A Novel Direct Manipulation Technique for Motion-editing using a Timeline-based Interface

Natapon Pantuwong[1]

ABSTRACT

Recently, motion data is becoming increasingly available for creating computer animation. Motion capture is one of the systems that can generate such motion data. However, it is not suitable to capture a lot of motion due to the cost of motion capture technique, and the difficulty of its postprocessing. This paper presents a timeline-based motion-editing system that enables users to perform motion-editing tasks easily and quickly. A motion sequence is summarized and displayed in the 3D environment as a set of editable icons. Users can edit the motion data by performing a sequence of operations on a single key frame or over an interval. The recorded sequence is then propagated automatically to a set of target key frames or intervals, which can be either user defined or system defined. In addition, we provide a simple interaction method for manipulating the duration of specific intervals in the motion data. Methods for combining and synchronizing two different motions are also provided in this system. In contrast with the previous work that allows only temporal editing, the proposed system provides editing functions for both geometry and temporal editing. We describe a user study that demonstrated the efficiency of the proposed system.

Keywords: Motion Editing, Timeline-based Interface, Direct Manipulation

1. INTRODUCTION

Motion capture data is commonly used to generate a movement of 3D character for animation system. The benefit of motion capture is that it provide realistic and high quality of computer animation, because the movement of each joint is captured from real motion. In case of capturing motions of several behaviors, motion capture could be the promising system to generate realistic motions. However, this benefit is achieved by high cost of the motion capture system, and several complex postprocessing. Therefore, capturing motion repeatedly to produce variations of the style of motion for single behavior is not a suitable method.

A great amount of previous researches aims at providing the motion manipulation tools for users. Such tools can be used to edit the previously captured motion to the desired style of motion. Almost of the previous methods describe the motion data as a mathematical model, and then users can edit the motion data by varying the parameter values of such mathematical model. Typically, this strategy is often computationally expensive. Although this strategy can provide an intuitive method to edit captured motion data, it requires a large set of example motions to generate the mathematical models. It may be difficult for users to obtain such large set of example motions. For systems with only a few examples, manual editing technique is needed. Moreover, manual editing should be suitable for novice users, because any complex calculation is not required by this editing strategy.

For manual motion editing, users can edit the motion directly at the specific key frame of the motion data. Typically, a timeline-based interface is often used in many computer animation applications for both motion visualization and motion editing purpose. This interface displays a set of main actions (key poses) of the motion at specific times (key frames). Users edit the motion by manipulating several key poses, with the software interpolating between key poses to generate the overall result. Although this strategy is straightforward, it is not suitable for editing of captured motion data because a set of key frames is not provided by the motion captured system. Therefore, users may need to edit the motion at every key frame, which leads to the high completion-time problem. Furthermore, if the motion to be edited is periodic or cyclic motion, or users need to create a cyclic motion from an arbitrary motion, users may need to perform similar editing operations repeatedly that is an inefficient repetition, which may be too difficult for novice users. Recently, this problem has been solved in [17]. The editing technique called "Editing Propagation" has been proposed to reduce the difficulty and completion-time for editing of cyclic motion. This technique allows users to select a single editing interval, and then, after users complete the editing task for that interval, similar

Manuscript received on June 22, 2018 ; revised on March 9, 2019.

Final manuscript received on March 10, 2019.

[1] The author is with Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520, Phone(+6699)1939149, E-mail:pantuwongn@gmail.com

editing operations are propagated to all similar intervals in the motion data automatically. Although this research can reduce the completion-time of motion editing task, it provides only temporal editing of motion data, i.e., users cannot directly edit the geometry information on each key pose visualized on the timeline.

We propose a novel motion-editing technique using a timeline-based interface. The main purpose of this research is to reduce the editing completion-time of typical key-frame editing technique for both geometry(pose) and temporal editing of motion data. Unlike previous methods [17], which discuss only temporal editing of motion data, the proposed system also enables editing of geometry or pose information about the motion. For the captured motion data, since it has no information about key frames, a set of key poses must be firstly extracted from the captured motion data. We use the method described in [25] to extract a set of key poses, and locate each key pose on the timeline. Unlike typical timeline-based systems that visualize the timeline and the set of key poses separately, our system displays each pose as an editable pose icon directly on the timeline. This strategy allows users to observe and edit the motion simultaneously. We display our timeline via a 3D interface to avoid ambiguous projections of poses, because users can freely rotate the viewing angle around the timeline to understand the behavior of the motion being edited. In addition, users can interact with each key pose in 3D environment to edit such key pose directly. We developed several editing techniques in four editing categories, namely pose editing, duration editing, pose transfer, and duration synchronization. Each type of editing involves only a few operations. The basic idea of editing propagation in [17] is extended to pose editing. Apart from this editing technique, we also propose editing from record, and copy-and-paste interaction techniques for this system. For editing from record, users can record several of key poses by editing the pose of the standard skeleton, and then users can locate each key pose on the timeline to generate the resulting motion. The copy-and-paste interaction technique is provided to transfer geometry and temporal information from source motion to target motion. By using such proposed techniques, the motion editing tasks can be completed with small number of interaction steps.

The remainder of this paper is organized as follows. Related works are discussed in Section 2. Overview of the proposed system is introduced in Section 3. Each editing technique is explained in Section 4. Experimental results are shown and discuss in Section 5 and Section 6 respectively. Finally, we conclude the paper in Section 7.

2. RELATED WORK

Motion editing is very important because it can reduce the cost of motion creating system, such as motion capture. A number of researchers developed several editing techniques that can be used to edit the stylistic of motion data, so users can obtain new motion with the desired style easily with low cost. This section discusses about several techniques to edit captured motion data to new motion.

One possible technique is called style transfer [3,13,21]. This technique allows users to transfer the style information from one motion to the second motion. This technique tries to extract the style from the source motion, and then adjust the extracted style to the target motion. A shortcoming of this approach is that there is no way to modified the style of source motion. That is, if the expected styled are not provided in the set of example motion, users cannot edit the target motion to their desired style.

Parameterization is another method to edit the motion data. In [14], several physical parameters for human model are extracted from an example motion performed in a particular style. Then, the extracted parameters can be used to synthesize new motions with the style that is controlled by the extracted physical parameters. In [5,9,15,16], the algorithms to analyzed and described the motion data in term of mathematical model is introduced. This technique required a set of example motions performed in a wide variety of motion styles for particular actions. Then, the algorithm creates the motion model with two main parameters, identity and style. Users can edit the style of motion by adjust the value of each parameter until the desired motion is obtain. Since the style parameter can be extracted by using this algorithm, style transfer is also possible by using this method. Again, the parameterization strategy requires a set of example motion, it is impossible to generate a resulting motion, if the example of motion is not provided. Especially, the motion model technique require a large set of examples, which may be too difficult for user to get that set. Although, adjusting the parameter values is not a difficult task, users may need to perform this task repeatedly until the desired motion is obtain, which can leads to the tedious problem of users.

Several methods also use example of motions to generate new motion. [20] directly blend motions of varying styles to generate new motion. Motion graph based algorithms [1,8,10] calculate the suitable transition point between two motions to concatenate both motions together to generate new motion. Although these techniques can generate new motion data, the stylistic information cannot be modified by using these approaches.

It is possible to synthesize new motion from scratch without an example of motion data. This can be achieved by using knowledge about movement gathered from the arts literature [6] or personal experience [19] to develop an algorithm to control the motion. The shortage of these algorithms is that the final quality of the motion may not be as good as example based approaches explained above. [18] solves this problem by using the motion data as an input instead of knowledge about movement. This method extracts several parameters from the input motion, and then users can edit the style of that motion through the parameters. The shortage of this method is that it is suitable for editing the style of motion rather than generating new motion, i.e., the resulting motion still represents the same action as the input motion.

Directly editing motion data is another way to complete both editing style of motion and generating new motion. The advantage of this strategy is that users can edit the motion data directly without a large set of example motions. Furthermore, complex calculations are not required by using this technique. Therefore, this approach should suitable for novice users to edit motion data. This approach can be performed in both frequency domain and spatial domain. In the former approach, the frequency information is extracted from the motion, then users can edit the motion by modifying the information about the extracted frequency information. [4] divides input motion into frequency bands that can be individually edited. [24] allows any pose in a motion sequence to be varied and use low frequency offset curves to blend this edit into the motion sequence.

To obtain the resulting motion by editing the information in frequency domain may too difficult to understand for novice users. Furthermore, such kind of users may design the resulting motion in the spatial domain. Therefore, directly editing motion data in spatial domain should be more suitable for novice users. In [22], simple mouse interaction is used to provide an editing function for users to edit the velocity of a path animation. For pose editing, keyframe editing technique is a method typically used in several computer animation softwares. By using this technique, motion data needs to be represented by a set of key poses. Each of them are located at each key frame of the motion. Users can edit motion data by performing editing operations at each key frame, and then the resulting motion is obtained by interpolating between key frames. However, this technique is not suitable for captured motion data, because such data has no information about key frames. Therefore, users need to perform editing operation for every single frame of the motion. This problem is discussed and solved in [17]. This method extracts key pose by using the method introduced in [2] to summarize the motion data in a set of key poses, and visualize each key pose in the 2D timeline based interface. The editing function called editing propagation is proposed by this method. This function allows users to select a single editing interval, and then perform several editing operations about the temporal information on the selected interval. Then, the system propagates the performed editing operations to all similar interval in the motion data. This system enables users to directly edit captured motion data, and reduce the difficulty as well as completion-time of editing task. However, the problem this system allows users to edit only the temporal information. For pose or geometry information of each key frame, typical key-frame editing technique is required. Moreover, using 2D interface as [2] could not provide intuitive interface for users because it is difficult for them to understand and manipulate a pose of each frame.

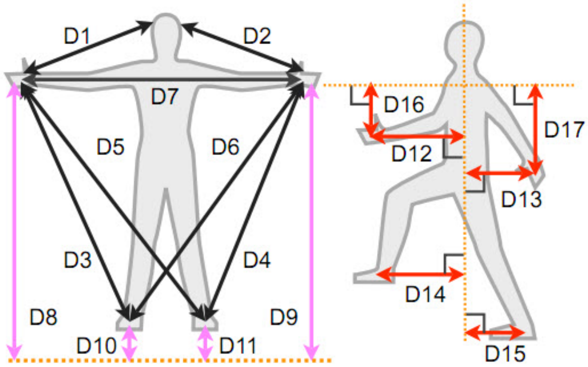

Fig.1: *Distances used to encode the pose of each frame of the input motion [?]*

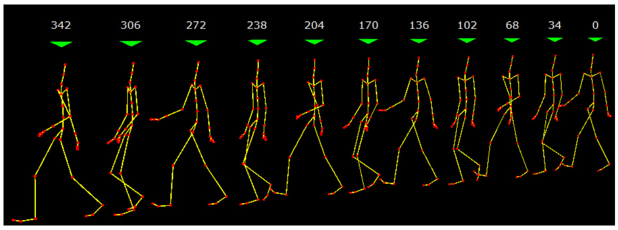

Fig.2: *Timeline-based interface of the proposed system is visualized in 3D environment. Users can select each key pose by selecting the icon above that key pose. The number displayed above each icon is the frame number of that key pose.*

We would like to develop a motion editing technique that is easy for novice user to use, and does not require a complex calculation as well as a large set of examples. Therefore, directly editing technique is used in our system. In contrast with [17], the proposed system provides several editing functions for both geometry and temporal editing. Editing propagation is extended for pose editing. Moreover, we proposed editing from record, and copy-and-paste interaction techniques, which are useful editing functions for users to edit input motion to new motion.

3. SYSTEM OVERVIEW

The overview of the proposed motion editing system is described in this section, which briefly explain the interface design and all proposed motion editing techniques.

Starting from a motion data to be edited, the first step is to extract the set of key frames from this motion. In our implementation, we use the method in [25] to extract a set of key frames from the input motion. This method encodes each frame the motion data into a set of feature values including the relative positions of each joint of the motion data, and the distances to the floor from some joint. Figure 1 illustrates these feature values for a single frame of motion. Then, this method extract the frames which have distinct changes of the values as key frames.

After the key frames have been extracted from the motion data, we can visualize them on the timline-based interface. Typically, timeline-based interface is used to visualize the motion data. By observing the changing between each key frame on the timeline, users can understand the whole motion easily. [25] also proposed the idea to visualize motion data on the 2D environment. Each extracted key pose is analyzed, and then the suitable viewing direction is selected for visualization of that key pose. In [17], this problem is also addressed. Both researches try to find the best viewing direction that can explain the variation of the movement of the input motion data. Although users can understand the context of motion data, which is visualized by using this strategy, users may not correctly understand the position of each joint of each key frame because of ambiguous projection of each key pose. To avoid such problem, we visualize the timeline in 3D environment as shown in Figure 2. In that figure, a walking motion is visualized. Above each key pose of the timeline, the selectable icon is displayed to allow users to select the pose to be edited. The number shown above each icon means the frame number of each key pose. This visualization enables users to understand the position of each joint of each key pose easily, so it should be suitable for pose editing, which requires editing in spatial domain. As for temporal editing, 3D visualization should also be suitable due to the benefit of timeline-based interface that conveys the temporal information.

After the input motion is visualize, users can perform the editing task for both pose and temporal editing. We provide four editing categories in the proposed system, including of pose editing, temporal editing, pose transfer, and duration synchronization. The detail of each category is summarized as below.

- For pose editing, editing propagation is extended to geometry editing. Users select a key pose from the timline, and then all key poses that have similar pose are selected automatically. Then, users can edit the pose of the selected key pose, which the editing operations are propagated to all selected key poses. This strategy can reduce the difficulty and completion-time of pose editing, especially when users need to edit a cyclic or periodic motion. We also proposed editing from record, which can support users to create a cyclic motion from an arbitrary motion. It allows users to record a set of key poses, and then users can locate each recorded key pose on the timline. Each key pose can be located at several different intervals on the timeline.
- Editing propagation is also implemented for temporal editing by using the basic idea of editing propagation in [17]. Users can modify the temporal information of the selected intervals on the timeline for all segments or only specific segments of character skeleton.
- Pose transfer is developed by using copy-and-paste interaction technique. This function allows users to transfer the geometry or pose information of specific part of the source motion to the same part of the target motion. This function can help users to create new motion easily, when an example of the motion that contains the desire motion is provided.
- Copy-and-paste interaction technique is also proposed for duration synchronization. This editing function is provided for synchronizing the duration of the selected interval of the target motion with the selected interval of the source motion.

4. PROPOSED MOTION EDITING TECHNIQUES

In this section, user interaction and all required algorithm for implementation of each proposed motion editing technique is described. The user interactions are also demonstrated in the accompanying video

4.1 Pose Editing

4.1.1 Editing Propagation

Editing of periodic motion, such as walking, or running, by using typical key-frame editing is very time-consuming. Moreover, it is a tedious task for users, because users have to perform similar editing operations repeatedly for every period of such motion. Because of the inefficient repetition, it may be too difficult for novice users to perform the editing task.

Editing propagation method in [17] could be used to solve this problem. Because this method is firstly developed for temporal editing, we extend it to pose editing process. This method propagates the set of editing operations performed for one key pose to all similar key poses. Figure 3 illustrates this function. A user chooses one key pose on the timeline, and then all similar key poses are then selected automatically by using the features previously used to extract all key poses. Users can modified the automatically selected key poses, if desired. The user then edits the

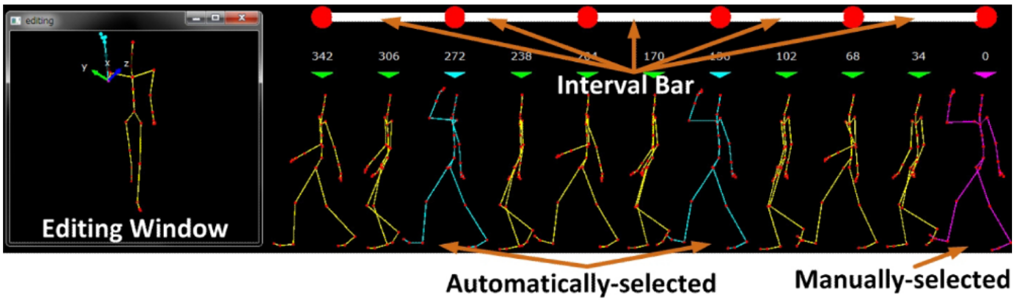

Fig.3: *Example of editing propagation. A user selects and edits a single key pose. Similar editing operations are then propagated to all similar key poses, which can be selected automatically by the system.*

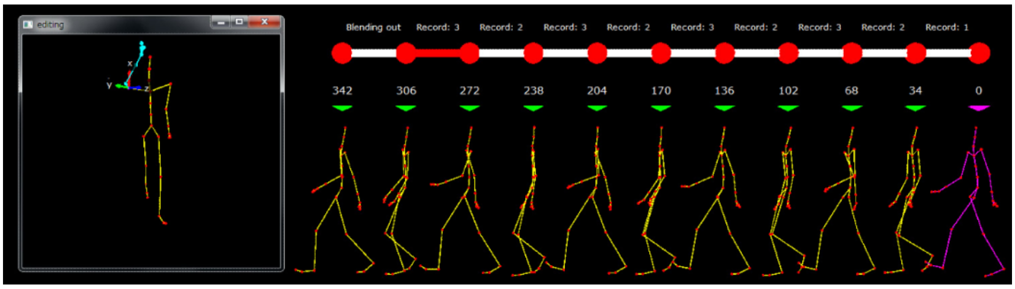

Fig.4: *Example of editing from record. A user creates and records all required key poses. Then, each recorded key pose can be located on the timeline to generate new motion.*

chosen key pose via the editing window. These editing operations will then be propagated to the selected set of key poses automatically. The user can set the blending interval for each edited pose by adjusting the interval bars. The overall motion is then obtained by cubic-spline interpolation between editing durations.

After users select one key pose, all similar key poses are extracted by determining the similarity between sets of encoded features. We use 17 encoded features values used to extracted all key poses during the timeline creation process for determine the similarity between key poses. Assume that the set of encoded feature of the user-selected key pose is k. We have to determine the pose similarity between key pose ,k, and another key poses, k', on the timeline. The pose similarity, s, is calculated by using correlation coefficient as shown in the following equation.

$$s(k,k') = \frac{\sum_{i=1}^{17}(k_i - \overline{k})(k'_i - \overline{k'})}{\sqrt{\sum_{i=1}^{17}(k_i - \overline{k})^2 \sum_{i=1}^{17}(k'_i - \overline{k'})^2}} \quad (1)$$

where k_i is the value of i^{th} encoded feature of the user-selected key pose, and k'_i is the value of similar encoded feature of the key pose being determined (k_i). If s has positive value, and greater than the predefined threshold, the key pose being determined, k', is selected to be involved with the editing task. In our implementation, we would like to select only the key pose that closely similar to the user-selected key pose, so we set the threshold of 0.7, finding it provides a correct result as shown in Figure 3. From that figure, the pose on the rightmost key pose is selected by user, and then two other key poses are selected automatically by using the method as described.

After all similar key poses are selected automatically, users can modified that selected set of key poses by themselves. Then, users can edit the pose of all selected key poses via the editing window. At the editing window, an editable icon of pose is provided. Users can select any joint in that pose, and then edit the rotation angle of that joint. Although this editing function allows users to edit the motion freely, we would like to ensure the correctness of the resulting motion. Therefore, we employs rotation constraints to each joint to ensure that the impossible pose is not created. Position constraints, such as feet must be not penetrate the floor, are also applied. The value of editing is then propagated to all selected key poses, and used to edit the rotation angle of that joint of each key pose.

Finally, the resulting motion is obtained by interpolating between key poses. We also provided blending interval icons, which are called interval bars, that allow users to adjust the interval for interpolation process. For each interval bar, users can adjust its starting and ending position (red points of each interval bar in Figure 3) on the timeline to guide the system about the starting and ending point of the interpolation. All poses of the key poses under both endpoints of each interval bar are used in the interpolation. We use cubic spline interpolation to create the result edited motion. We calculate the interpolation

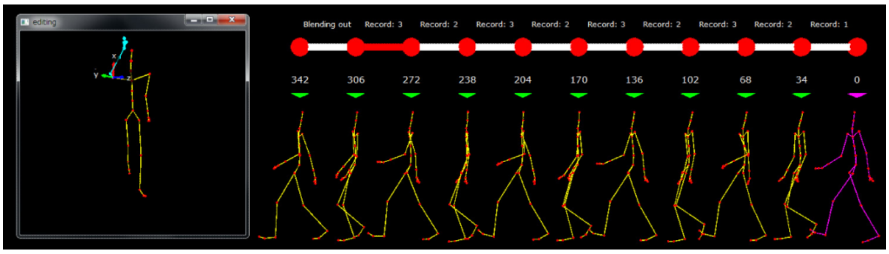

Fig.5: *Example of duration editing. A user selects the editing interval and moves it to a desired position on the timeline.*

result only for the edited joints, and keep the original movement for other joints. After the interpolation process is completed, the resulting edited motion is obtained.

4.1.2 Editing From Record

This editing function is developed to reduce the difficulty of creating a periodic motion from an arbitrary motion. This kind of editing is also a difficult task, if the typical key-frame editing technique is used. For example, if a user needed to create a walking-plus-hand-waving motion from a normal walking motion, the user would need to perform similar editing operations (creating the hand waving) in several key poses. The editing from record is illustrated in Figure 4. Here, users can record three key poses of the hand-waving motion (lift right arm, swing right, and swing left) in the editing window. Then, users can locate these records within specific intervals by adjusting the interval bar. The same record could be used for several different intervals. Finally, the system would interpolate all editing durations to generate the overall motion.

A standard editable pose icon is displayed in the editing window, which user can manipulate the pose of such icon to their desire key pose. Similar transformation constraints as mentioned previously are applied to such pose icon to ensure the correctness of the resulting motion. After users complete each key pose, rotation angle of each edited joint is stored.

Interval bar is used to enable users to locate each created key pose on the timeline. Information about each record pose are shown on each interval bar, which each record pose can be located at several intervals as shown in Figure 4. Both endpoints, which describe the points for interpolation, of each interval bar also adjustable. In this case, we also perform the interpolation for only the joints that have been edited. For each interval bar, the values used for interpolation are the rotation angle of all edited joints of the pose in previous interval. For example, values used for interpolation of the 34^{th} frame in Figure 4 are the value from the first recorded posed, and such values for the 68^{th} frame. In case of the first interval (the rightmost one in Figure 4), since there is no information about the previous interval, the values of the pose at the starting point of that interval are used for the interpolation. The cubic spline interpolation is also used to generate the resulting motion.

4.2 Temporal Editing

Temporal editing is a function that is provided to edit the speed of all joints and some specific joints of the motion data in the selected intervals. We developed this function based on idea of editing propagation proposed in [17], but we slightly change some calculation to reduce the computation time.

Figure 5 illustrates the temporal editing. A user first specifies an editing interval and creates a duration bar whose endpoints can be moved along the timeline. The user then move each endpoint of the duration bar to the desired position on the timeline. Dynamic time warping algorithm is then used to generate the overall result. The duration editing operation can also be propagated to any similar intervals, which can be detected automatically by determining the pose similarity as described before.

[17] applies data-indexing technique called match web to the motion data to improve the runtime performance of the similar-interval searching. In stead of using that technique, we use the pose similarity calculated by using correlation coefficient in Equation 1, since this calculation is not time-consuming, and precalculation of indexing is not required. Algorithm 1 describes the method the search for all similar intervals. Assume that there are N_p key poses on the timeline, and there are N_s key poses in the user-selected interval. Let $\{p_{s1}, p_{s2}\}$ be the user-selected interval, starting from the p_{s1}^{th} frame to the p_{s2}^{th}. The similar interval should contain similar pose at the starting and ending point. Moreover, it should contain all of the key poses in the user-selected interval in correct order. Therefore, the candidate interval must have the number of key poses greater or equal to the number of key poses in the user-selected interval, which lead us to stop the searching at the $(N_p - N_s)^{th}$ frame. In this algorithm, first, we try to search for the key poses that similar to the starting and ending key poses of the user-selected interval(p_i and p_j in Algorithm 1). Then, we determine the similarity of the poses in user-selected interval and the interval being determined. If all poses in the user-selected interval are found in cor-

rect order, the interval being determined is selected. The next searching iteration will then start at the end point of the newly-selected interval.

After all similar intervals are selected, users can adjust the both endpoints of the duration bar to the desired interval as shown in Figure 5. Users can decide the retain the number of all frames of the motion, or change it. In the former case, we have to change the speed of both selected and unselected intervals. For example, in Figure 5, users need to reduce the duration of the selected interval, i.e, the speed of this interval will be increased. Therefore, we have to increase the duration (decrease the speed) of the intervals before and after the user-selected interval to retain the overall number of frames of the resulting motion. If users allow the number of frames to be changed, we can edit the speed only for the selected intervals, and retain the information for all unselected intervals. In our implementation, speed information is edited by dynamic time warping algorithm described in [13].

Algorithm 1 searchSimilarIntervals

while i $\leq N_p - N_s$ **do**
if p_i is not in the selected interval && $s(p_i, p_{s1}) \geq$ *threshold* **then**
for j = i+1 to N_p **do**
if p_j is not in the selected interval && $s(p_j, p_{s2}) \geq$ *threshold* **then**
Calculate pose similarity for all key poses in $\{p_{s1}, p_{s2}\}$ and $\{p_i, p_j\}$
if All key poses in $\{p_{s1}, p_{s2}\}$ are found in $\{p_i, p_j\}$ in correct order **then**
Determine $\{p_i, p_j\}$ as similar interval
i = j
end if
end if
end for
else
i = i + 1
end if
end while

In addition to editing the speed of all segments, temporal editing can be also applied to a specific part of the body. For example, users may need to increase or decrease the speed of arm-swing motion. Our system allows users to achieve this task by editing the speed of motion of the specific part in term of the number of cycles. First, users select the interval to be edited. Then, we search for the periodic characteristic in the selected interval by using the method described in [23]. If that characteristic is found, we adjust the user-selected interval to the detected period of movement. Otherwise, we determine that the user-selected interval contain one cycle of movement. Then, users can increase or decrease the number of cycles of movement. We do not allow users to set the number of cycles less than 1 to ensure the continuity of the motion after the selected interval. After users decide the number of cycles, selected segments of all key poses in the selected interval will then be rearranged along the timeline to give the desired number of cycles. The overall motion is then generated by interpolating only the selected part of all key poses in the selected interval.

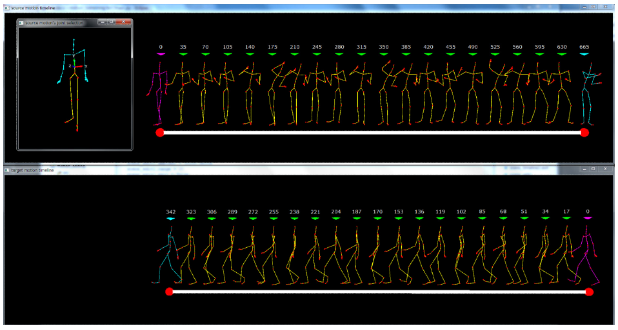

Fig.6: *Example of pose transfer. A user selects a specific segments of the punching motion, and also select an interval to be copied (the window above). Then, the user select the editing interval of the walking motion (the window below). Finally, the selected motion data of the punching motion is transferred to the walking motion to generate walk-plus-punch motion.*

4.3 Pose Transfer

Pose transfer is developed by using copy-and-pasted interaction. This function enables users to create a new motion by transferring pose information of the motion data for a selected segment of the source motion to a target motion. For example, walk-plus-punch motions could be created by adding punching-motion data for arm segments to a walking motion. Figure 6 illustrated this example. In the above window, punching motion is visualized, which users can select the specific part of the motion and specific interval. In this case, the upper part of body is selected, because we need to transfer the movement of arms and body segments. Motion data of the selected part in the selected interval is then copied, and transferred to the similar part of the editing interval selected in the walking motion (the window below). We uniformly locate copied motion data inside the editing interval to preserve the temporal information. Finally, cubic spline interpolation is used to generate final result. In case that the sizes of the selected parts of the source and target motions differed significantly, a motion-adaptation algorithm [11] would be required to generate satisfactory motion.

The editing propagation technique is also applied to this editing function. For the target motion, all similar intervals are selected automatically by using the Algorithm 1. Then, the copied motion is trans-

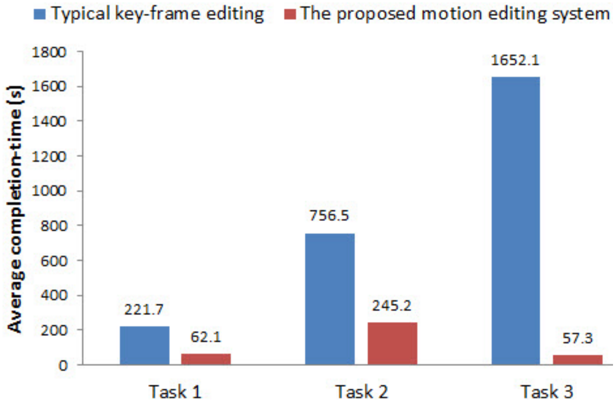

Fig.7: *Comparison of the average completion-time of the three tasks performed in the evaluation.*

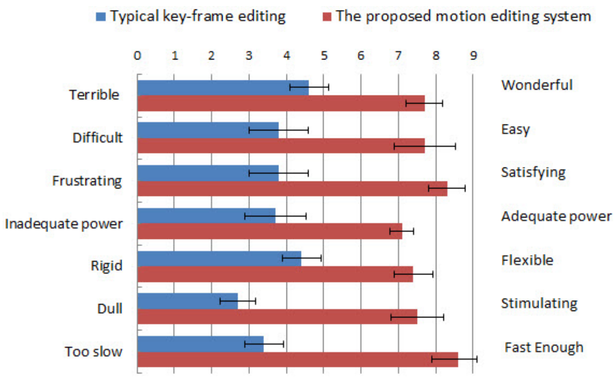

Fig.8: *Results for the QUIS questions used in the evaluation. Error bars represent the standard deviation (SD) for each question.*

ferred to all selected intervals to generate a resulting motion.

4.4 Time-duration Synchronization

The copy-and-paste interaction is also used to developed time-duration synchronization function. This editing function synchronizes the duration of selected interval in the target motion with the selected interval in the source motion. The interface of this function is similar to the pose transfer function. First, users select an interval in the source motion. Then, users select the interval to be synchronized with the selected interval of the source motion. Time warping algorithm described in [12] is then used to synchronize the target motion interval with that of the source motion. Again, all similar intervals of the target motion can be selected automatically by using Algorithm 1 to be synchronized with the time-duration of the selected interval of the source motion.

5. EXPERIMENTAL RESULT

We conducted an experiment to evaluate the proposed system. The purpose of this experiment is to measure the performance and ease-of-use of the proposed algorithm. Several measures could be considered in this evaluation, such as editing-completion time or the correctness of propagated key poses or interval extractions. In this paper, we focus on completion time and user feedback.

This experiment was conducted in a laboratory. Since the purpose is to identify the ease-of-use, all participants were selected from a group of people who lacks of computer as well as computer graphics skill. Seven male and three female computer users participated in the evaluation. They were 25-30 years of age, and were unfamiliar with computer animation and motion editing. They were asked to complete three motion-editing tasks for both geometry and temporal editing tasks as listed below.

1. Participants were to manipulate the maximum angle of the right shoulder when it swung forwards of the normal walking motion. There were three key frames that showed such a pose.
2. Participants were to create a walking-plus-hand-waving motion from normal walking motion. Four cycles of hand-waving motion were created in this task.
3. Participants were to increase the number of right-arm swing cycles from two to four.

Before started the evaluation, since all participant were unfamiliar with the computer graphics and motion editing system, we briefly explain how to use the implemented system to edit the motion data by using the typical key-frame editing technique, and all proposed editing techniques. After the explanation, we allowed participants to practice using the implemented system for 30 minutes. We did not compare our proposed techniques with the method in [17] because the direct geometry editing technique has not been discussed in this paper.

We divided the participants into two groups. Initially, one group used our implemented system to perform all tasks by using a typical key-frame editing technique, typically used in animation software such as Blender, while the other group used the proposed editing functions provided by our implemented system to complete all three tasks. After completing all tasks, each group repeated the tasks with the other editing technique. For the group that used the proposed editing system, since there were several editing functions implemented in the system, participants had to choose the editing function to complete the tasks by themselves. Each participant were asked to perform each task three times for each editing system.

We measured the completion times during the evaluation, and compared the average completion times for each task between using the typical key-frame editing technique and using the proposed editing functions. Figure 7 shows this comparison. Using the typical key-frame editing technique, the average completion times were 221.7 s (SD = 19:68), 756.5 s (SD = 63:85), and 1652.1 s (SD = 61:93) for the first, second, and third tasks, respectively. Using the proposed editing functions, times were reduced to 62.1 s (SD = 5:20), 245.2 s (SD = 16:70), and 57.3 s (SD = 16:55), respectively. A t-test indicated a significant difference between the two methods ($p < .001$).

We also asked the participant to complete the questionnaire to give us feed back. The participant questionnaire rated the two editing systems using seven questions from QUIS [7] which is focusing on intuitiveness of computer softwares. Each question had a rating scale of zero(poor) to nine(excellent). Figure 8 shows the average score on each question. Because ordinal scales were used, we used the Wilcoxon signed-rank test to analyze the QUIS, which indicated a significant difference between the two methods (p <: 005). Participants also noted that they preferred to use our proposed editing functions because they were quicker and significantly easier than using the typical key-frame editing technique.

We also asked the participants to compare between 2D and 3D visualization of the timeline. We visualized ten different motions, such as running, jumping, and pitching, in both 2D and 3D environment, and asked participants to select their prefer method for motion visualization and motion editing. For 2D visualization, viewing direction selection in [17] was used. From the questionnaire, all participants answered that there was no significantly different between both visualization techniques. They could understand the context of each motion easily by using both 2D and 3D visualization. However, 3D visualization was preferred by the participants because they could rotate the viewing angle freely to choose the correct key pose to be edited.

6. DISCUSSION

In this section, we discuss the result from the evaluation for both benefit and limitation issues. We also discuss about possible further improvement of the proposed editing functions.

6.1 Benefit

According to the evaluation result, the proposed editing functions can significantly reduce the completion time for both geometry and temporal editing tasks. From our observation during the evaluation, we found that the participants felt difficult during perform each task by using the typical key-frame editing technique. The reason was that the participants needed to design for both editing key poses and all other key poses, whereas they thought only the editing key poses by using our proposed editing function. For example, in the first tasks, the participants had to manipulate three poses that its right shoulder swung forward, and they also had to manipulate the poses between such poses to ensure the continuity of the resulting motion. This difficulty is solved by using the proposed editing propagation for pose editing that allows the participants to edit at single key pose, which is also automatically propagated to all required key posed to be edited, and users can easily adjust the interval for interpolation. One participant would like to stop the editing task (for task 3) during the evaluation, because this task was too difficult and too tedious, when the typical key-frame editing technique was used, whereas this task could be complete with the small number of interaction by using the temporal editing function of our system. The feed back from the participants confirms that our proposed editing functions can reduce the difficulty from the typical key-frame editing technique.

6.2 Limitation

Although our proposed system got a good feed back from the evaluation, the participants also mentioned several problems of our system. All participants claimed that although temporal editing, pose transfer, and time-duration synchronization were easy to use, pose editing function was difficult because they needed to perform the pose manipulating directly on the pose icon. They said that it is still difficult to edit the pose by manipulating the rotation angle because they had no knowledge about computer graphics. At first, they thought that some editing operations, such as swing the arm-segment to the left to create hand-waving, was very easy, but it was difficult in practice because they needed to select the correct rotation axis before performing the editing task. Furthermore, they felt confuse when they needed to adjust the interval for interpolation, because they could not decided where is the best position for locating each endpoint of interval bar.

Another comment is that it may be difficult to ask nonprofessional users to use the propose system immediately after practicing with a very short time. For example, to create walking-with-hand-waving, the participants could understand that they needed to record three key poses, but they felt difficult when they had to complete the recording and allocating all key poses on the timeline.

These two problems are related to pose editing because users have to editing the motion in low-level. Therefore, knowledge and experience about computer graphics are still requires. This problem is not found in other editing functions, because the developed system does not require the users to edit directly on the key pose as the pose editing.

6.3 Possible Improvement

According to the limitations discussed in previous section, the reason of such problems is that users need to edit the motion in low-level, which requires knowledge about computer graphics. Therefore, the strategy to improve the proposed system is to provide more simple interface that users can use to edit the motion in higher-level of motion data.

First, the difficulty of key-pose editing via rotation angle of each joint should be solved. We could reduce the difficulty of key-pose editing task by editing joint position instead of rotation angle. Although using rotation angle is straight forward method, since rotation angle is used to described motion, users have to edit several joints before the desired pose is obtained. Simple interface, such as simple direction-pad icon for all x,y, and z-axis, should be as interface to edit the position of the selected joint of the key pose being edited. This interaction is not so low-level as editing of rotation angle of each joint. For example, users could select the hand of motion data, and push the up-icon to lift the hand up. Since only the position of the hand is edited by user, the rotation angle of each joint that should be affected by that position need to be calculated. The inverse-kinematic solver is required to achieve this purpose.

Another interface could be used instead of interval bar to reduce the difficulty of adjusting interval for interpolation. For example, in case of editing from record, users may supply the number of cycles to the system, and then the system allocates each recorded key-pose automatically. Another example is that the system may provide an option that users can supply the starting and ending number (or specific time) of key frame on the timeline to identify the interval for interpolation.

Although we do not choose the parameterization strategy to develop the motion editing system because it requires a large set of example motions to extract the parameters, some parameters, such as speed or magnitude of arm-swing angle, could be extracted from the motion without such set of examples, and complex calculations. This strategy could also reduce the difficulty of motion editing task, because it is not a low-level editing, and knowledge about computer graphics is not required.

7. CONCLUSION

This paper presents a novel timeline-based motion-editing system. It is well suited to editing both geometry and temporal motion data. The proposed system is developed to reduce the difficulty of direct key-frame motion editing. Several editing functions in four editing categories, including pose editing, temporal editing, pose transfer, and time-duration synchronization, are proposed in this paper. In a user study, significant differences were observed between the proposed system and a typical key-frame editing system with respect to task completion time and user ratings. The limitations and issues about possible improvement are also discussed in this paper. For the future work, we plan to evaluate the system further and to investigate additional editing functions that will enhance our system flexibility. We also plan to improve the proposed system according to the comments received from the evaluation.

References

[1] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Motion synthesis from annotations," in *ACM SIGGRAPH 2003 Papers, SIGGRAPH '03*, ACM, New York, USA, pp. 402-408, 2003.

[2] J. Assa, Y. Caspi, and D. Cohen-Or, "Action synopsis: pose selection and illustration," *ACM Trans. Graph.*, vol. 24 pp.667-676, 2005.

[3] M. Brand and A. Hertzmann, "Style machines," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, ACM Press/Addison-Wesley Publishing Co., New York, USA, pp. 183-192, 2000.

[4] A. Bruderlin and L. Williams, "Motion signal processing," in *Proceedings of the 22nd annual conference on Computer graphics and interactivetechniques, SIGGRAPH '95*, ACM, New York, USA, pp. 97-104, 1995.

[5] Y.-L. Chen, J. Min, and J. Chai, "Flexible registration of human motion data with parameterized motion models," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games, I3D '09*, ACM, New York, NY, USA, pp. 183-190, 2009.

[6] D. Chi, M. Costa, L. Zhao, and N. Badler, "The emote model for effort and shape," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH'00*, ACM Press/Addison-Wesley Publishing Co., New York, USA, pp. 173-182, 2000.

[7] J. P. Chin, V. A. Diehl, and K. L. Norman, "Development of an instrument measuring user satisfaction of the human-computer interface," in *Proc. of CHI'88*, 1988.

[8] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen, "Snap-together motion: assembling run-time animations," in *Proceedings of the 2003 symposium on Interactive 3D graphics, I3D '03*, ACM, New York, USA, pp.181-188, 2003.

[9] Z. He, X. Liang, J. Wang, and Y. Yue, "Flexible editing of style, identity and content of human motion," in *Symposium on Interactive 3D Graphics and Games, I3D '11*, ACM, New York, USA, pp.202-202, 2011.

[10] R. Heck and M. Gleicher, "Parametric motion graphs," in *Proceedings of the 2007 symposium on Interactive 3D graphics and games, I3D'07*, ACM, New York, USA, pp.129-136, 2007.

[11] E. S. L. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adapta-

tion," *ACM Transactions on Graphics*, vol.29, issue 4, no.33, pp.1-8, 2010.

[12] E. Hsu, M. da Silva, and J. Popović, "Guided time warping for motion editing," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '07*, Eurographics Association, Aire-la-Ville, Switzerland, pp.45-52, 2007.

[13] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," *ACM Trans. Graph.*, vol.24, pp.1082-1089, 2005.

[14] C. K. Liu, A. Hertzmann, and Z. Popović, "Learning physics-based motion style with nonlinear inverse optimization," in *ACM SIGGRAPH 2005 Papers, SIGGRAPH '05*, ACM, NewYork, USA, pp. 1071-1081, 2005.

[15] J. Min, Y.-L. Chen, and J. Chai, "Interactive generation of human animation with deformable motion models," *ACM Transactions on Graphics*, vol. 29, issue 1, pp.9:1-9:12, 2009.

[16] J. Min, H. Liu, and J. Chai, "Synthesis and editing of personalized stylistic human motion," in *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games, I3D '10*, ACM, New York, USA, pp. 39-46, 2010.

[17] T. Mukai and S. Kuriyama, "Pose-timeline for propagating motion edits," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, (SCA '09)*, ACM, New York, USA, pp.113-122, 2009.

[18] M. Neff and Y. Kim, "Interactive editing of motion style using drives and correlations," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation ,SCA '09*, ACM, New York, USA, pp. 103-112, 2009.

[19] K. Perlin, "Real time responsive animation with personality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1 issue 1, pp. 5-15, 1995.

[20] C. Rose, M. Cohen, and B. Bodenheimer, "Verbs and adverbs: multidimensional motion interpolation," *IEEE Computer Graphics and Applications*, vol. 18, issue 5, pp. 32-40, 1998.

[21] A. Shapiro, Y. Cao, and P. Faloutsos, "Style components," in *GI '06 Proceedings of Graphics Interface 2006*, Canadian Information Processing Society, Toronto, Canada, pp.33-39, 2006.

[22] S. S. Snibbe, "A direct manipulation interface for 3d computer animation," *Comput. Graph. Forum*, vol. 14, issue 3, pp. 271-284, 1995.

[23] K.-T. Tang, H. Leung, T. Komura, and H. P. H. Shum, "Finding repetitive patterns in 3d human motion captured data," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication, ICUIMC'08*, pp. 396-403, New York, USA, 2008.

[24] A. Witkin and Z. Popovic, "Motion warping," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95*, pp. 105-108, New York, USA, 1995.

[25] H. Yasuda, R. Kaihara, S. Saito, and M. Nakajima, "Motion belts," in *Proc. of mage Electronics and Visual Computing Workshop*, 2007.

Natapon Pantuwong is a researcher in a field of human computer interaction, pattern recognition, and machine learning. He got a Ph.D. in Electrical Engineering and Information Systems from the University of Tokyo, Japan in 2012. He received his B.Eng from department of Information Technology Engineering and M.Sc. from faculty of Information Technology, KMITL, Thailand , in 2003 and 2006 respectively