xCoAx 2021 9th Conference on
Computation, Communication, Aesthetics & X

2021.xCoAx.org

Granular Dance

Keywords: Generative Art, Dance, Artificial Intelligence, Machine Learning, Granular Synthesis, Concatenative Synthesis

Daniel Bisig
ad5041@coventry.ac.uk
Coventry University, Coventry, United Kingdom

daniel.bisig@zhdk.ch
Zurich University of the Arts, Zurich, Switzerland

This publication presents a tool that can be trained with motion capture data and then used to generate new dance movement sequences. This tool combines two different components: a deep learning model based on a recurrent adversarial autoencoder architecture, and a sequence blending mechanism that is inspired by granular and concatenative sound synthesis techniques. The publication contextualizes this tool with respect to other artificial intelligence inspired approaches in dance. Subsequently, the implementation of the tool is detailed and results from its usage are presented. These results are discussed in terms of their artistic potential. Finally, the publication provides a brief outlook into possible future research directions.

1. Introduction

Research from the field of artificial intelligence (AI) has a long history of providing inspiration and informing novel techniques for creative practitioners, in particular those who work with algorithmic and generative methods. Recent progress in machine learning has led to a surge of interest in data-driven approaches. Compared to the more established rule-based methods that have so far formed the foundations of algorithmic and generative art, data-driven approaches offer different challenges with respect to their adoption by artists. Some of these challenges are related to issues of originality, idiosyncrasy, and mastery. The issue of originality arises from the fact that many machine learning systems excel at imitating the data on which they have been trained. Accordingly, the capability of such systems to create novel and original output is limited. The issue of idiosyncrasy is caused by the large amount of data that is typically required to train deep learning models from scratch. This requirements forces artists to resort to the use of standardized datasets rather than their own personal and unique material. The issue of mastery has to do with the specialized expertise that is required to make informed decisions when modifying existing machine learning models or designing new ones. As a result, many artists are tempted into using off-the-shelf models as black box mechanisms.

The publication tries to address some of these issues by presenting a hybrid tool. This tool combines a machine learning model with a rule-based algorithm for the purpose of generating new dance movement sequences from previously recorded motion capture data. This combination offers a balance between exploiting the impressive imitation capabilities of state of the art deep learning models and the creative development of and experimentation with rule-based algorithms.

The publication starts with an overview of AI-inspired approaches in dance. This overview is divided into two sections, one focusing on artistic motivations and the other on technical principles. It then describes in some detail the tool's implementation. After that, preliminary results are presented that have been obtained through the author's own experimentation with the tool. These results are then discussed in terms of their artistic potential. Finally, the publication concludes with an outlook into possible future research directions.

2. Artistic Background

While artistic applications of machine learning for the purpose of creating imagery and music have garnered much public visibility, the field of dance has an enthusiastic community of its own that experiments with creative uses of machine learning.

The *Open Ended Group* (OEG) has played a pioneering role at the intersection of AI and dance. In 2001, OEG collaborated with choreographer Merce Cunningham on the development of an AI that could record and analyze Cunningham's hand movements for the purpose of controlling live visuals (OEG 2001). In a subsequent collaboration between OEG and choreographer Wayne McGregor, the *Choreographic Language Agent* was created. This software operates as partially autonomous sketchbook that translates phrase-based instructions into abstract geometric animations which can be interpreted by dancers through body movements (Church et al. 2012). In 2016, OEG and Wayne McGregor collaborated again on the development of *Becoming*, an AI-controlled abstract and fully autonomous dancer that was displayed during dance rehearsals (Leach and Delahunta 2017).

Many topics that motivate artistic interest at the intersection of AI and dance are already present in these pioneering examples. These motivations can be roughly grouped into four categories: gain novel insights into dance, enable intuitive forms of interactions, create artificial dancers, and enhance a choreographer's own creativity.

The tool that is presented in this publication is meant to be used in co-creative scenarios. For this reason, the topic of creativity enhancement is addressed in a bit more detail than the other topics.

2.1. Insights into Embodied Creativity

Marc Downie, one of the two members of OEG, proposes in his PhD thesis that metaphors taken from biology and AI can serve as foundations for developing a 'theoretical, technical, and aesthetic framework for the innovative art form of digitally augmented human movement' (Downie 2005). The multi-year interdisciplinary research project *Entity* was initiated in 2000 by Wayne McGergor and dance scholar Scott deLahunta with the purpose of studying the potential of AI to 'broaden understanding of the unique blend of physical and mental processes that constitute dance and dance making' (deLahunta 2009). Mariel

Pettee and colleagues argue that machine learning can be used as tool to 'spark introspection and exploration of our embodied knowledge'. They suggest that machine learning can shift our description of movement away from culturally centred opinions and encourage 'normative discussion about what it means to choreograph' (Pettee et al. 2019).

2.2. Intuitive and Embodied Interfaces

A popular use of machine learning in the context of interactive media performance is to design interactivity through demonstration rather than by specifying rules and algorithms (Gillies et al. 2016). The authors argue that this approach is particularly suitable for creative practices in that it emphasizes the exploratory, playful, embodied, and expressive aspects of the design process (Fiebrink and Caramiaux 2016). One example is a two user training scenario for an interactive artificial dancer in which one user plays the role of the human dancer and the other user performs the artificial dancer's intended responses (Gillies, Brenton, and Kleinsmith 2015).

2.3. Artificial Dancers

AI-inspired methods have also been used for the creation of systems that can be used as autonomous artificial dancers. These methods aim to endow the system with the capability of making creative movement decisions on its own. One example project places an artificial dancer at the center of its artistic concept by exposing the system's learning during the performance to the audience (Berman and James 2018).

2.4. Creativity Enhancement

The integration of AI-inspired methods into software tools has been explored with the purpose of supporting the creative workflow of choreographers and dancers. Here, the biggest potential lies in the development of co-creative systems whose functionality is between that of a creativity support tool and a fully autonomous creative system (Carlson et al. 2016). Many software tools for enhancing a choreographer's creativity have been proposed, a small selection of which is presented here. Kristin Carlson and colleges have developed several tools such as *Scuddle* (Carlson, Schiphorst, and Pasquier 2011) and *Cochoreo* (Carlson et al. 2016). These tools combine genetic algorithms with a fitness function that quantifies movement based on Laban effort qualities and Bartenieff movement patterns. The output of these tools is meant to foster the

exploratory creativity of choreographers. Other researchers have presented deep-learning based software tools that can be trained on a choreographer's own pose or movement material. These tools can generate output that is stylistically similar to the movement material that they have been trained with. For their system Chor-rnn, the authors suggest a form of creativity facilitation that involves the system and the choreographer taking turns in creating movement material (Crnkovic-Friis and Crnkovic-Friis 2016). Similarly, Pettee et al. (2019) present a suite of deep-learning based tools whose output is meant to be more or less directly used for creating a new choreography.

3. Technical Background

The tool presented in this publication is trained to generate short movement sequences for a single dancer which can then be combined into longer sequences. There exists a large diversity of technical approaches for generating synthetic dance movements. Some of these approaches are based on machine learning, others use more conventional statistical approaches, and still others resort to entirely different techniques.

The following section provides a brief overview over some of these techniques. For a much more exhaustive review of machine learning techniques for synthesizing body movements, the reader is referred to Alemi and Pasquier (2019).

3.1. Concatenation and Interpolation

Conventionally, in computer animation and game design, character movements are created either by interpolating between poses that serve as key-frames or by concatenating shorter movement sequences into longer ones. One example of combining these operations with machine-learning is through the use of autoencoders. An autoencoder is an architecture that operates as information bottleneck by encoding and mapping high-dimensional information into a low-dimensional latent-space. To make this compression as lossless as possible, the autoencoder learns to extract the statistically most significant features of the original information. When using an autoencoder, mathematical operations can be conducted in latent-space and the result then converted back through decoding into poses and movement sequences. Some examples of this approach include (Augello et al. 2017; Berman and James 2018). The benefits of this approach over more conventional methods is that a latent space not only reduces the amount of data the computer has to deal with but also captures in its spatial organization some of the fundamental principles of a human body's

morphology and movement capabilities. This can be exploited for a variety of purposes such as: correcting corrupted poses/movements, avoiding movement blending artefacts, and employing euclidean distances as movement similarity measures (Holden et al. 2015).

3.2. Direct Sequence Generation

Alternatively or in combination with the previous approach, machine learning can also be used to directly create movement sequences. Since a movement sequence can be represented as time series, any model that is able to be trained on and predict time series could in principle be used for this purpose. Auto-regressive systems are able to learn sequential relationships in training data which enables them to predict the continuation of sequences. In the context of deep learning, the most frequently used auto-regressive systems are Recurrent Neural Networks, in particular those that maintain and transmit an internal memory state alongside the neurons' regular output such as Long Short-Term Memory (LSTM) Networks (Hochreiter and Schmidhuber 1997) or Gated Recurrent Units (GRU) (Cho et al. 2014). Some example applications of recurrent neural networks for human motion synthesis include (Crnkovic-Friis and Crnkovic-Friis 2016; Li et al. 2017). More recently, recurrent neural networks are facing competition from Temporal Convolutional Networks (Lea et al. 2016) since the latter can handle very long time sequences and be trained in parallel. A comparison between the two approaches for the purpose for movement generation can be found in Pavllo et al. (2019).

3.3. Combined Approaches

Each of the two previously mentioned approaches offers its own benefits and drawbacks. The creation of movement sequences by navigating latent space provides ample possibilities for manual control but makes it difficult to obtain aesthetically convincing movements. Auto-regressive systems excel at creating aesthetically interesting movements but they offer limited means for manual intervention and control.

In two publications, autoencoders and auto-regressive systems are compared from a choreographic point of view. Based on a subjective evaluation of Mixed Density Networks, autoencoders, and LSTMs that have trained on poses and pose sequences, respectively, the authors conclude that only LSTMs perform well on criteria such as posture prediction, temporal coherence, motion consistency, and aesthetics (Kaspersen et al. 2020). Another comparison between

autoencoders and LSTMs places a stronger focus on the creation of movement variations (Pettee et al. 2019). This comparison ends up given more attention to autoencoders than LSTMs.

Accordingly, it seems reasonable to combine auto-regressive systems and auto-encoders. Such a combination has been undertaken by several researchers such as Holden et al. (2015), Fragkiadaki et al. (2015), Habibie et al. (2017), Holden, Saito, and Komura (2016).

3.4. Alternative Approaches

It is worthwhile to mention some entirely different approaches to generate movement sequences. Many of these alternative approaches focus on the agency exhibited by an artificial character and how movement emerges from the interplay between character and environment.

Reinforcement Learning is an approach to machine learning that allows an agent to learn through trial and error from rewards or punishments it receives when interacting with its environment. This approach has for example been used to create locomotion animations across varied and difficult terrain (Peng, Berseth, and Van de Panne 2016).

Other approaches focus on the cognitive plausibility of their models rather than their performance. One example is the work by Infantino et al. (2016) which employs a sophisticated cognitive architecture for the purpose of controlling the movement of a humanoid robot in response to music.

Finally, some researchers follow an Artificial Life approach by implementing a computational ecosystem within which agents struggle for resources. Here, body movements result from behaviors that are selected by agents to increase their chances of survival. An example of this approach is Antunes and Leymarie (2012).

4. Implementation

The system presented in this publication combines a deep-learning model for pose sequence generation with a sequence blending mechanism. The model and the blending mechanism are implemented in Python and make use of the *Tensorflow* machine learning platform (Abadi et al. 2015).

4.1.Machine-Learning Model

The architecture of the machine-learning model is depicted in Fig. 1. The model consists of an encoder, decoder, and discriminator and follows one of the designs proposed by Wang et al. (2020). The encoder takes as input a sequence of poses in which each pose is represented by joint orientations in the form of unit quaternions. This input is passed through a two layer LSTM network followed by a two layer Dense network before being output as latent vector. The decoder operates in reverse. It takes as input a latent vector which is passed through a two layer Dense network followed by a two layer LSTM network before being output as a sequence of poses. The discriminator takes as input a latent vector which passes through a three layer Dense network before being output as scalar value. The purpose of the discriminator is to force the latent vectors to follow a specific prior distribution, which in this case is a Gaussian distribution. It does so by entering into an adversarial game with the encoder in which the discriminator is rewarded for successfully distinguishing between vectors coming from a true Gaussian distribution and latent vectors output from the encoder, whereas the encoder is rewarded for fooling the discriminator. Controlling the prior distribution ensures that the latent space is free of gaps and that distances within it represent a measure of similarity. This ensures that arbitrarily chosen latent vectors can be converted by the decoder into meaningful pose sequences.

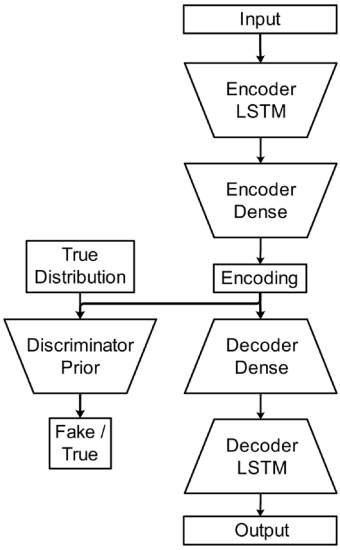


Fig. 1. Architecture of a Recurrent Adversarial Autoencoder. The inputs and outputs of the autoencoder are pose sequences. The trapezoid shapes with which the LSTM and Dense networks are depicted indicate the dimension reduction and expansion that is performed by the encoder and decoder, respectively.

During training, the loss function used for the discriminator is based on the cross entropy between the discriminator's output and a vector of zeros for the encoder's output and a vector of ones for samples taken from a true Gaussian distribution. The autoencoder is trained on four different loss functions that quantify its error in reconstructing a pose sequence and its capability to fool the discriminator. The loss functions associated with the reconstruction error are based on the deviation of quaternions from unit length, the difference in joint orientations between input and output, and the difference between joint positions between input and output. Joint positions are derived from joint orientations through forward kinematics. This combination of quaternion-based joint orientations with a loss function operating on joint positions has been suggested by Pavllo et al. (2019). In contrast to Pavllo et al. (2019) it was found that using both orientation difference and position difference as loss criteria improved the quality of the output.

4.2. Sequence Blending Mechanism

The sequence blending mechanism is inspired by two methods from computer music that combine short sound fragments to generate longer sounds: Granular Synthesis and Concatenative Synthesis. In a nutshell, Granular Synthesis employs very short (microseconds to milliseconds) sound fragments, whose amplitude fades in and out by means of a windowing function. By combining a large number of grains, new sounds can be generated that, depending on the length of the grains, are acoustically more or less similar to the sounds contained within the grains. This approach has been popularized among others by composer Curtis Road (Roads 2004). Concatenative Synthesis is a more recent method. Contrary to the former method, the sound fragments are typically longer (milliseconds to seconds) and their combination is based on finding best matches (Schwarz et al. 2004; Zils and Pachet 2001).

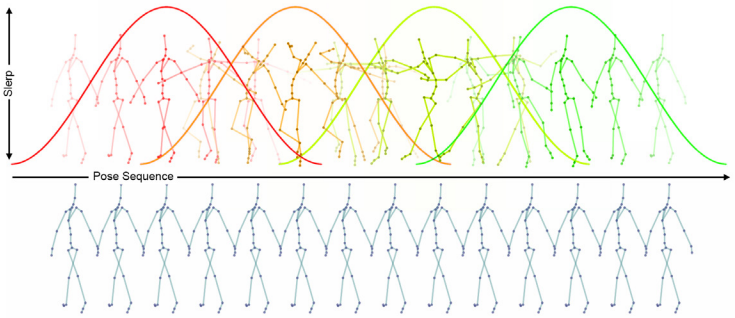


Fig. 2. Pose Sequence Blending. This figure schematically depicts the operation of the pose sequence blending mechanism. Prior to blending, the result pose sequence is populated with a base pose (bottom). Short pose sequences are blended one after the other with the result pose sequence (top) using quaternion SLERP. The bell shaped curves represent Hanning windows which control the amount of SLERP.

For this project, the sequence blending mechanism is used to combine short pose sequences generated by the decoder into longer pose sequences. To obtain smooth transitions between successive pose sequences, two approaches are employed. Similar to Granular Synthesis, a window function (Hanning in this case) is superimposed on the pose sequence. But rather than controlling an amplitude, this function blends the joint orientations of the overlapping pose sequences by spherical linear interpolation (SLERP) (Shoemake 1985). This method is depicted in Fig. 2. Similar to Concatenative Synthesis, sequences are selected for blending based on similarity criteria. Since the latent encodings follow a Gaussian distribution, the euclidean distances between them can be used as measure of similarity between pose sequences. Fig. 4 shows two example distributions of sequence encodings in latent space.

5. Data Acquisition

Training data for machine learning was acquired using a markerless motion capture system (*The Captury*). The recording was conducted at MotionBank, University for Applied Research Mainz. The recorded subjects were professional dancers specialized in contemporary dance. The recording used for training was taken from a single male dancer who was freely improvising to excerpts of music including experimental electronic music, free jazz, and contemporary classic. This recording is about 9.5 minutes in length which corresponds to a sequence of 28600 poses consisting of 29 joints each and taken at 50 frames per second. This data was cleaned using the software *MotionBuilder*.

6. Results

The results presented here stem from experiments with two versions of the machine-learning model. These versions differ with respect to the length of pose sequences they operate on and the encoding dimension. One model works with sequences of 128 poses and an encoding dimension of 64. The other model works with sequences of 8 poses and an encoding dimension of 16. From now on, these models are referred to as model128 and model8. The models have been chosen with two application scenarios in mind. Using sequence blending on the output of mode128 largely preserves the recognizability of the individual sequences with blending having little influence on this. With model8, the recognizability of the individual sequences is mostly lost but blending provides more control on the dynamics of the resulting sequence.

The publication documents results obtained with both models when conducting the following types of experiments: movement reconstruction, latent space organisation, latent space navigation.

6.1. Movement Reconstruction

An obvious thing to do when analyzing a trained machine learning model is to evaluate its performance on data that it has not been trained with. This evaluation provides some insights into the kinds of materials the model works best width and the types of artifacts it introduces. For this analysis, the data has been split into an 80% training set and a 20% validation set. The movement reconstructions tests where conducted on the validation set and involved a subjective comparison between the original and reconstructed pose sequences. An example of a reconstruction test is shown in Fig. 3. Additional reconstruction examples are provided online as videos.[1 2 3 4]

1. https://player.vimeo.com/video/507600887
2. https://player.vimeo.com/video/507600952
3. https://player.vimeo.com/video/507595938
4. https://player.vimeo.com/video/507595896

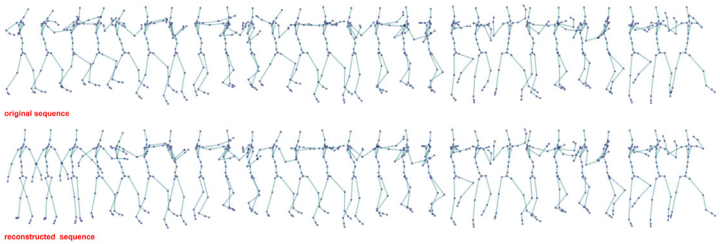


Fig. 3. Pose Sequence Reconstruction Test. The figure depicts the first 30 seconds of an original (top) and reconstructed (bottom) pose sequence with individual poses drawn at ten frames intervals. The deviation between the two sequences at their beginning is due to the SLERP algorithm gradually fading in a pose sequence on top of a base pose with which the result pose sequence has initially been populated with.

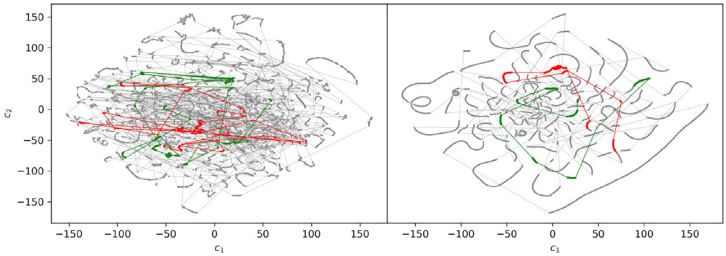
6.2. Latent Space Organization

Gaining an understanding for the organization of latent space forms an important prerequisite for creative experimentation with autoencoders. One approach is to visualize the distribution of the training data within latent space. Such a visualization conveys information about which regions in latent space are densely populated and this in turn points to locations from which familiar or unfamiliar pose sequences can be decoded. Latent space visualizations for model128 and model8 are depicted in Fig. 4.

In latent spaces that follow a Gaussian distribution, the euclidean distance between latent vectors represents a measure of similarity between their decoded outputs. This can be exploited to identify similar pose sequences that smoothly transition when concatenated by sequence blending. Several similarity tests have been conducted based on a pairwise comparison of pose

sequences that follow each other in the original motion capture recording. The results of these tests are available as online videos. Three videos display those paired sequences with smallest euclidean distances between their encodings by model128[5 6 7] and model8.[8 9 10] Another three videos display those paired sequences with largest euclidean distances between their encodings by model128[11 12 13], and model8. [14 15 16]

One of the biggest challenges in working with latent space concerns the typically inapprehensible relationship between latent vectors and their decodings. Usually, there exists no direct correspondence between dimensions of latent space and perceptual aspects of the decoded output. Nevertheless, it is possible to examine this relationship by systematically varying the values of each latent vector dimension, one at a time. Fig. 5 shows such a variation for the first four dimensions for model128. Online videos of variations for the first eight dimensions are available for model128[17 18] and model8.[19 20]



6.3. Latent Space Navigation

A popular approach of using autoencoders for the purpose of movement generation is to navigate through latent space and collect latent vectors along the way which are then decoded and concatenated into a sequence. This approach has been chosen both by researchers working with encodings of poses e.g. Berman and James (2018); Kaspersen et al. (2020); Pettee et al. (2019) and researchers working with encodings of pose sequences e.g. Holden et al. (2015); Holden, Saito, and Komura (2016); Habibie et al. (2017). Using model128 and model8, the following latent space navigation experiments have been conducted: random walk, trajectory offset following, trajectory interpolation.

5. https://player.vimeo.com/video/507947066
6. https://player.vimeo.com/video/507946028
7. https://player.vimeo.com/video/507945397
8. https://player.vimeo.com/video/507962838
9. https://player.vimeo.com/video/507962433
10. https://player.vimeo.com/video/507962060
11. https://player.vimeo.com/video/507947528
12. https://player.vimeo.com/video/507948141
13. https://player.vimeo.com/video/507948603
14. https://player.vimeo.com/video/507960527
15. https://player.vimeo.com/video/507961391
16. https://player.vimeo.com/video/507961782
17. https://player.vimeo.com/video/507668465
18. https://player.vimeo.com/video/507682338
19. https://player.vimeo.com/video/507682930
20. https://player.vimeo.com/video/507683557

Fig. 4. Pose Sequence Encodings in Latent Space. The two figures show two-dimensional representations of the distribution of all encoded pose sequences that have been used for training model8 (left) and model128 (right). For dimension reduction, the t-Distributed Stochastic Neighbouring algorithm has been used. In these figures, individual pose sequences are represented as dots. »

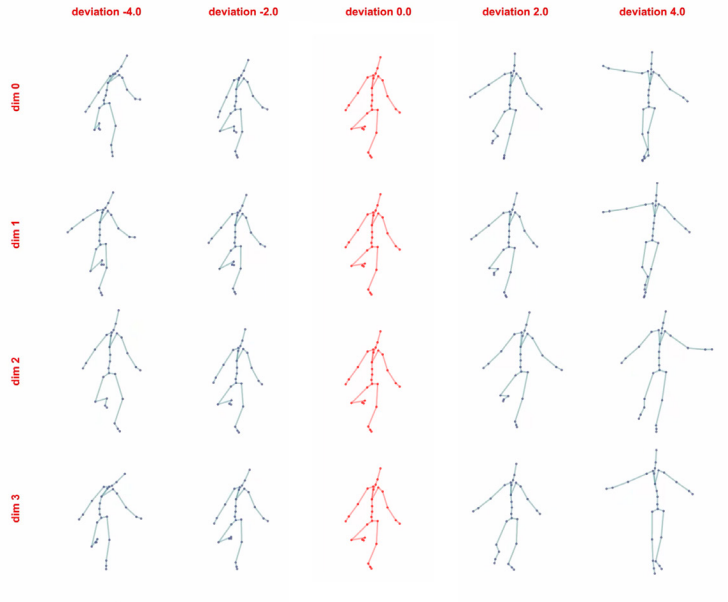
» Thin lines connecting these dots represent pose sequences that follow each other in the original mocap recording. Colored dots and lines highlight those pose sequences which have been used for sequence reconstruction and latent space navigation experiments. All other pose sequences are shown as grey dots and lines.

21. https://player.vimeo.com/video/508401476
22. https://player.vimeo.com/video/508445339

Fig. 5. Value Variations Along Latent Vector Dimensions. The figure shows a single pose from a pose sequence that has been encoded by model128. The latent vector representation of this pose sequence is varied by changing its value for each dimension in turn. In the figure, value changes run along the horizontal axis with no change in the center (red poses), and increasingly negative and positive changes (blue poses) to the left and right, respectively. The dimension increases from top to bottom. For space reasons, only changes for the first four dimensions are shown.

Random Walk

In this experiment, the encoding of a pose sequence is chosen as starting point for a random walk within the neighboring latent space. During the random walk, a random offset is repeatedly added to the latent vector. If the latent vector exceeds a user specified distance limit from the starting position, the offset reflects the vector back towards the starting position. The latent vectors that have been obtained from the random walk are decoded and the resulting pose sequences are concatenated. An example of this approach is shown in Fig. 6. An online video is available for model128[21] and model8.[22]



Trajectory Offset Following

A consecutive set of pose sequences is encoded into a series of latent vectors that describe a trajectory through latent space. Then a user specified fixed offset is added to these latent vectors. This creates a second trajectory that runs at a distance in parallel to the original trajectory. Latent vectors from this second trajectory are then decoded and the resulting pose sequences are concatenated. An example of this approach is shown in Fig. 6. An online video is available for model128[23] and model8.[24]

23. https://player.vimeo.com/video/508402996
24. https://player.vimeo.com/video/508446279

Trajectory Interpolation

An intuitive approach that provides fairly predictable results is to interpolate between two (or more) trajectories through latent space. These trajectories can be obtained for instance by encoding different consecutive pose sequences. A new trajectory can then be created by following the given trajectories while gradually approaching one trajectory and withdrawing from the other. The latent vectors from this new trajectory are then decoded and concatenated. A similar but less predictable approach can be chosen to obtain more original pose sequences. In this case, a new trajectory is created by extrapolating between the given trajectories, i.e. moving away from one trajectory in the opposite direction of the other trajectory. The decoded latent vectors can then be concatenated into a new pose sequence that increasingly exaggerates the differences between the two given pose sequences. Two examples of this approach, one for interpolation and one for extrapolation, are shown in Fig. 6. Online videos of each approach are available for model128[25 26] and model8.[27 28]

25. https://player.vimeo.com/video/508403507
26. https://player.vimeo.com/video/508404186
27. https://player.vimeo.com/video/508449003
28. https://player.vimeo.com/video/508450467

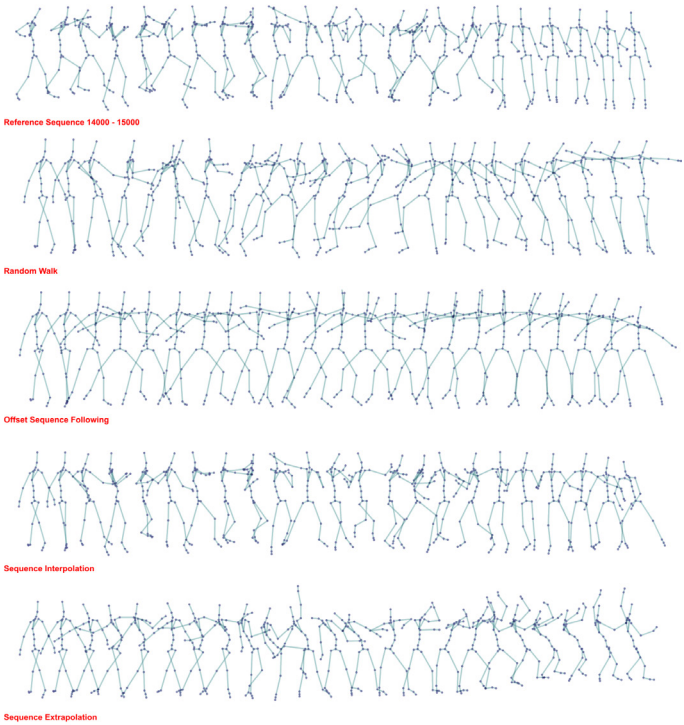


Fig. 6. Latent Space Navigation. The figure depicts several approaches of navigating latent space in combination with sequence blending. The latent space used in this example corresponds to model128. The sequences are depicted as poses taken at intervals of 20 frames. From top to bottom, the sequences represent: an original sequence, a sequence created from a random walk starting at the encoded beginning of the original sequence, a sequence created by adding an offset of 2.0 in all dimensions to all encodings of the original sequence, a sequence obtained by interpolating between the encodings of two original sequences, a sequence obtained by extrapolating between the encodings of the same two original sequences.

7. Discussion

The experiments conducted so far highlight some of the benefits of combining a machine learning system with a sequence blending mechanism for the purpose of creating new pose sequences. The latent space of sequence representations that autoencoders establish offers an interesting environment for exploration and discovery. Since the principle of navigation is easy to understand, users with very different levels of technical expertise can devise their own methods for navigation. The chosen autoencoder only operates on sequences containing a fixed number of poses. The use of sequence blending overcomes this limitation. This blending mechanism is also easy to understand and use. But in addition, it provides the opportunity for creative experiments that draw inspiration from musical approaches of working with Granular and Concatenative Synthesis.

In the following, the results from the previous section are discussed with respect to their artistic potential.

7.1. Latent Space Organization

Visualizations of latent space can grant new insights into movement material that choreographers or dancers are working with. These visualizations can for instance be interpreted in terms of the diversity of material that is available, the duration of movement phrases, the number of phrases in a sequence, or the difference that consecutive phrases exhibit with respect to each other. This information might be helpful to find aligned and contrasting movement phrases that can then be used either sequentially in time or simultaneously for different dancers.

Comparing the similarity of encodings of pose sequences with one's own perception of these sequences raises interesting questions concerning the universality and characteristics of salient movement features. For practical applications, the similarity of encodings can be used as measure of originality of movement material. If movement material ends up in a location within latent space that is sparsely populated, then this movement is under-represented in the material that has been used for training. An alternative and stronger indication of originality is a failure of the autencoder to reproduce the movement. Errors in reproduction are exploited for detecting anomalies, an application of which is the forecasting and prevention of catastrophes such as earthquakes. In dance, an anomaly would be a strong indicator of a very original movement.

The systematic variation of values in a latent vector is a tedious method for creating new movement material. Nevertheless, this approach might be useful for introducing very nuanced deviations in a pose sequence, for instance for the purpose of creating movements for a group of virtual characters in which each character should exhibit some degree of individuality.

7.2. Latent Space Navigation

Navigating a latent space of encodings is a popular method for creating new movement material. This method is useful for a variety of purposes, including data cleaning, the design of behaviors for artificial characters, and choreographic ideation.

The mundane task of data cleaning benefits from the fact that autoencoders discard features that appear seldomly. Therefore, autoencoders can eliminate non systematic artifacts in a mocap recording (Holden et al. 2015). From an aesthetic point of view, this effect might be useful to smooth out small variations or rare extremes in a pose sequence. To achieve either of these goals, latent space navigation would exactly follow the trajectory of encodings from a movement sequence and then reconstruct this movement through decoding and sequence blending (see 6.1 Movement Reconstruction)

Similar random walks in latent space as described in section 6.3 have been employed by Berman and James (2014;2018). In these publications, a random walk is used to create improvisation-like movements for an artificial dancer. Contrary to these previous examples, the random walk presented here operates on pose sequence encodings rather than pose encodings. This requires the use of sequence blending to prevent movement discontinuities. But even with sequence blending, it is difficult to obtain movements that look plausible. Often, the resulting movements are repetitive and erratic. This issue is more pronounced for model8 than model128. To obtain somewhat interesting results, it is necessary to balance the size of the random steps taken in latent space and the size of the overlap used for sequence blending.

Following a trajectory through latent space at a fixed offset provides an alternative to a random walk. This method avoids the occurrence of repetitive movements while still succeeding in creating new movement material. The size of the offset can be used to control the amount of novelty. The examples presented so far are quite rudimentary in that they employ the same offset value for all dimensions of latent space. A more sophisticated approach would take into account

how variations along individual dimensions affect the resulting pose sequence. Also, rather than being fixed, the offset could change while it follows a trajectory. This would result in an output that exhibits varying levels of similarity with the original material. But even the current rudimentary implementation provides some interesting results, in particular concerning the size of the pose sequence that the autoencoder and sequence blending operate on. The result obtained from model128 is a pose sequence that changes minimally and slowly. This is not the case for the result obtained from model8. This points to an interesting difference in application for the models. Model128 is more useful for creating more or less faithful reconstructions of the original movement but generates less interesting results when exploring neighboring regions of latent space. Model8 is more useful for the opposite application.

The interpolation and extrapolation between multiple trajectories constitutes the possibly most productive approach to latent space navigation that is described in this publication. Both methods offer intuitive means of controlling the similarity and variability of the resulting movement material. In case of interpolation, the given trajectories provide boundaries for latent space navigation. In most cases, the regions between those boundaries have become densely populated with encodings during training. Therefore, interpolation typically generates a movement sequence that blends properties of the target sequences in a predictable and plausible manner. In case of extrapolation, the generated results are more varied and unpredictable but this comes at the cost of plausibility and realism. One reason for this is an increased likelihood that extrapolated trajectories cross parts of latent space that have been scarcely populated during training. Extrapolation frequently results in the generation of pose sequences that are neither plausible or realistic and that differ so much from each other that they are difficult to combine by sequence blending. Since interpolation and extrapolation are not mutually exclusive, the strengths and weaknesses of each approach can be balanced against each other.

8. Outlook

The results obtained from combining a recurrent adversarial autoencoder with a grain-based sequence blending mechanism seem promising enough to warrant further research and development. So far, sequence blending has been used to seamlessly concatenate decoded pose sequences. A next step would be to experiment with additional uses of sequence blending. This includes working with a larger range of different sequence lengths and experimenting with more varied sequence combinations such as: non-consecutive placement of grains,

different grain weightings, additive and subtractive grain combinations, stacking multiple grains on top of each other, etc.

It also seems promising to explore additional methods for navigating latent space. The simplest improvement would be to employ more sophisticated versions of random walks. Rather than directly randomizing position offsets, randomization could be applied to first or higher order derivatives to obtain smoother trajectories. More sophisticated approaches could be based on the simulation of flocking behavior. This would allow to create multiple trajectories that are clustered and aligned but still vary from each other. Such trajectories could be used to control the movement of a group of virtual characters. It might also be interesting to extract features from an external modality such as music and use them to control navigation in latent space. Such an approach has been used for instance by (Augello et al. 2017).

It's also worthwhile to address the difficulty of obtaining an understanding for the relationship between latent vectors and their decodings. One approach would be to condition the autoencoder on higher level control parameters (e.g. Wang et al. 2020). Another approach is to extend an autoencoder with a control network that learns to disambiguate latent space (e.g. Li et al. 2017).

The possibly most promising improvement would combine machine-learning with a simulation of the bio-mechanical properties of the human body. Such a combination would get rid of a common problem that plagues purely data-driven approaches: the generation of physically impossible movements. But since training neural networks is based on gradient decent and gradients typically don't propagate through a physics' simulation, a different learning paradigm is needed. This is the paradigm of reinforcement learning. Reinforcement learning is still in its infancy but has recently attracted significant research interest. Accordingly, its likely challenging to come up with an implementation that is both robust and accessible to artists for creative experimentation.

Acknowledgements. The author's thanks go to the dancers who have contributed countless hours of their spare time to the motion capture recordings. Further thanks go to MotionBank for providing their infrastructure. This publication forms part of research conducted in the context of a Marie Curie Fellowship and funded by the European Union.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, and Geoffrey Irving.
2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Alemi, Omid, and Philippe Pasquier.
2019. "Machine Learning for Data-Driven Movement Generation: a Review of the State of the Art." arXiv preprint arXiv:1903.08356.

Antunes, Rui Filipe, and Frederic Fol Leymarie.
2012. "Generative choreography: animating in real-time dancing avatars." International Conference on Evolutionary and Biologically Inspired Music and Art. Springer, 1–10.

Augello, Agnese, Emanuele Cipolla, Ignazio Infantino, Adriano Manfre, Giovanni Pilato, and Filippo Vella.
2017. "Creative robot dance with variational encoder." arXiv preprint arXiv:1707.01489.

Berman, Alexander, and Valencia James.
2014. "Towards a live dance improvisation between an avatar and a human dancer." Proceedings of the 2014 International Workshop on Movement and Computing. 162–165.

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.

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.

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

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
2014. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078.

Church, Luke, Nick Rothwell, Marc Downie, Scott DeLahunta, and Alan F Blackwell.
2012. "Sketching by Programming in the Choreographic Language Agent." PPIG. Citeseer, 16.

Crnkovic-Friis, Luka, and Louise Crnkovic-Friis.
2016. "Generative choreography using deep learning." arXiv preprint arXiv:1605.06921.

DeLahunta, Scott.
2009. "The choreographic language agent." Conference Proceedings of the 2008 World Dance Alliance Global Summit.

Downie, Marc Norman.
2005. "Choreographing the Extended Agent: performance graphics for dance theater."Ph.D. diss., Massachusetts Institute of Technology, School of Architecture and Planning .

Fiebrink, Rebecca, and Baptiste Caramiaux.
2016. "The machine learning algorithm as creative musical tool."arXiv preprint arXiv:1611.00379.

Fragkiadaki, Katerina, Sergey Levine, Panna Felsen, and Jitendra Malik.
2015. "Recurrent network models for human dynamics." Proceedings of the IEEE International Conference on Computer Vision. 4346–4354.

Gillies, Marco, Harry Brenton, and Andrea Kleinsmith.
2015. "Embodied design of full bodied interaction with virtual humans." Proceedings of the 2nd International Workshop on Movement and Computing. 1–8.

Gillies, Marco, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, et al.
2016. "Human-centred machine learning."Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. 3558–3565.

Open Ended Group.
2001. Loops. http://openendedgroup.com/artworks/loops.html. Accessed: 2021-01-29.

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.

Hochreiter, Sepp, and Jürgen Schmidhuber.
1997. “Long short-term memory.” Neural computation 9 (8): 1735–1780.

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.

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

Infantino, I, A Augello, A Manfré, G Pilato, and F Vella.
2016. “Robodanza: Live performances of a creative dancing humanoid.” Proceedings of the Seventh International Conference on Computational Creativity. 388–395.

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.

Lea, Colin, Rene Vidal, Austin Reiter, and Gregory D Hager.
2016. “Temporal convolutional networks: A unified approach to action segmentation.” European Conference on Computer Vision. Springer, 47–54.

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

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.

Pavllo, Dario, Christoph Feichtenhofer, Michael Auli, and David Grangier.
2019. “Modeling human motion with quaternion-based neural networks.” International Journal of Computer Vision, pp. 1–18.

Peng, Xue Bin, Glen Berseth, and Michiel Van de Panne.
2016. “Terrain-adaptive locomotion skills using deep reinforcement learning.” ACM Transactions on Graphics (TOG) 35 (4): 1–12.

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

Roads, Curtis.
2004. Microsound. MIT press.

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

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

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

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