Automatic generation of realistic connetion between arbitrary CG motions using convolutional filter

Takuya Kato, Yoshitsugu Manabe, Noriko Yata
Graduate School of Advanced Integration Science
Chiba University
Chiba, Japan
katotaku@chiba-u.jp, manabe@faculty.chiba-u.jp, yata@chiba-u.jp

***Abstract*—By the progress of Computer Graphics(CG) technology, it has become possible to create high quality human CG objects. To make them operate real, high quality motion data is needed. Generally, motion data is created by a motion capture system. However, the cost and time required for creating many motion data can't be ignored. In particular, when connecting motions, it takes time and effort to adjust. Addition, connecting motion as not to cause discomfort is very difficult. When the number of motion data is n, there are n×n connections of motion in all. It is not realistic to manually adjust all of them. Therefore, this research proposes a method to automatically generate connection between arbitrary motions using convolutional filter. It's considered that the efficiency of CG motion creation can be improved by means of the proposed method.**

Keywords—CG motion; machine learning; convolutional autoencoder; convolutional network, Naginata-Do

I. INTRODUCTION

With precise motion capture system, creation of human CG motion has become easy. However, in order to connect the created motions without discomfort, a person has to adjust the motion data. For contents created by connecting short motions like dancing and action games, it is necessary to adjust as many as the number of connected motions. Fukayama succeeded in automatically generating dance motion adapted to music by machine learning a large amount of dance data [1]. However, this method can't create a dance motion using several selected arbitrary choreographies because all the choreography is automatically generated stochastically. Holden et al. proposed a method that creates a convolutional filter that detects distinctive motion by using a convolutional autoencoder [2]. However, this method requires about 10 hours of walking motion data. This research proposes a method for naturally connecting arbitrary motions by means of a machine learning using convolutional filter. However, since human motion is complex and has a large variety, this research limits the motion to be handled to the operation of "Naginata-Do" which is a traditional Japanese Budo (Naginata-Do is a Budo dealing with weapon similar to 2 meters of spear).

II. RELATED WORKS

"Machine Dancing" is a method for automatically generating human CG motion [1]. In this method, "dance vocabulary" is created by machine learning of dance data associated with music data. Dance vocabulary is the basic form of dance movement, and generated as many as the number of distinctive actions by clustering. And dance motion is automatically generated by combining dance vocabulary stochastically according to music. In addition, by learning the dance vocabulary, it is also possible to newly generate a dance action that has not been input as learning data. Thus, even a person who can't create CG motion can easily let the CG character dance. However, since the dance action is automatically generated, it is difficult for the user to create a dance with arbitrary choreography.

The dance vocabulary represented a distinctive part in the dance movement. On the other hand, in the image recognition field, it is well known that features such as edges and corners can be detected by a convolutional filter. Holden et al. applied this, and created convolutional filter that detects distinctive motion by using a convolutional autoencoder [2]. Figure 1 shows convolutional filter.

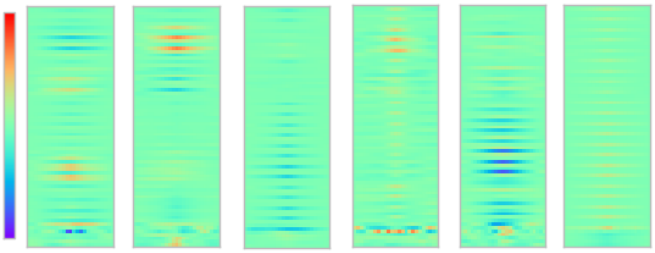

Figure 1. Convolutional filter that detects distinctive motion. The horizontal axis represents time, and the vertical axis represents the degrees of freedom [2].

III. PROPOSED METHOD

This research consists of two stages. First, convolutional filter that detects distinctive motion is created by convolutional autoencoder. Next, convolutional neural network is created with convolutional filter. When this network inputs two motions, it outputs a motion smoothly connecting them.

A. The Motion Dataset for Learning

The purpose of this research is to connect arbitrary motions. However human motion is complex and has a large variety.

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

Therefore, this research limits motion data to be handled to Naginata-Do which is a traditional Japanese Budo dealing with weapon similar to spear. Since the feature of Naginata-Do is smooth movement, it is considered to connect arbitrary motions.

Motion dataset for learning is created using a motion capture system (Perception Neuron [3]). Model of created motion data has 59 joints and each joint have 6 degrees of freedom (translation: 3, rotation: 3). Frame rate is 120 fps.

These motion data are converted into shapes suitable for learning. In order to reduce the amount of data, joints are reduced to 21 and degrees of freedom are three translational movements. Figure 2 shows using model.

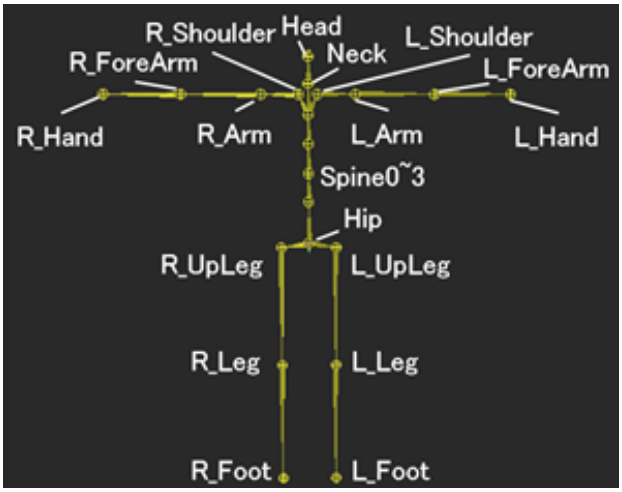

Figure 2. Model with 21 joints

Since the coordinate value of Hip represents the world coordinate of this model, the value range is very wide compared with other joints. Using this value has a high possibility of adversely affecting learning, so special processing is done. First, the translational component is converted into velocity by subtracting the current value from the value of the next frame. Then, the rotation speed about the Y-axis is similarly obtained. Hip has only 4 degrees of freedom and model has 64 degrees of freedom in total according to these process.

B. Generation of Convolutional Filter

Convolutional filter that detects distinctive motion is created by convolutional autoencoder. Figure 3 shows image of convolutional autoencoder. This filter is a two-dimensional filter with time and degrees of freedom, but the convolutional autoencoder performs a one-dimensional convolution over the temporal domain.

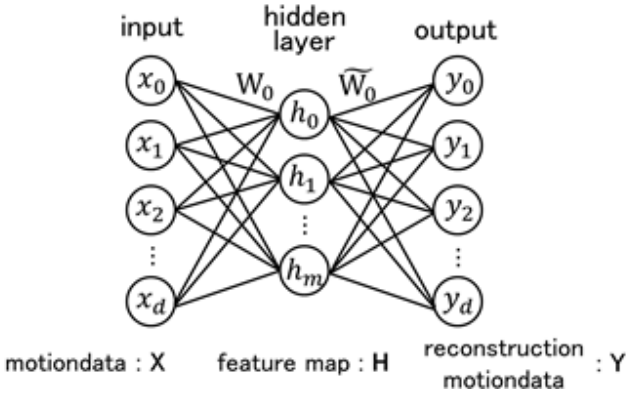

Figure 3. Convolutional autoencoder

This network consists of a forward operation Φ(encoding) and a backward operation $\boldsymbol{\Phi}^{\dagger}$ (decoding). The input vector to network is **X** and the outputs of forward operation is **H**. **H** is generally called a feature map. Here, $\mathbf{X} \in \mathbb{R}^{n \times d}$, and $\mathbf{H} \in \mathbb{R}^{n \times m}$, where n is the number of frame to input, d is degrees of freedom of model, m is the number of creating convolutional filter.

The forward operation:

$$\boldsymbol{\Phi}(\mathbf{X}) = ReLU(\mathbf{X} * \boldsymbol{W}_0 + \boldsymbol{b_0}) \qquad (1)$$

In the equation, $(*)$ represents a convolution process, $\boldsymbol{W}_0 \in \mathbb{R}^{m \times d \times w_0}$ is a weight matrix, and $\boldsymbol{b}_0 \in \mathbb{R}^m$ is a bias, where w_0 is the temporal filter width. ReLU(x) is nonlinear operation that outputs max(x, 0).

The backward operation:

$$\mathbf{Y} = \boldsymbol{\Phi}^{\dagger}(\mathbf{H}) = (\mathbf{H} - \boldsymbol{b_0}) * \widetilde{\boldsymbol{W}_0} \qquad (2)$$

$\mathbf{Y} \in \mathbb{R}^{n \times d}$, and $(* \widetilde{\boldsymbol{W}_0})$ represents deconvolution with weight $\boldsymbol{W}_0$. Using this network, learning is performed so as to minimize the error between the input **X** and the output **Y**. Then convolutional filter is created. Since the ideal convolutional filter is sparse as shown in the figure 4, learning is performed by adding a sparse term to the loss function so that the network parameter $\theta = \{\boldsymbol{W}_0, \boldsymbol{b_0}\}$ is minimized.

The loss function:

$$\mathrm{Loss}(\mathbf{X}, \theta) = \|\boldsymbol{X} - \boldsymbol{Y}\|_2^2 + \alpha\|\theta\|_1 \qquad (3)$$

In the equation, the first term represents the square error and the second term is the sparse term as described above. α is the learning rate and is a sufficiently small value of 1 or less. The network is implemented by using "chainer", and Adam's method [4] is used for the optimization method, and to avoid overfitting, Dropout's method [5] is used.

Figure 4. The ideal convolutional filter. The value of the green part is 0 and it shows that this filter becomes sparse.

C. Generation of Connection between Motions

Using a created filter, a feed forward network that connects arbitrary motion data smoothly is created. Figure 5 shows image of feedforward network.

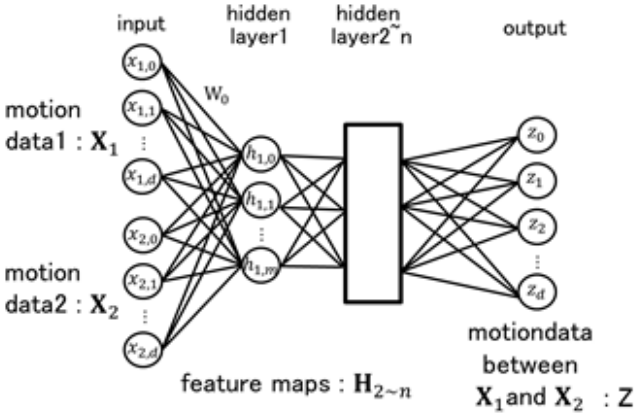

Figure 5. Image of feedforward network

The input to this network is a $n\times d$ dimensional vector of the two vectors $\mathbf{X_1} \in \mathbb{R}^{n\times 0.5\times d}$ and $\mathbf{X_2} \in \mathbb{R}^{n\times 0.5\times d}$. The output is $\mathbf{Z} \in \mathbb{R}^{n_z\times d}$, where n_z is the number of frames of motion data to be output. $\mathbf{X_1}$ and $\mathbf{X_2}$ represent the end part and the start part of the motion data to be connected. $\mathbf{Z}$ represents a motion data complementing them. Therefore, $\mathbf{X_1}$, $\mathbf{Z}$ and $\mathbf{X_2}$ are joined in order, and smoothly connected motion data can be obtained.

The feedforward operation :

$$\mathbf{Z} = \mathbf{\Omega}(\mathbf{X_1}\&\mathbf{X_2}) = ReLU(ReLU(ReLU((\mathbf{X_1}\&\mathbf{X_2}) * \boldsymbol{W}_0 + \boldsymbol{b_0}) * \boldsymbol{W}_1 + \boldsymbol{b}_1) * \boldsymbol{W}_n + \boldsymbol{b}_n) \quad (4)$$

In the equation, (&) represents the synthesis of vectors, and $\mathbf{X_1}\&\mathbf{X_2}$ is a vector of dimension $n\times d$. Using this network, learning is performed so that the following loss function is minimized. Teaching data is represented by $\mathbf{T}$.

The loss function :

$$\mathrm{Loss}(\mathbf{Z},\mathbf{T}) = \|\boldsymbol{Z} - \boldsymbol{T}\|_2^2 \quad (5)$$

$\mathbf{T}$ used for learning is the motion data between $\mathbf{X_1}$ and $\mathbf{X_2}$ in the motion data set. The image of input and output of the feedforward network at learning is shown in the figure 6.

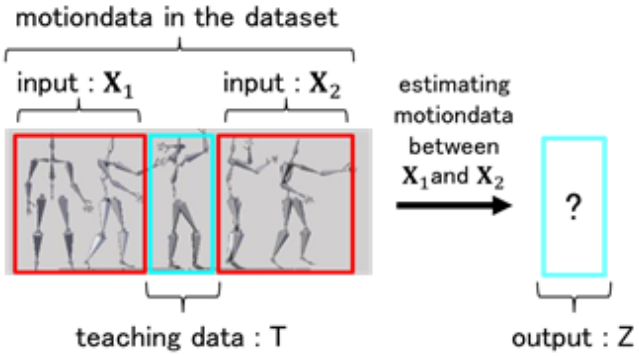

Figure 6. Image of input and output of feedforward network at learning

Thereby, when two kinds of motion data which are not temporally continuous are input, a network which outputs motion data for interpolating them is created.

IV. Conclusion

This paper proposes a method using a convolutional filter in order to connect arbitrary motion data naturally. As a result, improvement of the efficiency of CG motion creation can be expected. Currently, it is possible to acquire a sparse convolutional filter by convolutional autoencoder. However, the reconstructed motion data using this filter is unnatural in motion compared to the original motion data. Therefore, it cannot be thought that the distinctive motion of Naginata-Do can be completely detected by this convolutional filter. In the future, we will optimize hyperparameters and improve the convolutional filter. Then, we create a feedforward network.

V. References

[1] S.Fukayama, M.Goto, Automated Choreography Synthesis Using a Gaussian Process Leveraging Consumer-Generateed Dance Motions, Proceedings of the 12th IEEE international Conference on Signal Processing(ICSP2014),Nov.2014.

[2] D.Holden, J. Saito, T. Komura, A Deep Learning Framework for Character Motion Synthesis and Editing, SIGGRAPH '16 Technical Paper, July 24-228, 2016, Anaheim, CA

[3] "Perception Neuron by Noitom",< https://neuronmocap.com/>(2017-10-23

[4] Diederik P.Kingma, Jimmy Lei Ba, Adam: A method for stochastic optimization, conference paper at ICLR, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research, Volume 15 Issue 1, Pages 1929-1958, January