot along the view direction from the current mouse position (see Figure 14). This path is then denoised and parameterized. As indicated in Figure 10, to align the motion along the sketched path, we first find the current direction $\overrightarrow{o}$ of the avatar using the quarternion rotation of its root joint at $p(u_t)$. Then, we tilt avatar's direction toward the tangent direction $\overrightarrow{d}$

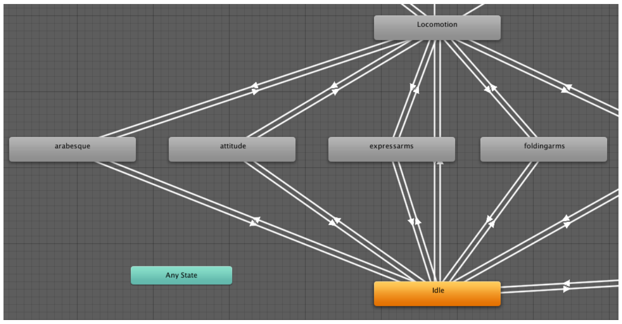

Figure 11. Unity3D's state machine is used to blend different mini-motions.

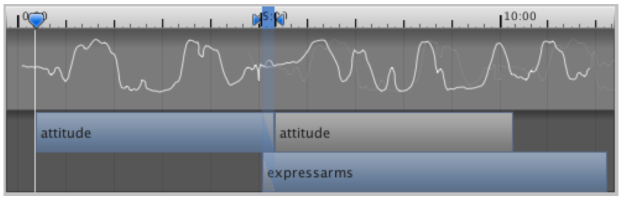

Figure 12. Blending two mini-motions using linear interpolation.

approximated by $p(u_{t+1})-p(u_t)$. The cross product $\overrightarrow{o}\times\overrightarrow{d}$ is used at each frame to calculate the dancer's tilt angle. By aligning the direction of the root join at $p(u_t)$ with the given path, the footsteps will follow the path automatically. To avoid visual discontinuity between footsteps of different mini-motions, they are adjusted to playback smoothly for each mini-motion.

To animate a smooth dance, the mini-motions need to be blended smoothly. For this purpose, we use the state machine mechanism of the Motion Graphs [30]. The mechanism used in the Unity3D [28] is based on similar technique. Figure 11 shows a portion of the states and transitions that we created using MecAnim Animation System of Unity3D. A sample transition is shown in Figure 12 that linearly blends the end frames of mini-motion 'attitude' to the starting frames of the succeeding mini-motion 'expressive arms'.

V. Results and discussion

Figure 14 shows a screenshot of the prototype created to support the approach discussed in section IV-D. In the left panel, the choreographer can focus on annotating dance steps by sketching their representative poses. If a mini-motion is successfully retrieved for this sketch, the corresponding icon can be added to the list of movements. Mini-motions in the list are represented with different color codes. The order of the mini-motions in this list can be easily changed. The trajectory of the dancer can be drawn interactively in the top view window. By hitting the play button in the playback bar, the dynamic blending mechanism is started. At the same time, the dancer is shown in the perspective view as well as the orthogonal view.

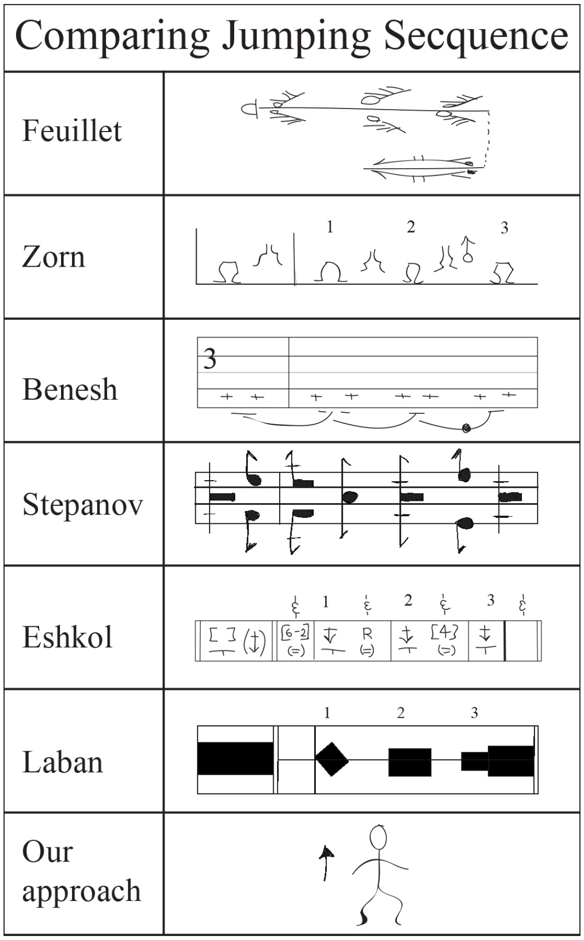

Figure 13. Comparing different notations for a jumping sequence.

This simple and natural interface allows people to observe the dance steps from different angles as visualized in Figures 15 and 16. Animated avatar in 3D scene provides a great opportunity to observe and learn ballet movements. Additionally, a sequence of screenshots taken from a good angle can be used for traditional media such as books. This can save a lot of time in drawing and publishing the training materials.

Figure 13 compares representation of a standard jump motion in our new annotation against other annotations that are commonly used for ballet choreography. Evidently, the ability of retrieving mini-motions helps us to greatly simplify the input annotation.

A. Implementation details

Our prototype is designed and implemented in the Unity3D game engine [28] with C# as a scripting language. As shown in Figure 17, we first import the mocap data of Carnegie Mellon University [26] into Motion Builder [27]. Inside Motion Builder, we re-target these mocap files to a rigged character by manually assigning the joints of the

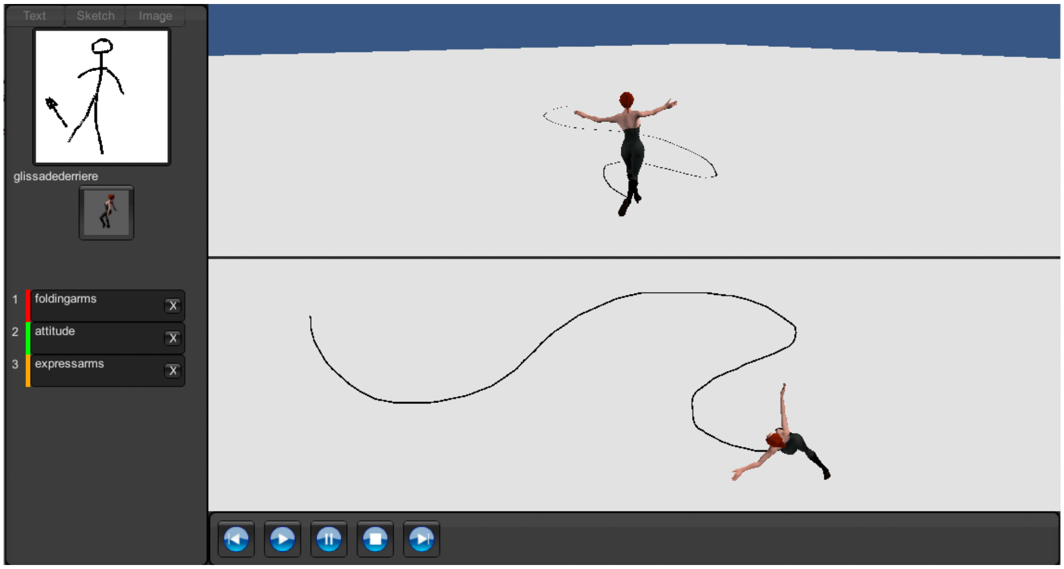

Figure 14. The user interface of our system.

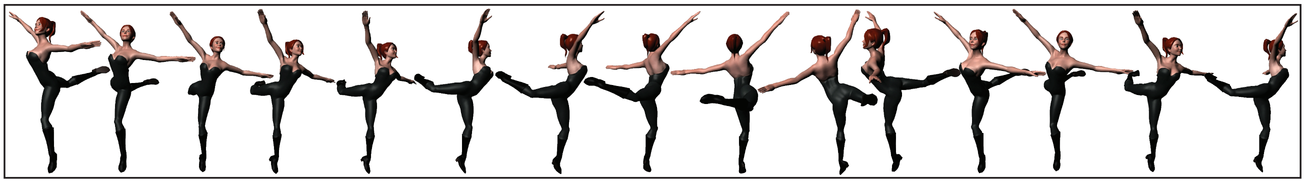

Figure 15. The Attitude pose shown from different angles.

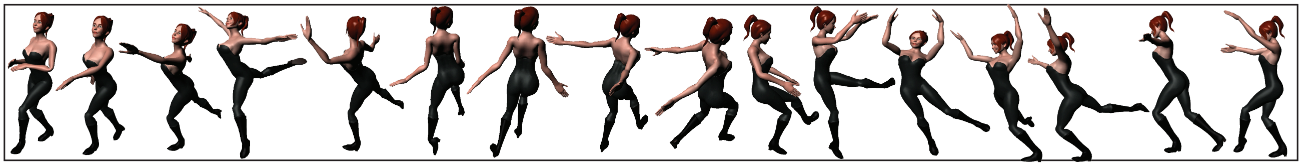

Figure 16. Combination of Arabesque, Jete en tourant and Bending back mini-motions shown from different angles.

dancers to a standard humanoid avatar in T- Pose. Next, we export the cleaned animations in FBX format to be imported into Unity 3D [28]. Then, we divide these motions into their mini-motions using Unity3D's MecAnim feature. Finally, we construct MecAnim's state machines and blend trees for smooth blending of these mini motions.

VI. Conclusion

We have proposed a sketch-based approach for creating 3D animations of ballet dance. Based on a brief study on traditional ballet annotations, we introduce a simple stick figure annotation for doodling the dance motions. To generate realistic motions, we use real MoCap data. In the future, we plan to conduct a formal study to evaluate the usability of our prototype for choreographers. Also, we believe that our approach can be customized for other applications such as: fashion shows, figure skating, gymnastics, and marching bands. Integrating and controlling music is also an important direction for future research.

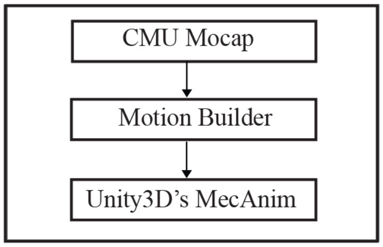

Figure 17. Our pre-processing pipeline.

Acknowledgement

Thanks go out to Troy Alderson and Ali Mahdavi-Amiri for their editorial comments. This research was supported in part by GRAND Network of Centre of Excellence of Canada.

References

[1] J. Adshead-Lansdale and J. Layson, *Dance History: An Introduction*. Routledge, 1994.

[2] A. Bland, *A history of ballet and dance in the western world*. Praeger, 1976.

[3] J. Anderson, *Ballet & modern dance: a concise history*. Princeton Book Co., 1992.

[4] L. Blom and L. Chaplin, *The Intimate Act of Choreography*, ser. Pitt paperback. University of Pittsburgh Press, 1982.

[5] A. Guest, *Choreo-graphics: A Comparison of Dance Notation Systems from the Fifteenth Century to the Present*. Gordon and Breach, 1998.

[6] T. Arbeau and L. Fonta, *Orchésographie*, ser. Bibliotheca musica Bononiensis. Forni, 1888.

[7] R. Feuillet, *Chorégraphie Ou L'art de Décrire la Dance*, ser. Monuments of music and music literature in facsimile. Ser. 2. Broude Brothers, 1700.

[8] R. Laban, *Principles of Dance and Movement Notation: With Basic Movement Graphs and Their Explanation*, ser. The Dance Horizons series. 20, 1970.

[9] N. Eshkol and A. Wachmann, *Movement Notation*. Weidenfeld and Nicolson, 1958.

[10] F. Zorn, A. Sheafe, and B. Coates, *Grammar of the Art of Dancing, Theoretical and Practical: Lessons in the Arts of Dancing and Dance Writing (choreography)*. Heintzemann Press, 1905.

[11] R. Benesh, *Benesh Movement Notation*. Institute of Choreology, 1955.

[12] M. S. Geroch, "Motion capture for the rest of us," *J. Comput. Sci. Coll.*, vol. 19, no. 3, pp. 157–164, Jan. 2004.

[13] A. Guest, *Dance notation: the process of recording movement on paper*. Dance Books, 1984.

[14] ChoreoPro, "Dance designer," https://www.choreopro.com, 2014.

[15] S. James, M. J. Fonseca, and J. Collomosse, "Reenact: Sketch based choreographic design from archival dance footage," in *Proceedings of International Conference on Multimedia Retrieval*, ser. ICMR '14. New York, NY, USA: ACM, 2014, pp. 313–320.

[16] T. Calvert, L. Wilke, R. Ryman, and I. Fox, "Applications of computers to dance," *IEEE Comput. Graph. Appl.*, vol. 25, no. 2, pp. 6–12, Mar. 2005.

[17] A. Soga, B. Umino, T. Yasuda, and S. Yokoi, "A system for choreographic simulation of ballet using a 3d motion archive on the web," in *Articulated Motion and Deformable Objects*, ser. Lecture Notes in Computer Science, F. Perales and B. Draper, Eds. Springer Berlin Heidelberg, 2004, vol. 3179, pp. 227–238.

[18] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 491–500, Jul. 2002.

[19] J. Davis, M. Agrawala, E. Chuang, Z. Popović, and D. Salesin, "A sketching interface for articulated figure animation," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 320–328.

[20] M. Thorne, D. Burke, and M. van de Panne, "Motion doodles: An interface for sketching character motion," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 424–431, Aug. 2004.

[21] M. Oshita, "Pen-to-mime: A pen-based interface for interactive control of a human figure," in *Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling*, ser. SBM'04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 43–52.

[22] P. Figueroa, J. Arcos, D. Rodriguez, J. Moreno, and F. Samavati, "A pen and paper interface for animation creation," in *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications (GRAPP/IVAPP)*. SciTePress, Jan 5-8 2014.

[23] Q. L. Li, W. D. Geng, T. Yu, X. J. Shen, N. Lau, and G. Yu, "Motionmaster: Authoring and choreographing kung-fu motions by sketch drawings," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 233–241.

[24] A. H. Guest, "Dance notation," *Perspecta*, vol. 26, pp. 203–214, 1990.

[25] V. Stepanov, *Alphabet of movements of the human body: a study in recording the movements of the human body by means of musical signs*. Dance Horizons, 1969.

[26] "Cmu graphics lab motion capture database," http://mocap.cs.cmu.edu, 2014.

[27] "Autodesk - motionbuilder," https://www.autodesk.com/MotionBuilder, 2014.

[28] "Unity - game engine," https://unity3d.com, 2014.

[29] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock, "Gestures as point clouds: A $p recognizer for user interface prototypes," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ser. ICMI '12. New York, NY, USA: ACM, 2012, pp. 273–280.

[30] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 473–482.