Motion Generation Using Combination of Motion Graphs and Key Pose Selection With Statistical Models

Shuntaro Kono, Kunio Yamamoto, and Masaki Oshita

Kyushu Institute of Technology, 680-4 Kawazu, Izuka-shi, Fukuoka, 820-8502, Japan

Abstract. In computer animation, generating motion is difficult. There is a need for a technology to generate actions by reusing motion data in animation and game production. In this study, we develop a system that allows users to generate motions interactively by specifying the types of motions and detailed features based on existing motion data. Because there is no need for new motion capture, this system is useful for quickly creating animation prototypes. The idea of the system is to map key postures in motion to a latent space such that the user can control the posture used for motion generation. Furthermore, detailed posture features can be specified interactively if necessary. The set of original motions and key postures in the motion are input in advance. At runtime, the user specifies the types and characteristics of the motions. A motion graph is used to generate and output the motions. We created a prototype of the system and confirmed that it can generate motions by specifying key postures.

Keywords: Computer Animation, Motion Synthesis, Interaction.

1 Introduction

In animation and game production, motion data are generally acquired through motion capture, which is costly and time consuming. Reusing existing motion data to generate motion would be more efficient. The system developed in this study is particularly effective for generating continuous motions in animation and game production, and for quickly creating prototypes of character motions. Sequential motions are those in which multiple types of actions, such as fighting and dancing, are performed sequentially. The method using motion graphs [1] has a problem in that the user cannot generate motions by specifying the front and back postures. The deep learning method [2,3] requires the user to create and provide a posture as a constraint condition. The purpose of this study was to develop a system that allows users to generate motions interactively by specifying the type and characteristics of the motion.

The pre-input for this system is a set of motions and the user-specified time of a key posture during the motion. The runtime input comprises the type and characteristics of the actions specified by the user. The outputs are the motions generated by the user operations.

The issue whereby users cannot generate motions by specifying the front and back postures in the method of [1] using motion graphs is solved by mapping the key postures in the original motion to the latent space so that users can select a posture from among them and map that posture to an edge in the motion graph. The issue of the user having to create and provide a posture as a constraint condition in the method of [2,3], which uses deep learning, is solved by selecting a key posture to be used for motion generation from the candidate postures displayed in the latent space, and by specifying additional detailed features of the posture as needed so that the constraint conditions can be provided intuitively.

A motion graph is created from a set of pre-input motions. From the set of pre-input motions and times of the key postures, the key postures are obtained and mapped to a two-dimensional latent space for each type of posture using the multidimensional scaling method. The key postures are mapped to the edges of the motion graph. When a user selects a key posture from the latent space, the system finds the corresponding edge in the motion graph, searches for the path to the edge corresponding to the previous key posture, and uses it for motion generation.

In this study, we created a prototype and generated motion. We confirmed that the motion graph could generate motions by specifying key postures in the latent space. We demonstrated that the prototype system can be used as an interface for users to select a key posture by mapping the key posture using a statistical model. We also confirmed that the motion graph can be used to generate motions based on the key postures specified by the user. As an issue to be addressed before the final version of this system is created, we will map each type of key posture so that users can select detailed features as conditions when selecting a key posture. In addition, it should be possible to select a key posture while viewing the position where the motion is actually generated.

2 Related Work

Kovar et al. [1] proposed a method called motion graph, which reconstruct motion-captured motion data and synthesize new motion data. When motion-captured motion data are input, an effective graph known as a motion graph is constructed automatically. The nodes in the motion graph are the selection points for the transitioning motion, whereas the edges are fragments of the original motion data. When the user enters conditions such as the type of motion to be generated and the position and orientation of the character, the paths in the motion graph that satisfy these conditions are searched and the motion is generated. However, this method does not allow users to control the specification of motions and postures interactively. This research aims to solve this problem by providing an interface that allows the user to control the motion interactively.

Qin et al. [2] proposed a method for generating behavior transitions using two Transformer encoder-based networks, given multiple context frames and one target frame. Tang et al. [3] proposed a method for generating motion transitions using motion manifolds and conditional transitions. Li et al. [4] proposed a method to generate a variety of motion data from existing motion data by performing multi-step motion matching. Starke et al. [5] proposed a method that uses deep learning to generate trajectories of various future motions from trajectories of past motions. These trajectories are added, overwritten, and mixed to create layers of animations and generate trajectories of future motions. Our study uses motion graphs for motion generation. By using motion graphs, it is possible to output a sequence that combines motion data that matches part of the original motion data and satisfies the posture given as a constraint condition. These conventional methods using deep learning and motion matching have a wide range of motions that can be generated. In contrast, the method using motion graphs does not require specific specification of the posture, although it is difficult to specify the posture specifically, and the position and orientation of the posture can be determined automatically.

Choi et al. [6] developed a system that allows users to view the overall movement by having the system display a stick figure generated from a database. Users can search for movements from the database of movement data by drawing a stickman. The search results are updated while the user is drawing the stick figure, allowing for interactive searching. Although a stickman alone does not provide information about the overall motion, the idea is proposed to solve this problem by placing the stickman on a two-dimensional space corresponding to the scene in which the motion was captured. Sakamoto et al. [7] proposed a method to display postures in motion data on a motion map. By selecting the posture, the user can obtain a part of the motion data by specifying the postures in order. The motion map uses a self-organizing map. In this study, we use a multidimensional scaling method to map the posture to a two-dimensional latent space.

Zhang et al. [8] proposed a method for generating motion using Vector Quantised-Variational AutoEncoder (VQ-VAE) and Generative Pretrained Transformer (GPT). Guo et al. [9] proposed a method for sampling motion lengths from input text and using time-variant autoencoders to generate a variety of motion data of the sampled lengths. It is able to capture the local context of actions and generate actions that are faithful and natural to the input text. The method for generating actions from these texts is based on the input texts ``A man rises from the ground, walks in a circle and sits back down on the ground." or ``Someone is walking forward and holding a handrail very carefully, as if they are afraid of failing." Although it is easier to specify actions using text for motion generation, it is not possible to specify actions in as much detail as with methods using deep learning or motion graphs.

Arikan and Forsyth [10] proposed a method that uses motion graphs to generate multiple actions that satisfy multiple given constraints and gives them as choices. The constraints given by the user are the length of the motion, the position and orientation of the body in a particular frame, a particular state of the joints (e.g., head, hands) in a particular frame, or the posture of the entire body in a particular frame.

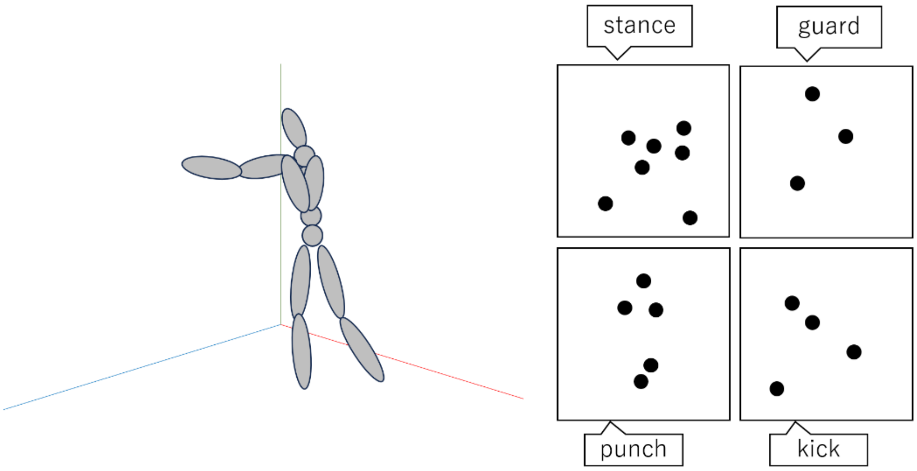


Fig. 1. Interface.

3 Proposed Method

3.1 Key Posture Selection Using Statistical Models

The system developed in this study can be employed to reuse existing motions and to create new motions while controlling them. As shown in Fig. 1, this method maps the key postures in motion to a latent space. Users can control the type of motion generated by selecting a posture from the latent space. Detailed posture characteristics can also be specified if necessary. This allows the user to see which movements and postures are included in the system in the latent space, which is useful for controlling the movements to be generated.

3.2 System Overview

The overview of the system is shown in Fig. 2. The inputs are divided into two categories: pre-input and runtime input. The pre-input of the system is a set of actions and the times of key postures during these actions. The runtime input comprises the type and characteristics of the actions specified by the user. The output is a single behavior generated by the operation of the user.

For example, let us assume that this system is used to generate actions in an animation in which fighting actions are performed. It is assumed that the output actions perform a punch, kick or guard against an opponent's attack. Therefore, examples of input motions are stance, forward, backward, punch, kick, and guard. Each type of motion should contain at least 10 key postures so that the user can select one of them as a candidate for motion generation.

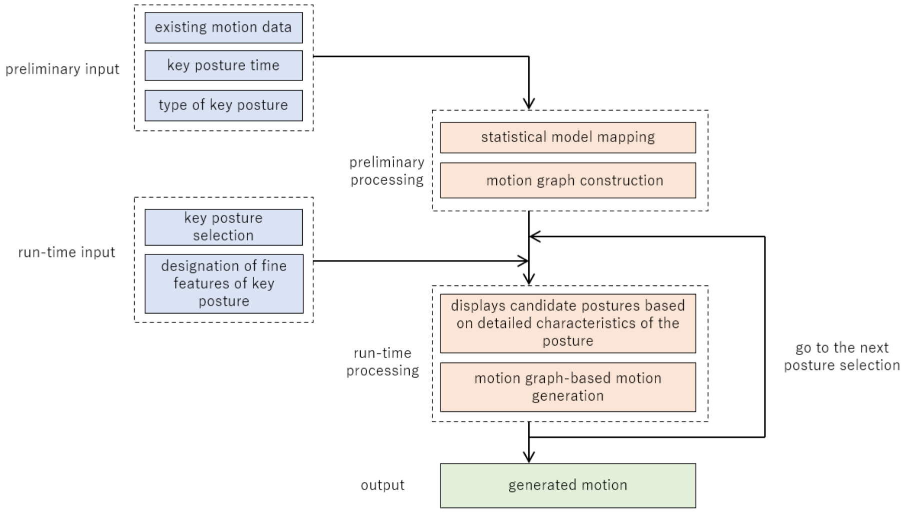

preliminary input
existing motion data
key posture time
type of key posture
preliminary processing
statistical model mapping
motion graph construction
run-time input
key posture selection
designation of fine features of key posture
run-time processing
displays candidate postures based on detailed characteristics of the posture
motion graph-based motion generation
go to the next posture selection
output
generated motion

Fig. 2. System overview.

3.3 Interface

Once the set of movements and time of the key posture are pre-input, a latent space is created for each type of key posture, as shown in Fig. 1. Users can select a posture in the latent space to obtain an approximate posture for the entire body.

Next, if the user wishes to specify a detailed posture, a posture can be selected in two manners. First, as shown in Fig. 3, multiple postures close to the posture specified in the latent space are displayed in the three-dimensional (3D) space, and a posture can be selected from among them.

The second method, as shown in Fig. 4, is to specify a position and orientation in 3D space, and postures close to these conditions are displayed in 3D space and latent space, from which a posture is selected. The same posture is visually represented in the same color.

3.4 Statistical Model Mapping

The key postures are obtained from the times of the key postures during the pre-input motion. These postures are mapped to a two-dimensional latent space for each posture type using a multidimensional scaling construction method.

The similarity between the two mapped postures is determined using the posture distance. The posture distance is obtained as follows:

$$D = \min_{\theta,x_0,z_0} \Sigma_i \left|\left|\boldsymbol{p}_i - \boldsymbol{T}_{\theta,x_0,z_0}\boldsymbol{p}'_i\right|\right|^2 \qquad (1)$$

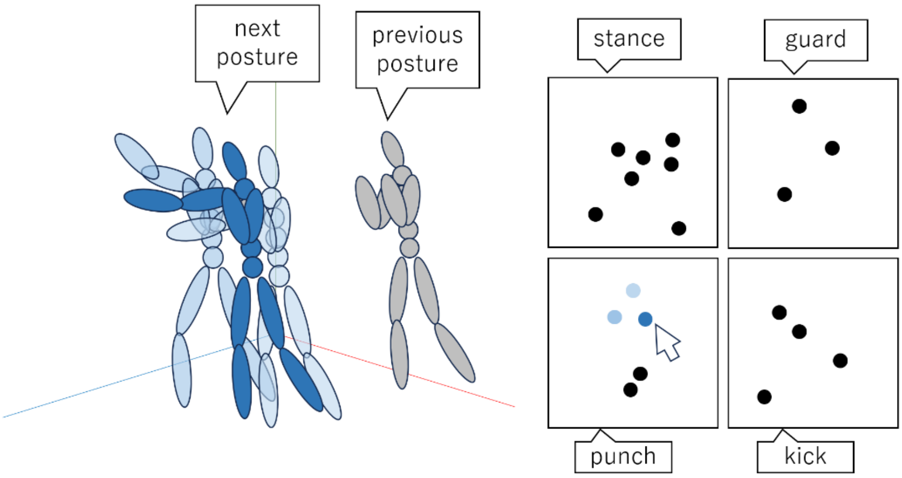


Fig. 3. A method of displaying multiple postures in 3D space that are close to a specified posture in potential space and selecting a posture from among them.

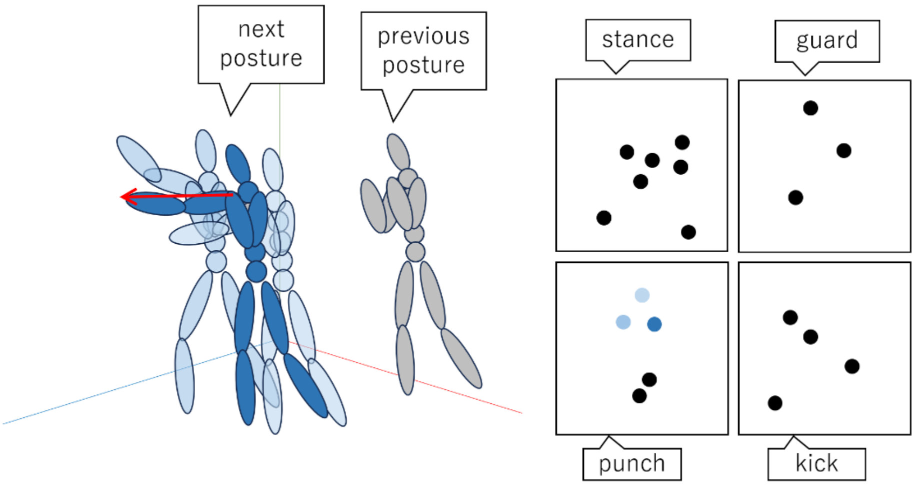


Fig. 4. When the position and orientation are specified in 3D space (red arrows in the figure), postures close to these conditions are displayed in 3D space and potential space, and a posture can be selected from among them.

where D is the posture distance, i is the joint number, $\boldsymbol{p}$ is the joint position of the first posture, and $\boldsymbol{p}'$ is the joint position of the second posture. The second posture is deformed so that the posture distance D between the two posture is minimized. The x-axis displacement of this deformation is x_0 and the z-axis displacement is z_0. The amount of rotation around the y-axis is θ.

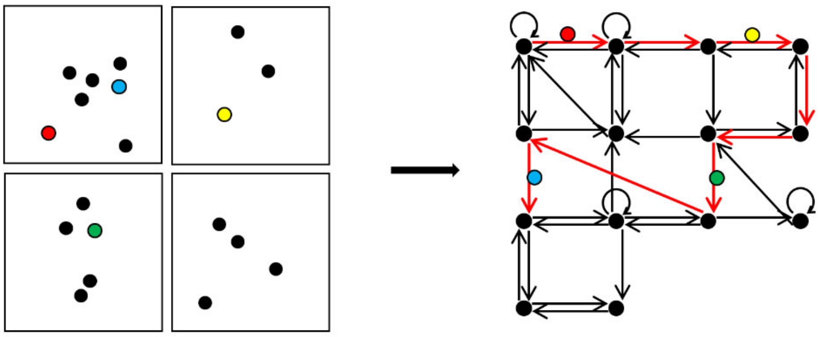
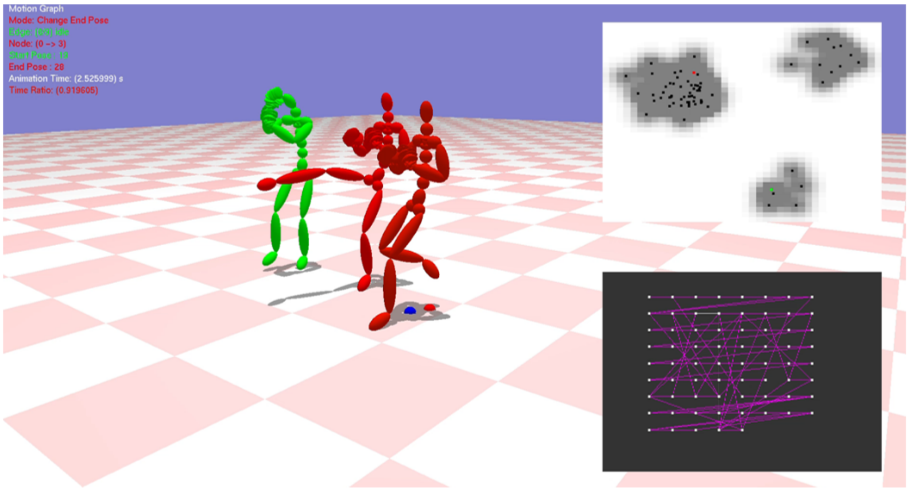


Fig. 5. The motion graph is used to generate motion from the key postures selected in the latent space. Suppose that the key postures selected by the user are red, yellow, green, and blue, in that order (left). The edges on the motion graph that contain these key postures are found (right). A path (red path) that connects these edges in the shortest possible time is determined and used for motion generation.

3.5 Motion Graph-based Motion Generation

Based on the time of the pre-input key posture, the key posture is mapped to the edge in the motion graph in which it is contained. The edges of the motion graph corresponding to the key postures selected by the user are determined. The path that connects these edges in the motion graph in the shortest time is found. The motions corresponding to the edges in the path are generated in that order. This process is shown in Fig. 5.



Motion Graph
Mode: Change End Pose
Edge: (8/9) idle
Node: (0 -> 3)
Start Pose : 19
End Pose : 28
Animation Time: (2.525999) s
Time Ratio: (0.919605)

Fig. 6. Operation screen.

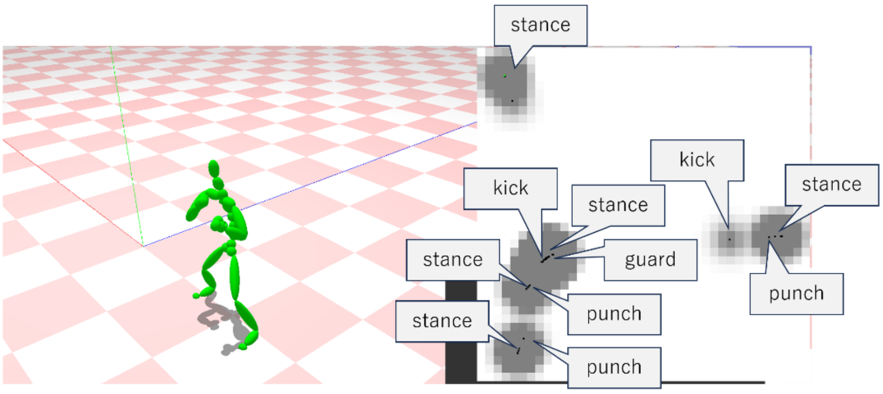


Fig. 7. Potential space generated by operation verification.

4 Experiments and Evaluations on Prototypes

A prototype of the system was created. The user can specify two postures, and the system generates movements between them. The function for specifying detailed postures has not yet been implemented, and the user specifies the posture that he/she wishes to use directly from the latent space. The prototype operation screen is shown in Fig. 6. The latent space is displayed in the upper right corner and the motion graph in the lower right corner.

To verify the system operation, 71 s of fighting movements were input, including six types of movements such as stance, punch, kick, and guard. A total of 72 key postures were used as inputs. The motion graph generated by the system had 74 nodes and 245 edges. The results of the mapping to the latent space are shown in Fig. 7.

By verifying the operation, the prototype could specify the starting and ending postures and generate motion. Table 1 shows the relationship between the specified posture and generated motion. The specified posture and generated motion are shown in Fig. 8. Although the current prototype maps all key postures to a single latent space, it is useful for specifying postures because it allows us to determine approximately which region in the latent space corresponds to which posture. If the latent space is mapped separately for each type of key posture in future implementations, the distribution of key postures is expected to be understood in more detail. One problem is that the generated movements include unspecified movements. This can be improved by providing a motion input of sufficient length and performing a motion graph search for the path with the shortest motion playback time.

Table 1. Operation verification result.

Motion	Start	End	Generated motion
1	Punch	Guard	Punch, Backward, Kick, Forward, Guard
2	Guard	Kick	Guard, Punch, Kick
3	Stance	Punch	Stance, Punch, Guard, Kick, Forward, Punch

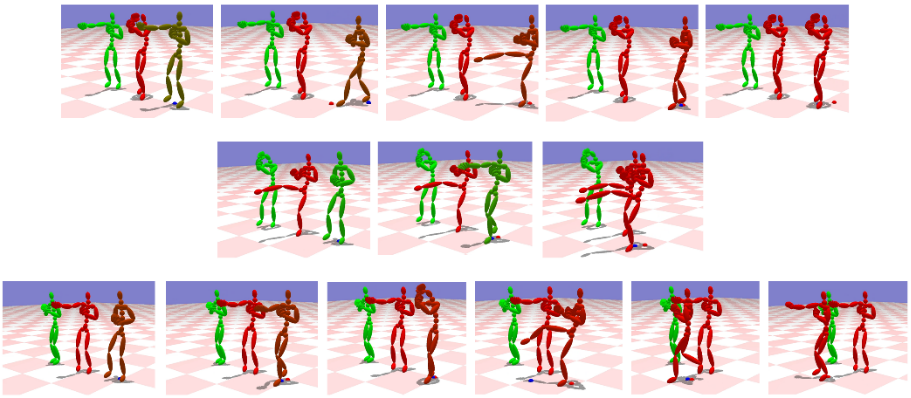


Fig. 8. Operation verification result. The motions are the same as in Table 1: motion 1 (top row), motion 2 (middle row), and motion 3 (bottom row). The left and right are the start and end times, respectively, whereas the time in between is the time during the generated motion. In each frame of the image, the green character in the far left is the starting posture specified by the user, the red character in the center is the ending posture specified by the user, and the character in the front right is the posture during the generated motion.

5 Conclusion

Using motion graphs and statistical models, we developed a system that can reuse actions to generate actions that can be controlled by users. A prototype of the system was developed and we confirmed that users can efficiently specify postures by mapping postures in motion to a latent space and visualizing them. One problem is that the generated motions sometimes include motions that are not specified. If the input motion is not sufficiently long, the motion graph will have a sparse structure and fewer candidate paths will be explored; therefore, transitions between two key postures will need to go through another posture. A simple solution is to provide a sufficient length of motion data for the input and to change the threshold for how similar postures should be grouped into a single node when constructing the motion graph.

References

1. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Trans. Graph. 21(3), 473–482 (2002). https://doi.org/10.1145/566654.566605.
2. Qin, J., Zheng, Y., Zhou, K.: Motion in-betweening via two-stage transformers. ACM Trans. Graph. 41(6) (2022). https://doi.org/10.1145/3550454.3555454.
3. Tang, X., Wang, H., Hu, B., Gong, X., Yi, R., Kou, Q., Jin, X.: Real-time controllable motion transition for characters. ACM Trans. Graph. 41(4) (2022). https://doi.org/10.1145/3528223.3530090.
4. Li, W., Chen, X., Li, P., Sorkine-Hornung, O., Chen, B.: Example-based motion synthesis via generative motion matching. ACM Trans. Graph. 42(4) (2023). https://doi.org/10.1145/3592395.

5. Starke, S., Zhao, Y., Zinno, F., Komura, T.: Neural animation layering for synthesizing martial arts movements. ACM Trans. Graph. 40(4) (2021). https://doi.org/10.1145/3450626.3459881.
6. Choi, M.G., Yang, K., Igarashi, T., Mitani, J., Lee, J.: Retrieval and Visualization of Human Motion Data via Stick Figures. Computer Graphics Forum (2012). https://doi.org/10.1111/j.1467-8659.2012.03198.x.
7. Sakamoto, Y., Kuriyama, S., Kaneko, T.: Motion Map: Image-based Retrieval and Segmentation of Motion Data. In: Boulic, R., Pai, D.K. (eds.) Symposium on Computer Animation. The Eurographics Association (2004). https://doi.org//10.2312/SCA/SCA04/259-266.
8. Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., Shen, X.: T2m-gpt: Generating human motion from textual descriptions with discrete representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) .
9. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5152–5161 (2022) .
10. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. ACM Trans. Graph. 21(3), 483–490 (2002). https://doi.org/10.1145/566654.566606, https://doi.org/10.1145/566654.566606.