Puppeteering an AI - Interactive Control of a Machine-Learning based Artificial Dancer

Daniel Bisig
Center for Dance Research, Coventry University, Coventry, United Kingdoms
e-mail: ad5041@coventry.ac.uk
Institute for Computer Music and Sound Technology, Zurich University of the Arts, Zurich, Switzerland
e-mail: daniel.bisig@zhdk.ch

Ephraim Wegner
Offenburg University, Offenburg, Germany
e-mail: ephraim.wegner@hs-offenburg.de

Abstract

This paper describes the authors' first experiments in creating an artificial dancer whose movements are generated through a combination of algorithmic and interactive techniques with machine learning. This approach is inspired by the time honoured practice of puppeteering. In puppeteering, an articulated but inanimate object seemingly comes to live through the combined effects of a human controlling select limbs of a puppet while the rest of the puppet's body moves according to gravity and mechanics. In the approach described here, the puppet is a machine-learning-based artificial character that has been trained on motion capture recordings of a human dancer. A single limb of this character is controlled either manually or algorithmically while the machine-learning system takes over the role of physics in controlling the remainder of the character's body. But rather than imitating physics, the machine-learning system generates body movements that are reminiscent of the particular style and technique of the dancer who was originally recorded for acquiring training data. More specifically, the machine-learning system operates by searching for body movements that are not only similar to the training material but that it also considers compatible with the externally controlled limb. As a result, the character playing the role of a puppet is no longer passively responding to the puppeteer but makes movement decisions on its own. This form of puppeteering establishes a form of dialogue between puppeteer and puppet in which both improvise together, and in which the puppet exhibits some of the creative idiosyncrasies of the original human dancer.

1. Introduction

The two authors have been collaborating for many years in the production of dance performances, contributing from their side generative systems for dancers to interact with and thereby influence live generated synthetic music and imagery. One of the main fascinations of these activities lies in the development of generative processes that are sufficiently complex in their behaviour to achieve a wide range of aesthetic results but at the same time can be exposed to a dancer in a manner that makes them readily understandable and engaging to improvise with. This same focus also lies at the forefront of the research presented here, but instead of applying it to rule-based generative systems, it now concerns a machine learning system that had previously been developed by one of the authors. This system with the name *Granular Dance* [1] has originally been devised as a co-creative tool for choreographers to generate synthetic motion material that is reminiscent in form and style of the movements of a dancer that the system had been trained on. With this purpose in mind, *Granular Dance* was geared towards a creative workflow in which the potential for exploration, ideation, and discovery was a bigger concern than intuitive interaction and real-time feedback.

The current research started after *Granular Dance* had been re-implemented for real-time and interactive application. This reimplementation offers among others the possibility to generate motions for an artificial dancer with whom a human dancer interacts with in an improvised duet. While the improved computational performance of the reimplemented system makes this scenario theoretically feasible, its practicability is far from obvious. This is due to particularities of the machine-learning architecture that forms part of *Granular Dance* and the method formerly employed for searching for and generating synthetic motions. *Granular Dance* employs among others an autoencoder that operates on dance recordings in the form of motion capture data. This data is then encoded by the autoencoder into a compressed representation in the form of latent vectors which can be decoded to obtain a more or less faithful reconstruction of the original motion data. Most of the creative applications of autoencoders including the one originally chosen for *Granular Dance* dispose with the encoding process and instead directly search for latent vectors that might produce interesting motions once decoded. This search for latent vectors corresponds to a navigation in latent space. Unfortunately, it is difficult to gain an intuitive understanding for the organisation of this latent space and to become able to navigate it in a guided manner. This difficulty is linked to the intractable relationship between motion data and its encoding. Moreover, the spatial organisation of the latent space is rarely related to perceptual aspects of the data. This might not be much of an issue when there is ample time for exploration and experimentation at one's disposal. But for a performer who needs to quickly decide how to interact and what to expect from this interaction, navigating latent space is an overwhelmingly daunting task.

For this reason, the authors have started to explore other means of interacting with an autoencoder, particularly focusing on

forms of interactive control that takes place on the level of motion itself rather than its encoding. These explorations have led to the discovery, that autoencoders, when applied iteratively on their own decoded output, can produce interesting synthetic motions, in particular when some of the values of this output are removed from the autoencoder's influence and set by other algorithms or through interaction instead. The focus of this paper lies on the second method, the interactive control of elements in a motion capture sequence which is at the same time repeatedly encoded and decoded by the autoencoder. Since the autoencoder produces an output that specifies the motions for the entire body of an artificial dancer, interactively controlling only some joints of this body is reminiscent of puppeteering, in which a puppeteer only controls a subset of a puppet's joints while the other joints are taken care of by another process. In traditional puppeteering, this other process is physics, whereas in the system presented here its is autoencoding. While physics ensures that the puppet performs physically plausible motions, autoencoding ensures that the artificial dancer exhibits motions that are stylistically plausible to the extent that they resemble movements that have been recorded from a dancer.

The remainder of the paper is organised as follows. First is a background section that covers some of the methods and fields that are related to the authors' current research. This includes the use of motion capture to digitise the movements of dancers, techniques for creating synthetic motions with a specific focus on creative applications in dance, and the field of digital puppetry that relies on interactive forms of motion synthesis. Subsequently, a brief summary of the original *Granular Dance* system is provided, followed by a more extensive introduction of the changes implemented as part of the current research. Then the experiments conducted so far with the modified *Granular Dance* system are presented and discussed. Finally, these results are placed in a wider context with an outlook on future research directions.

2. Background

2.1 Motion Capture

Motion capture is a digital technique whereby the motions of one or several performers are captured by means of reflective or magnetic markers attached to their body parts [2]. A variety of sensing technologies can be used for this purpose ranging from high end optical systems consisting of multiple cameras surrounding the performers to more affordable but also less precise systems such as gyroscopes, accelerometers, or low end distance sensing cameras. The recordings produce digital data in the form of time-series of surface point positions which are then typically used to reconstruct an abstract three dimensional representation of the performers' body postures. This so called skeleton representation can then be further processed and visualised through a variety of means. In the field of dance, motion capture has become an important tool for academic and artistic purposes [3, 4]. Academic applications of motion capture recordings include their use as computation-friendly complement to dance notation [5], as resource for motion analysis with the goal of extracting high level qualitative

information from low level physical descriptors [6], or the documentation and preservation of dance traditions such as Cypriot folk dance [7]. Creative applications of motion capture include its use for gesture-based interaction with live media during performance (e.g. *Apparition*[1], *Stocos*[2], and *Dökk*[3]), for animating artificial dancers in virtual reality (e.g. *Dust*[4], *Dazzle*[5]), telematic performance (e.g. *Telematic Touch and Go*[6], *La Comedie Virtuelle*[7]), computer animation (e.g. *Asphyxia*[8], *Digital Body Project*[9]), or for creating interactive synthetic dancers on stage (e.g. *Emergence*[10]10, *AI am here*[11]).

2.2 Motion Synthesis

The term motion synthesis is employed here for any animation technique for computer generated characters that doesn't solely rely on the playback of previously recorded motion. The most prominent application domain of motion synthesis is computer animation and game design. Other fields that also make use of synthetically generated motions are robotics, interaction design, and dance.

In dance, the generation of synthetic motion serves a variety of purposes: as source for inspiration and ideation during the creative process, as mechanism for controlling artificial dancers that act as improvisation partners during rehearsal or performance, or to highlight and study choreographic principles that might be difficult to comprehend when relying on video recordings only.

Synthetic motions can be created by a variety of means, including physics simulation, artificial evolution, and machine-learning.

2.2.1 Physical Simulation

Simulations that model rigid body dynamics or mass-spring systems are attractive since they generate synthetic motions that appear physically valid. Many examples that employ physics simulation for creative purposes use mass-spring systems to create abstract artificial bodies that possess non-anthropomorphic morphologies. A choreographic support tool that is inspired by choreographic thinking in physical metaphors creates synthetic motions for abstract bodies that consist of a minimal set of mass-points and springs [8]. An interactive installation entitled *Becoming* employs a mass-spring simulation to create an artificial body that exhibits self motivated movements [9]. The artificial body is displayed as abstract graphical

[1] Apparation: https://www.escapeintolife.com/art-videos/klaus-obermaier-apparition/
[2] Stocos: https://www.stocos.com/en/page/stocos/
[3] Dökk: https://www.fuseworks.it/en/works/dokk/
[4] Dust: https://digitalartarchive.siggraph.org/artwork/maria-judova-andrej-boleslavsky-dust/
[5] Dazzle: https://springbackmagazine.com/read/bfi-lff-expanded-dance-virtual-reality-documentary/
[6] Telematic Touch and Go: https://journals.gold.ac.uk/index.php/lea/article/view/155/118
[7] La Comedie Virtuelle: https://www.gillesjobin.com/en/creation/virtual-comedie/
[8] Asphyxia: https://www.thisiscolossal.com/2015/03/asphyxia-a-striking-fusion-of-dance-and-motion-capture-technolog
[9] Digital Body Project: https://www.alexanderwhitley.com/digital-body
[10] Emergence: https://johnmccormick.info/category/emergence/
[11] AI am here: https://www.xorxor.hu/projects/theatre/aiam.html

animations meant to evoke kinaesthetic empathy and thereby foster movement ideation among dancers who rehearse in presence of the installation. A simulation-based system that has been employed for educational purposes and in a dance piece creates an interactive artificial dancer whose non-anthropomorphic morphology is modelled as mass-spring system [10]. The responsive behaviours of the artificial dancer convey expressive movement qualities which are obtained by designing specific mappings to simulation parameters for each of them. In the *Neural Narratives* series of dance pieces, a simulated mass-spring system is combined with a simple artificial neural network to create synthetic body limbs that acts as artificial extensions to a human dancer's natural body [11].

2.2.2 Artificial Evolution

Artificial evolution has been employed for the creation of synthetic motions that can be used by choreographers as inspirational resource. A system entitled *Scuddle* creates incomplete motion data that serves as catalysts for a choreographer's creativity [12]. Later on, the same research team has developed a system entitled *Cochoreo* that generates unique key-frames as seed material for a choreographer's creative process [13]. Both of these systems generate synthetic motion using a Genetic Algorithm whose fitness function incorporates among others Laban Motion Analysis categories [14,15]. The *Dancing Genome Project* employs an interactive genetic algorithm to modify sequences of basic motions which are then shown as live scores to both an artificial and a human dancer [16].

2.2.3 Machine Learning

With recent progress in machine learning, the use of data-driven methods for synthesising motion has gained in popularity. Data driven methods have proven effective in generating synthetic motions that are natural looking and expressive. These methods can capture and imitate the idiosyncratic movement styles of individual dancers when trained on corresponding motion capture recordings.

Several machine learning systems based on neural networks have been proposed as co-creative tools for choreographers. A system entitled *Chor-rnn* implements a recurrent neural network [17]. The authors of this system suggest a collaborative workflow in which choreographer and tool take turns in creating motion material. In two publications, different deep-learning architectures are compared with regard to their usefulness for choreographic purposes. Based on a subjective evaluation of mixed density networks, autoencoders, and Long-Short Term Memory (LSTM) networks, the authors conclude that LSTMs perform best on criteria such as posture prediction, temporal coherence, motion consistency, and aesthetics [18]. Another comparison between autoencoders and LSTMs places a stronger focus on the flexibility of creating motion variations [19]. This comparison ends up given more attention to autoencoders than LSTMs.

Machine-learning systems have also been employed to realise interactive artificial dancers. Such an approach has been thoroughly explored by Berman and James [20-22] and McCormick and colleagues [23-25]. Both teams experimented with a variety of machine learning techniques to obtain a system

capable of synthesising motions that are similar to those of a human dancer. Their systems had initially been employed in interactive rehearsal settings and where later on adapted for stage performances in which an artificial and human dancer interact in a duet. The *LuminAI* system employs a clustering mechanism to select movements for an artificial dancer that mirror with some deviations those of an interacting human dancer [26]. The *Viewpoints AI* system implements an interaction-based authoring approach for the creation of synthetic motions that combines ideas from case-based learning and imitative learning [27].

When working with autoencoders, navigating the latent space of encodings is a popular method for creating new movement material. This approach has been chosen both by researchers working with encodings of poses (e.g. [22,18,19]) and researchers working with encodings of pose sequences (e.g. [28-30]). One of the main drawbacks of navigation latent space pertains to the difficulty of obtaining an understanding for the typically obscure relationship between latent vectors and their decoding. Several researchers have tried to address this issue. One possible approach is to condition the autoencoder on higher level control parameters (e.g. [31]). Another approach is to extend an autoencoder with a control network that learns to disambiguate latent space (e.g. [32]).

2.2.4 Digital Puppetry

The term Digital Puppetry refers to any set of techniques for interactively controlling the behaviours of computer generated characters. The behaviours might not only involve the motion of the artificial character but also for instance facial expressions and the uttering of words. Digital Puppetry draws its inspirational background from traditional puppetry and combines it with ”live action, stop motion animation, game intelligence and other forms into an entirely new medium” [33]. An extensive overview over the field including a recapitulation of major developments and applications in particular within the animation industry is provided in the PhD thesis by Leite Orvalho [34].

In traditional puppeteering such as marionette theatre, a mechanical puppet is set into motion by means of a human performer (the puppeteer) controlling a subset of the joints of the puppet. The puppet’s remaining joints then move according to physical laws. This form of interaction that doesn’t involve enough degrees of freedom (DOF) to control all the joints of a puppet directly constitutes one of the main technical challenges in digital puppetry. There exist a number of methods to construct motion for a puppet that is underspecified through interaction. The most frequently employed method is to generate motion through inverse kinematics for those puppet joints that are not directly under the puppeteer’s control (e.g. [35]). An alternative is to simulate a puppet’s physical dynamics to fully articulate it. An interesting example of this approach has been implemented in a system that reconstructs human motion from ground reaction forces and hand movements only [36]. A third option is to (partially) dispose with direct and continuous control of puppet joints and instead select the puppet’s poses and actions based on context. A combination of physics simulation and behaviour selection has been proposed by Ishegaki

and colleagues for controlling the behaviour of a character in a virtual environment [37]. Here, the selection of behaviours is based on a combination of recognising a user's intent and simulating only those actions that are appropriate in the current situation within the virtual environment. Several researchers have explored the use of motion capture recordings in combination with interaction mechanisms for generating motions for digital characters that are derived from these recordings. One example uses machine learning to recognise a user's individualised hand poses and then select a character animation from a database of human locomotion recordings [38]. Another example uses a generative recurrent machine-learning model that has been trained on motion capture recordings for synthesising motions of a digital character while it interacts with a human player [39].

3. Implementation

The system employed in this research uses a generative machine-learning model for synthesising short pose sequences in combination with a blending and sequencing mechanism for concatenating the generated sequences. This system with the name *Granular Dance* has been developed by the first author and is described in some detail in a previous publication [1]. The main architecture and operation of *Granular Dance* remains unchanged and is briefly summarised in this article. What has been added in the meantime is a mechanism to create synthetic motions in which one or several joints can be interactively controlled while the remaining joints remain under the control of the machine-learning model. This combination of interactive control and machine learning offers the possibility to create an artificial dancer that partially mirrors the activities of a human performer but at the same time also exhibits autonomous motions that preserve some of the stylistic uniqueness of the original motion capture recordings.

3.1 Granular Dance

The machine-learning part of *Granular Dance* consists of an adversarial autoencoder that has been trained on motion capture recordings of a professional dancer who freely improvised to music. The model's architecture consists of three neural networks, an encoder, decoder, and discriminator (figure 1 left side). The encoder and decoder parts are autoregressive. The encoder takes a sequence of poses as input and generates a low dimensional representation of this sequence in the form of a latent vector as output. The decoder takes latent vector as input a and reconstructs a sequence of poses as output. The discriminator takes a latent vector as input and produces as output a binary value indicating its assessment of whether the vector follows a Gaussian distribution or not. This mechanism ensures that the space of latent vectors is free of gaps and that distances within it represent a measure of similarity. This in turn ensures that arbitrarily chosen latent vectors can be decoded into meaningful pose sequences.

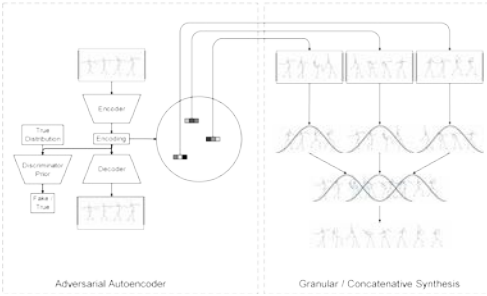

Figure 1: Granular Dance System Architecture. Shown on the left side is a schematic representation of the machine learning model depicting its three neural networks: an encoder, a decoder, and a discriminator. The trapezoid shapes of the encoder and decoder indicate the dimension reduction and expansion that is conducted by them. The circular region represents the latent space of pose sequence encodings. Shown on the right side is the pose sequence blending mechanism that mimics granular and concatenative synthesis techniques. Sequences are obtained by decoding sampled latent space vectors. Multiple decoded sequences are blended into longer sequences using overlapping windows which are depicted here as bell shaped curves.

The sequence blending mechanism serves the purpose of concatenating the short pose sequences generated by the autoencoder into longer sequences (figure 1 right side). The mechanism draws inspiration from methods in computer music that combine short sound fragments to generate longer sounds: Granular Synthesis [40] and Concatenative Synthesis [41,42]. Since poses are represented by joint angles in the form of unit quaternions, their blending is based on spherical linear interpolation (SLERP) [43]. Source sequences are blended one after the other with a target sequence with which they are aligned at successively increasing positions. The amount of blending between source and target is controlled by a window function (Hanning in this case).

3.2 Iterative Autoencoding and Interaction

The original *Granular Dance* system has been modified for real-time application which opens up the possibility of interactively controlling some or all of the joint orientations in a pose sequence. As part of this adaptation, the system has been ported from an implementation based on the *Tensorflow*[12] machine-learning framework to one using the *PyTorch*[13] framework. This change made it possible to integrate the trained neural networks into a software suitable for interactive real-time applications. This software was written in C++ using the *openFrameworks*[14] creative coding environment. Also for real-time purposes, a neural network has been chosen that operates on sequences that are 8 poses long (as opposed to 128 poses used in some of the experiments described in the original publication). By operating on such short sequences, the model can synthesise new motions frequently enough to be suitable for interactive applications.

The main novelty introduced in the new implementation involves the combination of an iterative autoencoding mechanism with interactive control of some of the joint orientations. In iterative autoencoding, the output of the decoder

[12] Tensorflow: https://www.tensorflow.org/
[13] PyTorch: https://pytorch.org/
[14] openFrameworks: https://openframeworks.cc/

is repeatedly feed back as input for the encoder. Each of these iterations starts with a pose sequence that is obtained from playback of the original motion capture recording. This pose sequence is then gradually transformed through the repeated encoding and decoding steps until it converges into a sequence that is considered by the autoencoder to be statistically most representative of the originally recorded material. Interactivity interferes with this convergence process by it setting the rotations of those joints that are interactively controlled and preventing them from being modified by the autoencoder. The autoencoder is forced to shift the convergence towards a pose sequence that respects the joint rotations specified through interaction and is statistically representative.

The number of iterations used for convergence via autoencoding and the number of iterations during which the joints are interactively controlled do not necessarily have to be the same. If they are the same, the joint rotations specified through interaction will remain removed from the influence of the autoencoder until the end of convergence and are fully visible in the final pose sequence. But if they are not, the interactively controlled joint rotations will only be kept removed from the influence of the autoencoder at the beginning of convergence and afterwards change as result of the autoencoding process. Accordingly, by varying these iteration numbers, the amount of influence interaction exerts on the autoencoding process can be modified. For purposes of brevity, the following two terms will be used from now on to refer to the two different types of iterations: *coding iterations* stands for the number of iterations of the autoencoder, *interaction iterations* stands for the number of iterations the orientations of the joints are set through interaction. It is worthwhile to mention that the method chosen here is related to two distinct operational principles of autoencoders. One principle is based on the de-noising capability of autoencoders. Since autoencoders represent high dimensional data as low dimensional encodings, they are forced to ignore instantiations that are statistically rare or exhibit unsystematic variance. Repeatedly applying an autoencoder leads to the removal of such instantiations with the number of iterations controlling the extent of this removal. The other principle is based on the conditioning of autoencoders. Autoencoders can be conditioned on the type of data they generate by adding one or several conditioning variables to the input of both the encoder and decoder. By fixing some of the joint rotations during autoencoding, these rotations become conditioning variables for the encoder. But in the method employed here, these rotations are only provided as conditioning variables to the encoder and not to the decoder. Therefore, the current method is not fully equivalent to autoencoder conditioning.

4. Results and Discussion

Prior to the experiments described below, the autoencoder has been trained on the same motion capture data and with the same training configuration as the original *Granular Dance* system. The recorded subject is a professional dancer who was freely improvising to music.

Several experiments have been conducted that differ from each other with respect to the excerpts chosen from

the motion capture recording, the method for providing interactive control, and the number of *coding iterations* and *interaction iterations*.

Five excerpts from the motion capture recordings have been selected. Each of these excerpts is 200 frames long which corresponds to a recording duration of 4 seconds. Excerpt 1: sitting pose with no body motion[15], Excerpt 2: slow torso bending with little arm and leg motion[16], Excerpt 3: arm and leg motion with little coordination[17], Excerpt 4: arm and leg motion with strong coordination[18], Excerpt 5: full body motion with partial coordination[19]. For illustrative purposes, all excerpts are depicted as filmstrips (figure 2).

As a first set of experiments, the number of *coding iterations* was varied between 1 and 10 with no interactive control of joints. These experiments served the purpose of evaluating how autoencoding itself converges the different excepts. The results of these experiment are available online as video[20] [21] [22] [23] [24].

[15] Excerpt 1 Mocap:
https://vimeo.com/641893597/801facc546
[16] Excerpt 2 Mocap:
https://vimeo.com/641894420/3397e70410
[17] Excerpt 3 Mocap:
https://vimeo.com/641894979/a2a7e7cc84
[18] Excerpt 4 Mocap:
https://vimeo.com/641895497/2fa62ef9f5
[19] Excerpt 5 Mocap:
https://vimeo.com/641896074/13f125a1e8
[20] Excerpt 1 Coding Iterations:
https://vimeo.com/641983602/ae61834f47
[21] Excerpt 2 Coding Iterations:
https://vimeo.com/641984449/9c7b6316de
[22] Excerpt 3 Coding Iterations:
https://vimeo.com/641985073
[23] Excerpt 4 Coding Iterations:
https://vimeo.com/641985646/7b02cd5a92

Some of the results are depicted as filmstrips (figure 3).

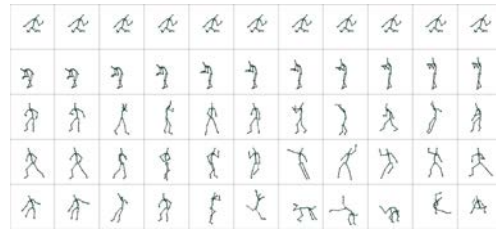

Figure 2: Excerpts of Motion Capture Recordings. From top to bottom, the filmstrips depict excerpts 1 to 5.

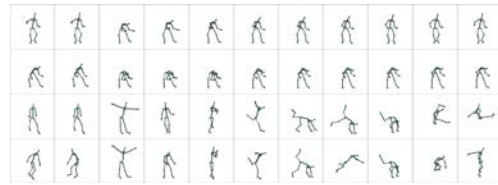

Figure 3: Variations in Coding Iterations. From top to bottom, the filmstrips depict: excerpt 2 coding iterations 3, excerpt 2 coding iterations 10, excerpt 5 coding iterations 3, excerpt 5 coding iterations 10.

The results of these *coding iteration* experiments can be summarised as follows. When conducting a single iteration, the autoencoder faithfully reconstructs the original motion with little qualitative difference. As the number of iterations increases, the reconstructed motion begins to deviate from the original motion. The only exception is excerpt 1 which consists of a mostly static pose that is properly reconstructed also at higher numbers of *coding iterations*. Some of the changes introduced through multiple *coding iterations* occur only for some excerpts while others are more common. For excerpt 2, an increase in

[24] Excerpt 5 Coding Iterations:
https://vimeo.com/641986453/727df69237

coding iterations results in a rotation of the skeleton that makes it front facing (e.g. rows 1 and 2 in figure 3). For all excerpts except number 1, an increase in *coding iterations* causes the originally smooth motion to become intermittent, with the skeleton pausing on a pose before jumping to the next pose. In most cases, the poses that are sustained occur in the original excerpt (e.g. row 2 in figure 3). In a few cases (excerpts 3 and 5), the skeleton also halts on poses that do not occur in the original excerpt (e.g. frame 3 in rows 3 and 4 in figure 3).

In a second set of experiments, a single joint (the left shoulder) was interactively controlled. The decision to interact with a single joint only served the purpose of simplifying the evaluation of the interplay between interaction and autoencoding. Also, interacting with a single joint constitutes the most extreme scenario for testing the suitability of the chosen approach for generating synthetic motions through underspecified interaction.

The first interaction experiment didn't involve any interaction at all. Rather, the orientation of the joint selected for interaction was either fully or partially removed from being modified by the autoencoder but otherwise left unchanged. This was done by increasing the number of *interaction iterations* to the same value or half of the value of the number of *coding iterations*, respectively. These experiments were conducted for each excerpt with the following variations: *coding iterations* 1 and *interaction iterations* 1, *coding iterations* 10 and *interaction iterations* 10, *coding iterations* 10 and *interaction iterations* 5. The results of these experiments are available online as videos [25] [26] [27] [28] [29]. In case of excerpt 2, the results are also depicted as filmstrips (figure 4).

Figure 4: Interactively Controlled Joint with Fixed Orientation. From top to bottom, the filmstrips depict: excerpt 2 coding iterations 1 interaction iterations 1, excerpt 2 coding iterations 10 interaction iterations 5, excerpt 2 coding iterations 10 interaction iterations 10.

Figure 5: Interactively Controlled Joint with Changing Orientation. From top to bottom, the filmstrips depict: excerpt 2 coding iterations 10 interaction iterations 5 cycle duration 1 sec, excerpt 2 coding iterations 10 interaction iterations 10 cycle duration 1 sec, excerpt 2 coding iterations 10 interaction iterations 5 cycle duration 5 sec, excerpt 2 coding iterations 10 interaction iterations 10 cycle duration 1 sec.

[25] Excerpt 1 Fixed Orientation: https://vimeo.com/642444172
[26] Excerpt 2 Fixed Orientation: https://vimeo.com/642444921/e0d935badf
[27] Excerpt 3 Fixed Orientation: https://vimeo.com/642445708/ba96a8f428
[28] Excerpt 4 Fixed Orientation: https://vimeo.com/642446118/9408fc7f06
[29] Excerpt 5 Fixed Orientation: https://vimeo.com/642446631/c6b5c6c3c9

The results can be summarised as follows. For all excerpts, the fixed orientation of the single joint is only clearly present in the generated motion when the number of *control* and *interaction iterations* are the same. If the number of *interaction iterations* is lower that the number of *control iterations*, the initially fixed orientation is quickly changed by the autoencoding process. The influence of the fixed joint orientation on the remaining joints is very small when the number of *coding* and *interaction iterations* is one. In this case, the original motion in each of the excerpts remains largely intact. As the number of iterations increases, the deviation from the original motion also increases, with excerpt 1 being the only exception. In case of excerpt 2, the presence of intermittent poses increases with most of these poses not originally present in the excerpt (rows 2 and 3 in figure 4). In case of excerpt 3 and 4, intermittent poses appear only briefly and are quickly switched between. In case of excerpt 5, several brief and one long intermittent pose are present. The brief poses appear during moments in the original excerpt that contain frequent arm motions. The long pose appears towards the end of the original excerpt when the arms barely move.

In the second interaction experiment, the orientation of a single joint was changed by rotating the joint at constant velocity and around a constant rotation axis over several full revolutions. Two velocities were chosen for the rotation, one completing a revolution in 1 second and the other in 5 seconds. These experiments were conducted for each of the excerpts with the same variations in the number of *coding* and *interaction iterations* as in the previous experiments. The results of these experiments are available online as videos[30] [31] [32] [33] [34] [35] [36] [37] [38] [39]. In case of excerpt 2, the results are also depicted as filmstrips (figure 5).

The results can be summarised as follows. For all excerpts, the rotating interaction joint is only clearly perceivable in the generated motion when the number of *control* and *interaction iterations* are the same. Also for all excerpts, the influence of the rotating interaction joint on the remaining skeleton joints is strongest when the rotation leads to a joint orientation that is markedly different from the one originally present in the excerpt. This effect is particularly pronounced when the orientation of the joint is unrealistic. For almost all excerpts, the velocity of the joint rotation affects the pacing of the entire skeleton motion. This is even the case for excerpt 1 in which the joint rotation adds motion to the formerly static skeleton pose. The effect of motion

[30] Excerpt 1 Changing Orientation 1 Second: https://vimeo.com/642447348/fe46ecd0de
[31] Excerpt 1 Changing Orientation 5 Seconds: https://vimeo.com/642447967/07f3d20889
[32] Excerpt 2 Changing Orientation 1 Second: https://vimeo.com/642448460/74c5a93b60
[33] Excerpt 2 Changing Orientation 5 Seconds: https://vimeo.com/642449116/d6def432fb
[34] Excerpt 3 Changing Orientation 1 Second: https://vimeo.com/642449562/3469ab1be0
[35] Excerpt 3 Changing Orientation 5 Seconds: https://vimeo.com/642450217/147b7af357
[36] Excerpt 4 Changing Orientation 1 Second: https://vimeo.com/642450850/624f5e5bb9
[37] Excerpt 4 Changing Orientation 5 Seconds: https://vimeo.com/642451216/46149451b3
[38] Excerpt 5 Changing Orientation 1 Second: https://vimeo.com/642451618/b77b2cbdb2
[39] Excerpt 5 Changing Orientation 5 Seconds: https://vimeo.com/642452046/cd29e08daf

pacing is weakest when the orientation of the rotating joint is similar to that of the original excerpt, and its is strongest if it is dissimilar. In case of excerpt 1, the amplitude of the induced motion is larger for slow joint rotations than for fast ones. Excerpt 5 is an exception in that the pacing and outline of the original motion is largely unaffected by the rotating joint.

For the last interaction experiment, the orientation of the interaction joint was controlled with a wearable sensor that was attached to the hand of one of the authors. The sensor is an inertial measurement unit (IMU) with an integrated component for deriving the sensor's absolute orientation[40]. This sensor was combined with a microcontroller with integrated Wii-module[41]. The sensor's absolute orientation in unit quaternion format was transmitted wirelessless at 50 Herz and directly mapped on the orientation of the interaction joint. This experiment could correspond to an improvisation setting on stage in which a human dancer performs a duet with an artificial dancer. The improvisation was conducted with two different iteration configurations. Configuration 1: *coding iterations* 2 *interaction iterations* 1. Configuration 2: *coding iterations* 10 and *interaction iterations* 9. These two configurations were chosen as examples in which interaction and autoencoding would weakly (configuration 1) or strongly (configuration 2) alter the original motion from the motion capture recording. In both cases, the number of iterations was chosen so that autoencoding affects during the last iteration all the skeleton joint orientations, including the interactive joint. This removed an otherwise obvious mirroring effect between the joint orientation of the human and that of the skeleton. The results of these two experiments are available online as video[42] [43].

5. Conclusion and Outlook

Based on the results obtained so far, it seems clear that iterative autoencoding in combination with interactive control of individual joints offers an interesting and flexible form for controlling an artificial dancer. This approach works acceptably well for cases in which the number of DOF of the control interface is dramatically lower than those of the artificial dancer and where the focus lies on the creation of synthetic motions that preserve some of the stylistic properties present in the original motion capture recordings. Furthermore, the approach offers an interesting combination of conventional motion playback, generative principles, and intuitive interaction. A similar level of intuitiveness might be difficult to achieve when working directly within the latent space of autoencoders.

But despite this intuitiveness, there is still a learning curve involved for gaining an understanding for the interplay between the properties of the original motion recording, the autoencoding mechanism, and the effect of interfering with both of them by controlling certain joint orientations through interaction. The

[40] Bosch BNO055: https://www.bosch-sensortec.com/products/smart-sensors/bno055/

[41] Arduino MKR1000: https://www.arduino.cc/en/Guide/MKR1000

[42] Improvisation Coding Iterations 2 Interaction Iterations 1: https://vimeo.com/642488225/1e32e7de68

[43] Improvisation Coding Iterations 10 Interaction Iterations 9: https://vimeo.com/642488605/e033dcaf0c

experiments conducted so far shed some light on this interplay.

An important insight gained concerns the balance between original motion material, interaction, and autoencoding. This balance is quickly shifted towards a dominance of the effect of autoencoding as soon as the number of *coding iterations* is chosen higher than 1. If this is the case, autoencoding brings to the forefront the most frequently occurring material in the original recording while removing everything else. It is important to keep this effect in mind when recording new movements and trying to ensure that some of these movements will reappear in the behaviour of the artificial dancer. Another important insight concerns the difference between the original orientation of a skeleton joint and the interactively generated orientation. The larger this difference is, the stronger does autoencoding alter the originally recorded motions, up to the point where the autoencoder generates synthetic motions which are entirely dissimilar to the currently played section of a motion recording. By paying attention to the motions currently executed by the artificial dancer, a human performer can exploit this principle and either make the artificial dancer reproduce motifs from the original motion recording or generate novel motions. These novel motions can either emerge as result of the autoencoder chaining together originally unrelated motions or by decoding vectors from regions of latent space that have been sparsely populated during training. The latter effect can be enforced by making the interactively controlled joints assume unrealistic orientations. A last important insight concerns the relationship between the velocity of the recorded motions and that of the interactively controlled joints. There are two aspects to this. First, if the recorded motions exhibits high velocity, then it is difficult to maintain a specific level of similarity between the interactively controlled orientation of a joint and its original orientation. As consequence, the motions of the artificial dancer become unpredictable. Second, if only a few joints exhibit high velocity in the original recording, then slowing their motion down causes the remaining joints to become faster. In combination with the opposite effect, i.e. making originally slow joints move faster, it becomes possible to partially or fully alter the pacing of the original motion.

The authors plan to continue this research along several directions. As an immediate next step, the current version of the implementation will be tested with a professional dancer who will develop small performance sketches that can be shown in a public demonstration. Letting a dancer experiment with this system and develop small choreographies for it will generate additional insights for the authors and thereby help with the system's future development. As a more long term goal, the authors plan to combine the current system with algorithmic methods for motion synthesis. The current implementation for controlling the orientation of joints makes it very simple to integrate other sources of control than interactive input. One first attempt in this direction will be based on a sound synthesis system that the authors previously developed for a performance with a violinist [44,45]. This sound synthesis system employs multiple simulated mass-spring systems that are coupled with each other. By modelling

the skeleton of the artificial dancer as a mass-spring system, the skeleton can be added to the sound synthesis system via an additional coupling. By doing so, the skeleton can not only be controlled by an additional generative mechanism, but it can also be rendered audible through sonification. After all, the coupling of the mass-spring systems works in both directions and by adding a mechanism that generates higher order harmonics to the oscillations caused by the skeleton, a feedback mechanism can be established in which the synthetic motions of the skeleton and the synthetic sounds mutually influence each other.

To conclude, the proposed approach of combining an autoencoder with interaction for creating synthetic motions for an artificial dancer has shown to be sufficiently promising to warrant further research. But since the current system is likely also suitable for combining autoencoding with any recorded or algorithmically generated data, the scope of this approach might expand beyond digital puppetry and dance. Therefore, this research can potentially contribute to a convergence of two different practices in generative art, those that favour the invention of idiosyncratic rule-based algorithms and those that apply state of the art deep learning methods.

6. References

[1] Bisig, Daniel. 2021. “Granular Dance.” Ninth Conference on Computation, Communication, Aesthetics & X. i2ADS, 176–195.

[2] Rahul, M. 2018. “Review on motion capture technology.” Global Journal of Computer Science and Technology.

[3] Whatley, Sarah, and Hetty Blades. 2019. “Digital Dance.” Bloomsbury Companion to Dance Studies. London: Bloomsbury. DOI: https://doi.Org/10.5040/9781350024489.

[4] Karreman, Laura. 2017. “How does motion capture mediate dance?” In Contemporary Choreography, 492–510. Routledge.

[5] Calvert, Tom. 2013. “The Evolution of Software for Dance.” keynote lecture presented at the conference Corporeal Computing: A Performative Archaeology of Digital Gesture, University of Surrey, Guildford.

[6] Camurri, Antonio, Gualtiero Volpe, Stefano Piana, Maurizio Mancini, Radoslaw Niewiadomski, Nicola Ferrari, and Corrado Canepa. 2016. “The dancer in the eye: towards a multi-layered computational framework of qualities in movement.” Proceedings of the 3rd International Symposium on Movement and Computing. 1–7.

[7] Stavrakis, Efstathios, Andreas Aristidou, Maria Savva, Stephania Loizidou Himona, and Yiorgos Chrysanthou. 2012. “Digitization of cypriot folk dances.” Euro-Mediterranean Conference. Springer, 404–413.

[8] Hsieh, Chi-Min, and Annie Luciani. 2005. “Generating dance verbs and assisting computer choreography.” Proceedings of the 13th Annual ACM international Conference on Multimedia. 774–782.

[9] Leach, James, and Scott Delahunta. 2017. “Dance becoming knowledge: designing a digital “body”.” Leonardo 50 (5): 461–467.

[10] Fdili Alaoui, Sarah, Cyrille Henry, and Christian Jacquemin. 2014. “Physical modelling for interactive installations and the performing arts.” International Journal of Performance Arts and Digital Media 10 (2): 159–178.

[11] Bisig, Daniel, and Pablo Palacio. 2016. “Neural Narratives: Dance with Virtual Body Extensions.” Proceedings of the 3rd International Symposium on Movement and Computing. 1–8.

[12] Carlson, Kristin, Thecla Schiphorst, and Philippe Pasquier. 2011. “Scuddle: Generating Movement Catalysts for Computer-Aided Choreography.” ICCC. 123–128.

[13] Carlson, Kristin, Philippe Pasquier, Herbert H Tsang, Jordon Phillips, Thecla Schiphorst, and Tom Calvert. 2016. “Cochoreo: A generative feature in idanceForms for creating novel keyframe animation for choreography.” Proceedings of the Seventh International Conference on Computational Creativity.

[14] Von Laban, Rudolf. 1975. Modern educational dance. Princeton Book Company Pub.

[15] Maletic, Vera. 1987. Body, Space, Expression: The Development of Rudolf Laban’s Movement and Dance Concepts. Mouton de Gruyte.

[16] Lapointe, François-Joseph, and Martine Époque. 2005. “The dancing genome project: generation of a humancomputer choreography using a genetic algorithm.” Proceedings of the 13th annual ACM international conference on Multimedia. 555–558.

[17] Crnkovic-Friis, Luka, and Louise Crnkovic-Friis. 2016. “Generative choreography using deep learning.” arXiv preprint arXiv:1605.06921.

[18] Kaspersen, Esbern Torgard, Dawid Górny, Cumhur Erkut, and George Palamas. 2020. “Generative Choreographies: The Performance Dramaturgy of the Machine.” Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications-Volume 1: GRAPP. SCITEPRESS Digital Library, 319–326.

[19] Pettee, Mariel, Chase Shimmin, Douglas Duhaime, and Ilya Vidrin. 2019. “Beyond imitation: Generative and variational choreography via machine learning.” arXiv preprint arXiv:1907.05297.

[20] Berman, Alexander, and Valencia James. 2014. “Towards a live dance improvisation between an avatar and ahuman dancer.” Proceedings of the 2014 International Workshop on Movement and Computing. 162–165.

[21] Berman, Alexander, and Valencia James. 2015. “Kinetic Dialogues: Enhancing creativity in dance.” Proceedings of the 2nd International Workshop on Movement and Computing. 80–83.

[22] Berman, Alexander, and Valencia James. 2018. “Learning as Performance: Autoencoding and Generating Dance Movements in Real Time.” International Conference on Computational Intelligence in Music, Sound, Art and Design. Springer, 256–266.

[23] McCormick, John, Kim Vincs, Saeid Nahavandi, and Douglas Creighton. 2013. “Learning to dance with a human.”

[24] McCormick, John, Kim Vincs, Saeid Nahavandi, Douglas Creighton, and Steph Hutchison. 2014. “Teaching a digital performing agent: Artificial neural network and hidden markov model for recognising and performing dance movement.” Proceedings of the 2014 International Workshop on Movement and Computing. 70–75.

[25] McCormick, John, Steph Hutchinson, Kim Vincs, and Jordan Beth Vincent. 2015. “Emergent behaviour: learning from an artificially intelligent performing software agent.” ISEA 2015: Proceedings of the 21st International Symposium on Electronic Art. ISEA 2015, 1–4.

[26] Liu, Lucas, Duri Long, Swar Gujrania, and Brian Magerko. 2019. “Learning movement through human-computer co-creative improvisation.” Proceedings of the 6th International Conference on Movement and Computing. 1–8.

[27] Jacob, Mikhail, and Brian Magerko. 2015. “Interaction-based Authoring for Scalable Co-creative Agents.” ICCC. 236–243.

[28] Habibie, Ikhsanul, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. 2017. “A recurrent variational autoencoder for human motion synthesis.” 28th British Machine Vision Conference.

[29] Holden, Daniel, Jun Saito, Taku Komura, and Thomas Joyce. 2015. “Learning motion manifolds with convolutional autoencoders.” In SIGGRAPH Asia 2015 Technical Briefs, 1–4.

[30] Holden, Daniel, Jun Saito, and Taku Komura. 2016. “A deep learning framework for character motion synthesis and editing.” ACM Transactions on Graphics (TOG) 35 (4): 1–11.

[31] Wang, Qi, Thierry Artières, Mickael Chen, and Ludovic Denoyer. 2020. “Adversarial learning for modeling human motion.” The Visual Computer 36 (1): 141–160.

[32] Li, Zimo, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2017. “Auto-conditioned recurrent networks for extended complex human motion synthesis.” arXiv preprint arXiv:1707.05363.

[33] de Graf, Brad, and Emre Yilmaz. 1999. “Puppetology: Science or cult.” Animation World 3, no. 11.

[34] da Costa Leite, Luís Miguel Barbosa. 2018. “virtual marionette: interaction model for digital puppetry.”

[35] Oshita, Masaki, Yuta Senju, and Syun Morishige. 2013. “Character motion control interface with hand manipulation inspired by puppet mechanism.” Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. 131–138.

[36] Ha, Sehoon, Yunfei Bai, and C Karen Liu. 2011. “Human motion reconstruction from force sensors.” Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 129–138.

[37] Ishigaki, Satoru, Timothy White, Victor B Zordan, and C Karen Liu. 2009. “Performance-based control interface for character animation.” ACM Transactions on Graphics (TOG) 28 (3): 1–8.

[38] Ouzounis, Christos A, Christos Mousas, Christos-Nikolaos

Anagnostopoulos, and Paul Newbury. 2015. “Using Personalized Finger Gestures for Navigating Virtual Characters.” VRIPHYS. 5–14.

[39] Wang, Yumeng, Wujun Che, and Bo Xu. 2017. “Encoder–decoder recurrent network model for interactive character animation generation.” The Visual Computer 33 (6): 971–980.

[40] Roads, Curtis. 2004. Microsound. MIT press.

[41] Schwarz, Diemo, et al. 2004. “Data-driven concatenative sound synthesis.”

[42] Zils, Aymeric, and François Pachet. 2001. “Musical mosaicing.” Digital Audio Effects (DAFx), Volume 2. Citeseer, 135.

[43] Shoemake, Ken. 1985. “Animating rotation with quaternion curves.” Proceedings of the 12th annual conference on Computer graphics and interactive techniques. 245–254.

[44] Bisig, Daniel, and Ephraim Wegner. 2020. “Strings.” Eighth Conference on Computation, Communication, Aesthetics & X. i2ADS, 299 – 312.

[45] Bisig, Daniel, Ephraim Wegner, and Harald Kimmig. 2021. “Strings P.” Ninth Conference on Computation, Communication, Aesthetics & X. i2ADS, 546–553.