Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009), pp. 1–11
E. Grinspun and J. Hodgins (Editors)

Pose-Timeline for Propagating Motion Edits

Tomohiko Mukai and Shigeru Kuriyama

Toyohashi University of Technology

Abstract

Motion editing often requires repetitive operations for modifying similar action units to give a similar effect or impression. This paper proposes a system for efficiently and flexibly editing the sequence of iterative actions by a few intuitive operations. Our system visualizes a motion sequence on a summary timeline with editable pose-icons, and drag-and-drop operations on the timeline enable intuitive controls of temporal properties of the motion such as timing, duration, and coordination. This graphical interface is also suited to transfer kinematical and temporal features between two motions through simple interactions with a quick preview of the resulting poses. Our method also integrates the concept of edit propagation by which the manual modification of one action unit is automatically transferred to the other units that are robustly detected by similarity search technique. We demonstrate the efficiency of our pose-timeline interface with a propagation mechanism for the timing adjustment of mutual actions and for motion synchronization with a music sequence.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-dimensional Graphics and Realism—Animation. Computer Graphics [I.3.6]: Methodology and Techniques—Interaction techniques.

1. Introduction

A long sequence of human motion data often includes similar movements that repeatedly appear in a regular or near-regular cycle. With dance motion, for example, similar-looking arm swings, turning, and jumping movements repeatedly appear in the same motion sequence. Similar action units should be edited to give similar impressions; for example, magnification of single arm-swing movement should be iteratively applied to other similar arm-swings. Signal-based techniques are suited to detect the unit movements of uniform cyclic property; the actual human motions, however, rarely show such uniform periodicity; for example, every step of a captured gait motion has a slightly different duration and timing. Such semi-cyclic properties force animators to manually edit each segmented action separately, which requires many inefficient repetitive operations.

Most animation systems, such as Maya and 3ds Max, provide a spline-based curve editor of joint rotation and the functions of inverse kinematics. Although they are suited for local editing of fine motions, their interaction scheme is not suited to the repetitive editing of semi-cyclic motion data. Such repetitive editing operations often require the many edited segments to be displayed in a compact representation in order to preview the effects as a whole. A motion summarization technique, which adaptively arranges the constituent poses on a 2D image with irregular sampling, is very suited to compactly display long motion sequences [ACCO05, YKSN07]. The summary on a timeline is particularly appropriate for visualizing both the essential poses and their appearance time. This property is well suited for browsing the motion sequence without animation playback. Furthermore, the integration of the summary timeline and editing interface helps users to intuitively modify the motion data.

This paper proposes a new system for editing the style and timing of motion data on a summary timeline. Our goal is to develop a system that enables rapid editing of cyclic or semi-cyclic motion data. The motion editing can be operated with the intuitive interface, which displays the motion sequence as a pose-timeline constructed by adaptively extracting keyframes of the motion data. The duration and timing of the motion are controlled by merely dragging the icons displayed on the pose-timeline. The temporal features, such as duration, timing, and coordination among body parts, can be transferred from the reference motion to the source ones using our graphical interface with drag-and-drop operations. The pose-timeline interface can be extended to transfer the kinematical features, which are controlled through the mean and variance of joint rotations, and the trajectory of hands and feet. Furthermore, each editing operation of a short segment is automatically applied to the multiple segments of similar movements based on the concept of *edit propagation* [LLW04, PL07]. The target segments of the edit propagation can be flexibly selected among the candidates extracted using the similarity search technique [KG04]. Consequently, our pose-timeline interface can provide an easy-to-use system for both local and global editing of the cyclic and semi-cyclic motions.

Our main goal is the overall integration of existing techniques, such as motion editing, motion retrieval, and motion summarization technique, into our novel pose-timeline interface. Most technical components of our system are built on previous work, and they have been slightly adjusted and seamlessly woven together to provide an intuitive timeline-based interface. We demonstrate how the integrated interface provides rapid editing of complex motion sequences. The major contributions of our system are as follows:

Timeline-based editing interface Our pose-timeline is composed by analyzing the kinematical motion features for allowing intuitive manipulations with 2D icons. Drag-and-drop controls of selected periods are suitable for our propagation function while efficiently displaying simultaneously both the editing target segment and the result of the deformation. This interface can supply intuitive cues for synchronizing with the sounds, movements, and events of relevant objects.

Propagation of motion edits The editing operation applied to a single segment is automatically propagated to multiple segments of similar appearance. We customize the existing motion retrieval technique for rapid propagation and introduce some possible editing operations suited to this propagation function.

The major limitation of our system is that the representation of the timeline is unsuited to supply an interface for controlling the position and orientation of the root node. As a result, the walking motions often cause undesirable foot-skating when lower body movements are edited through the manipulation of our timeline. This defect, however, could be compensated by some foot-skate removal techniques such as inverse kinematics.

1.1. Overview

Figure 1 shows the interface of our prototype system. The motion sequence is displayed by a timeline on which the pose at picked frame (blue) and the sequence of pose-icons (green) are respectively presented. The blue region indicates the time interval of the editing segment specified by the user. The green region indicates the motion segment which is similar in appearance to the blue region, where this similar region is detected automatically using the similarity search technique. The editing operation applied to the blue segment is propagated to the green segments. For example, dragging the end of the time interval changes the duration of the blue segment, and the change is simultaneously propagated to the duration of the green segments according to its scaling factor. The appearance timing of a pose is also controlled by dragging the corresponding pose-icon while maintaining the overall duration. Moreover, we can control the motion timing of several joints by selecting body parts (colored purple) with painting operations.

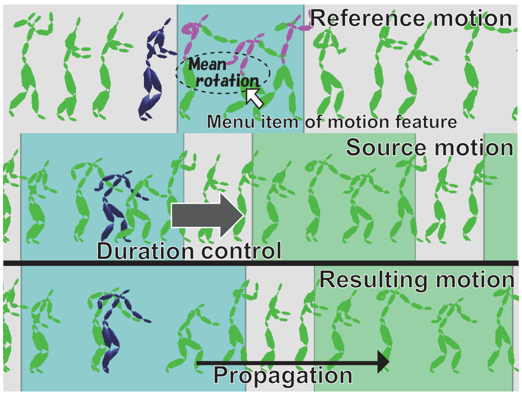

Figure 1: *Snapshot of our prototype system. The blue segment is temporally expanded by dragging the posterior end, and this control is automatically propagated to the green segments similar in appearance to the blue segment. Moreover, the motion feature of painted reference motion can be transferred to the source motion by dropping the menu-item onto it.*

Our system provides the drag-and-drop interface to transfer kinematical features from the reference motion to similar motion segments, where the features include the duration and timing of motion, the temporal coordination among multiple joints, mean and variance of joint rotations, and trajectory of hands and feet. Our current implementation displays the names of kinematical features on menu items, and dropping the name onto the timeline results in the transfer of the corresponding feature to the multiple segments by the propagation mechanism.

2. Related Work

2.1. Motion Editing Interface

Sketch-based techniques and other intuitive interfaces are used for rapid prototyping of character animation. The sketch-based motion synthesis technique translates a pen gesture into an animation sequence using collection of motion data [TBvdP04]. The performance-based key-framing [TM07] generates the interpolation curve of key-frame animation according to the shape and drawing speed of the stroke sketched on the screen. Although they provide an intuitive interface for motion prototyping, they are not suitable for detailed editing of captured motion data. On the other hand, the layering technique is proposed for rapid motion synthesis and editing [DYP03]. The concept of layering interface is similar to our selective feature transfer of motion style. Our interface design is also similar to that of [Sni95] which can directly edit the velocity of a path animation with simple mouse interaction. Some motion assembling methods also use a timeline-based interface to edit the duration and timing of motion data [YKSN07,Osh08,KHKL09]. Our system further enhances the efficiency of a timeline-based interface with drag-and-drop operations of editable pose-icons.

2.2. Motion Retiming

Motion retiming techniques have been used to control the timing and duration of the motion data. Motion blending techniques often use timing control for temporally aligning multiple motion samples [BW95, RBC98, KG03]. The physics-based method [MPS06, LHP06] controls the speed of the motion clips so that the edited motion matches given dynamic conditions. The dynamic time-warping (DTW) algorithm is used for splicing the movement of upper and lower body [HKG06] or arm and hand [MZF06]. Our system extends the DTW algorithm for transferring the property of temporal coordination. Our method is related to guided time-warping [HdSP07], which controls the speed of whole-body movement using discrete optimization so that the timing property matches the reference motion. This method provides the automatic editing of all subsequent appearances of similar actions in motion sequence, but it only deals with the retiming of whole-body motion. In contrast, our method can edit the temporal coordination by modifying the movement velocity of several body parts. Moreover, the combination with the motion retrieval technique enables the flexible selection of the target segment of edit propagation.

2.3. Motion Style Editing

Motion style is edited by several techniques. Signal-based filtering methods are introduced to control the style of cyclic motions by modulating the spectrum amplitude of motion curves [UAT95]. Although these methods are well suited to changing the style of cyclic motions, they do not consider the local editing of the non-cyclic motion sequence. The system identification method [HPP05] and Gaussian process model [IAF09] are used for estimating the transformation function of motion styles between two motions. These methods require a large amount of time to learn the transform function. Principal and independent component analysis decomposes the motion data into a set of style components [UGB*04, SCF06]. However, we think that style components should be represented with explicit variables, such as rotational difference, for intuitive operation. Motion displacement mapping and time-warping are used to directly edit the emotional impression of the captured motion [WP95, WMZ*06]. Our system uses such explicit variables of motion style to transfer the kinematic and temporal properties between two motion segments.

2.4. Edit Propagation

The concept of edit propagation is introduced by the conventional methods of image and appearance editing. It assumes that a local editing of visual data, such as image, shape, and animation, can be spatially and temporally propagated to other regions of similar appearance. The optimization-based colorization [LLW04] translates a gray-scale image into a colored image with a few painting constraints by assuming that the neighboring pixels indicating similar luminance should be painted with a similar color. On the other hand, the interactive BRDF editing method [PL07] constructs a nearest neighbor graph to propagate the manual editing of the measured BRDF to the entire region of similar appearance, including the regions spatially-distinct from the edited pixels. Our method also constructs a precomputed neighbor graph to propagate the user edits on the short segments to temporally-distinct multiple segments. Cardle et al. introduced a replicated editing technique for motion capture data [CVB*03]. This method propagates an edit operation to multiple similar actions which are detected by the motion retrieval technique. Our method provides a more effective interface suited to replicated editing.

3. Interface Design

The user interface of our system is designed to seamlessly integrate the editing interface and preview function into the two-dimensional timeline with standard drag-and-drop function. The user specifies the time interval of the editing target segment on the timeline, and the segments of similar movement are automatically detected and highlighted. The edits applied to the target segment are propagated to the detected segments. Our pose-timeline enables easy definition of the time intervals due to the pose-icons which represent the appearance time of action units. The pose-icon is also used as the editing interface for motion retiming and the selection of the body parts to be edited.

3.1. Pose-timeline

The motion sequence is displayed on a two-dimensional timeline with several pose-icons for two purposes. First, the content of the motion sequence is summarized in the timeline by which the essential features of the motion are comprehensibly understood. An automated method is therefore necessary to extract constituent postures from the motion sequence. Second, the timeline is used as the editing interface for the timing control and selection of the joints to be edited. Thus, the number and the projection directions of pose-icons should be flexibly controlled at will while maintaining interactive responses.

We use the simplified version of action synopsis

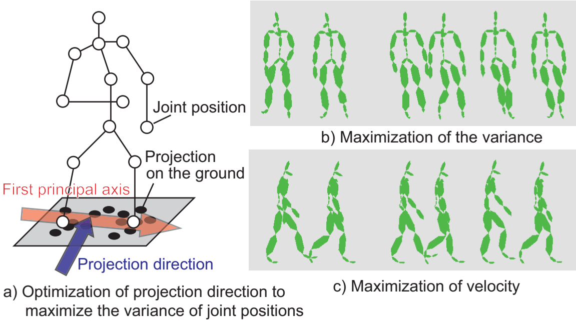

Figure 2: *Optimization of projection direction for the pose-icons of gait motion. a) Overview of the optimization for maximizing the variance of joint positions. b) Variance maximization enables the selection and control of joints of the editing target. c) Pose-icons of the velocity maximization well represent the moving direction of the character, and enable quick preview of the editing result.*

[ACCO05] for extracting key-poses to be displayed as the pose-icons. The 3D positions of all joints, which do not consider the 2D horizontal position and orientation about the vertical axis of the root node, are computed at every frame to compose the timeseries of the position vectors of all joints. The timeseries is then mapped into the low-dimensional space using principal component analysis (PCA). All poses are ranked in the low-dimensional space using the pose selection algorithm proposed in [ACCO05]. Given the number of pose-icons at run time, higher-ranking poses are displayed on the timeline. Although our method neglects the velocities and joint angles for selecting key-poses, the difference of extracted poses between these two methods is negligible for practical purposes.

The key-poses are projected onto the timeline by optimizing their orientation. Our system improves the previous method [YKSN07] for providing two types of projection: one maximizes the variance of joint positions of the pose-icon, and the other maximizes the moving velocity of the character on the timeline. The viewpoint selection technique [ACOYL08] proposed a similar criterion to optimize the viewpoint by which the user can easily understand the content of motion. We slightly modified the criterion for enhancing the usability of the interactive editing such as painting operation. To simplify the optimization problem, the vertical orientation of the character is fixed so as to be along the upward direction, so the projection direction is optimized with respect to the rotation angle about the vertical axis. The variance of joint positions is maximized by analyzing the distribution of all joint positions. Figure 2 (a) provides an overview of the projection direction optimization. First, the 3D positions of all joints at t-th frame are projected onto the 2D ground, which are represented as the point set $\left\{(p_{x,j}, p_{y,j})\right\}, j \in J$ where J is the set of all joints and end-effectors. Second, the principal axis of the point set is computed by the least square method: $\theta_{axis}(t) = \tan^{-1}\left\{\mathbf{p}_y \cdot (\mathbf{p}_x/(\mathbf{p}_x \cdot \mathbf{p}_x))\right\}$, where $\theta_{axis}(t)$ denotes the rotation angle between the principal axis and x-axis at t-th frame, and the vectors $\mathbf{p}_x$ and $\mathbf{p}_y$ are $\mathbf{p}_x = \{p_{x,j}\}$ and $\mathbf{p}_y = \{p_{y,j}\}$. Finally, the projection direction is determined so as to be at right angle to the principal axis. The frame-by-frame analysis, however, often causes jerky change of the projection direction. The rotation angle of the principal axis θ_{axis} is therefore smoothed using the moving average filter. The projected velocity of the whole body is also maximized by analyzing the distribution of the velocity vectors of all joints in a similar way.

Figure 2 (b) and (c) demonstrate the timeline of gait motion of two steps with six pose-icons. The key-pose extraction almost exactly detects the characteristic pose of the gait motion with feet contacting the ground. Figure 2 (b) shows the result of the maximization of the variance of joint positions. The key-poses are projected from the front to show the skeletal figure at a maximum area. This projection enables the easy selection and editing of the joint position. Figure 2 (c) represents the moving direction of the gait motion by which we can intuitively understand the content of the motion. This property is well suited to the interactive preview of the editing result.

3.2. Motion Retiming by Dragging Pose-Icons

We use time-warping (described in Section 5.1 to 5.3) to adjust the duration and timing of the motion segment; the interface for this time-warping consists of dragging and painting of pose-icons, which we outline here before giving details. The motion duration is locally changed by dragging the posterior end of the target segment as shown in Figure 1. The appearance timing of a pose is controlled by shifting backward or forward the corresponding pose-icon. The temporal coordination among multiple joints is edited using the painting tool. By painting several body parts of the pose-icon, the timing control affects only the rotation of painted joints while maintaining the movement of the unpainted ones. Notice that if only one end-effector is painted, the connected limb motion is time-warped in order to maintain the spatial trajectory of the end-effector.

3.3. Drag-and-Drop Style Feature Transfer

The style features of the motion segments are transferred by dragging-and-dropping the menu-item of extracted features that pops up from the reference segment. The set of menu-items includes the temporal features, such as duration and temporal coordination, and kinematical features, such as mean and variance of joint rotation and trajectory of end-effectors (described in Section 5.4. The menu-items adaptively appear according to the painting state of the pose-icons. For example, we obtain the items of mean rotation,

Table 1: *The relation between painting state of pose-icons and pop-up menu-items of motion features.*

Painting state	Types of pop-up feature menu
no painting	temporal alignment
several joints	temporal alignment mean rotation, rotation variance T.C. among selected joints and the fixed ones
end-effector	temporal alignment, effector trajectory T.C. among selected limb and the fixed parts
end-effector & several joints	T.C. among selected limb and joints & all other menu-items

T.C. : temporal coordination

rotation variance, and temporal coordination between joint sets when several joints are painted. The relation between the painting state and the menu-item is summarized in Table 1. The influence interval of the feature transfer can be arbitrarily selected by the dropping location of the menu-item. The single motion segment is locally edited by dropping the menu-item on it. All selected segments are similarly edited when the menu-item is dropped onto an unselected region of the timeline. The experimental results of the feature transfer are summarized in Figure 3.

4. Propagation of Motion Edits

Our edit propagation scheme consists of two steps: detection of motion segments whose poses are similar to those within a given period, and the propagation of an editing operation to the detected segments. The former step, similarity search, is common to any type of editing operation. In contrast, the method of propagation is specific to the type of editing operation. The propagation methods for motion retiming and kinematical feature transfer are explained in Section 5.

4.1. Operation and Propagation Window

Given the time interval of the editing target segment, other segments of similar movement are automatically detected from the entire sequence. We use the motion retrieval method with precomputed database for the similarity search. Once the user edits the single motion segment called *operation window*, the same editing operation is simultaneously applied to the extracted segments called *propagation windows*. We here respectively denote the intervals of the operation window and p-th propagation window by $[\tilde{t}, \tilde{t}+\tilde{T}]$ and $[t_p, t_p+T_p], p \in \{1, \cdots, P\}$, where $\tilde{T}$ and T_p denote the interval length, and P represents the number of propagation windows.

4.2. Precomputed Search Database

Many types of motion retrieval algorithm have been proposed for searching the motion segments similar to the

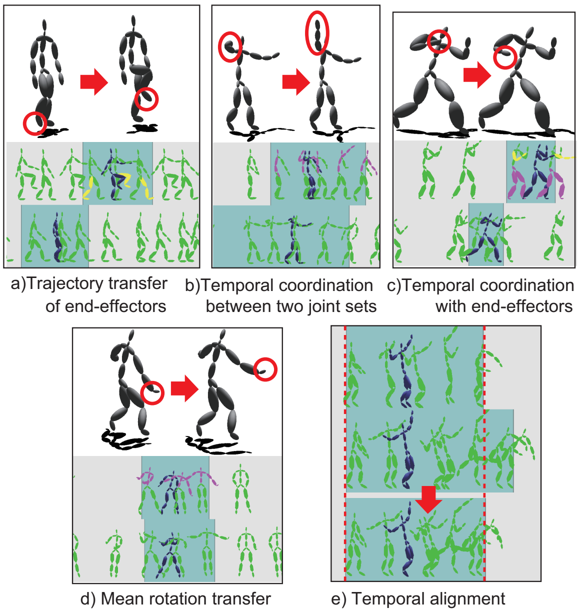

Figure 3: *Style feature transfer from the reference motion. a) The trajectory of the right foot is transferred from the marching motion to the normal walk. b) The temporal coordination transfer of right-arm swing (colored purple) of pitching motion from male to female motion data, which accelerates the swing timing. c) The temporal coordination between lower body (colored purple) and right hand (colored yellow) is transferred between two characters, which changes the punching speed. d) The mean rotation transfer makes the right-hand tennis stroke trajectory higher. e) Temporal alignment of the tennis serve motion with the smash motion.*

query motion clip from long sequences [CCW*04, MRC05, DGL09]. These methods, however, require high computational cost at run time for evaluating the similarity between the query clip and all segments of the source sequence. Furthermore, they lack information on the temporal correspondence between the motions our motion retiming algorithm uses. We employ an efficient data-indexing technique, called match web [KG04], to improve the runtime performance of the window search and time-warping operation. The match web efficiently stores the information on temporal correspondence of all possible pairs of similar motion segments. Given the time interval of the operation window, propagation windows are efficiently detected by only looking-up the match web.

We use a two-step search to find the segments slightly different from the query motion, which uses the extracted result of first search as the new queries [KG04]. However, such a multi-step search often causes undesirable overlaps of propagation windows; the partially overlapped period of two windows is doubly time-warped, and thus the duration of the period is undesirably changed. Therefore, the overlaps are heuristically dissolved by the following three rules: 1) the segment is removed if it is overlapped with both the neighboring two windows and the non-overlapped period is shorter than 50% of its duration; 2) two segments are merged into one segment when the overlapped period is longer than 80% of the duration of the shorter window; and 3) the shorter overlap, whose length is shorter than 20% of the shorter window, is separated by the center of the overlapped period. Notice that the above thresholds are experimentally determined. Although this solution successfully removes all overlaps, the modified windows often miss a part of the query motion clips because this algorithm simply truncates the part of extracted segment within the overlapped period.

5. Editing Functions

Our system integrates existing methods for spatially and temporally editing motion sequence. Most editing functions are built on previous work, and we adjust them to the pose-timeline interface and to the drag-and-drop operations while providing the edit propagation mechanism.

5.1. Duration and Timing

The duration of motion is controlled by changing the speed of the entire sequence of all joint rotations and root position. Let $\mathbf{q}(t)$ be a pose vector at the t-th frame, which consists of all joint rotation, and position and orientation of the root node. To change the duration of a motion segment, the speed at every frame is increased or decreased using a time-warp function $W(t)$. The piecewise linear time-warp [RBC98] is a simple method to change the duration of arbitrary segments, but the linear time-warp causes discontinuous motion at the boundary between two segments. We therefore use a cubic B-spline function to smoothly change the speed around both ends of the segments as illustrated in Figure 4.

Given the interval of the operation window $[\tilde{t}, \tilde{t}+\tilde{T}]$ and its modified interval $[\tilde{t}, \tilde{t}+\tilde{T}']$, the speed variation $\triangle s(t)$ is generated to satisfy the time constraints, $\tilde{T}-\tilde{T}' = \sum_{\tau=\tilde{t}}^{\tilde{t}+\tilde{T}'} \triangle s(\tau)$. Since the original speed of frame-by-frame animation is one, the frame correspondence between source frame t and time-warped frame t' is represented as $t = W(t') = \sum_{\tau=1}^{t'}(1.0 + \triangle s(\tau))$. The resulting motion $\mathbf{q}'(t')$ is therefore composed of $\mathbf{q}'(t') = \mathbf{q}(W(t'))$. The timing of an action event is edited by controlling the duration of two adjacent segments. For instance, the appearance time of a specific pose can be moved ahead by respectively decreasing and increasing the durations of the preceding and subsequent segments while maintaining the overall duration.

Propagation The durations of all propagation windows are changed by the same scaling factor of the operation window. When the duration of the operation window is

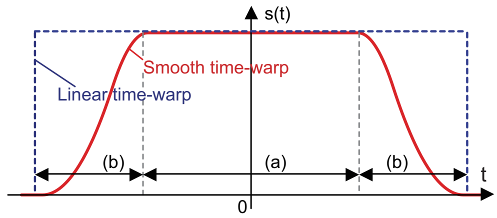

Figure 4: *Piece-wise time-warp function with cubic B-spline. (a) Uniform time-warp. (b) Ease-in and ease-out transition using cubic B-spline function.*

changed with the scaling factor α as $\tilde{T}' = \alpha\tilde{T}$, the propagation windows are simultaneously scaled as $T'_p = \alpha T_p$. These time-warps are operated simultaneously with a composite time-warp function W_C, which is synthesized by combining the time-warp functions of operation window $\tilde{W}$ and all propagation windows $\{W_p\}, p \in \{1,\cdots,P\}$ in an ascending order according to $\tilde{t}$ and t_p. For example, if $\tilde{t} < t_p$ and $t_p < t_{p+1}$ are satisfied for all p, the composite time-warp function W_C is computed by $W_C = \tilde{W}(W_1(W_2(\cdots W_P)))$. If the time-warp functions are combined in a descending or random order, some segments are redundantly modified by different time-warp functions because the time-warp of a partial segment temporally shifts the rest of the segment.

The motion retiming is also propagated from the operation window to the propagation windows. The appearance time of a key-pose is temporally shifted from $\tilde{\tau}$ to $\tilde{\tau}'$ within the interval of operation window $[\tilde{t}, \tilde{t}+\tilde{T}]$, and the durations of two adjacent segments $[\tilde{t}, \tilde{\tau}]$ and $(\tilde{\tau}, \tilde{t}+\tilde{T}]$ are respectively changed to $[\tilde{t}, \tilde{\tau}']$ and $(\tilde{\tau}', \tilde{t}+\tilde{T}]$ by the duration control. To propagate this retiming operation, the intermediate frame of p-th propagation window τ_p, which temporally corresponds to $\tilde{\tau}$, is searched by looking-up the match web (Figure 5 (a)). The propagation window is then automatically edited in a manner similar to the operation window; the intervals of two segments $[t_p, \tau_p]$ and $(\tau_p, t_p + T_p]$ are respectively changed into $[t_p, \tau'_p]$ and $(\tau'_p, t_p + T_P]$ so that the scaling factor of the former segment becomes identical to that of the operation window, and the latter is determined so as to maintain the overall duration (Figure 5 (b)). The composite time-warp function is synthesized by combining all time-warp functions in an ascending order according to $\tilde{t}$, t_p, $\tilde{\tau}$, and τ_p.

5.2. Temporal Alignment

The temporal alignment is the process by which the timing and duration of source motion are aligned with the reference motion. The automatic time alignment method is proposed based on the DTW algorithm [KG03]. Since all information of temporal correspondence between the arbitrary pair of motion segments is precomputed as the match web, the time alignment is processed at an interactive rate [KG04].

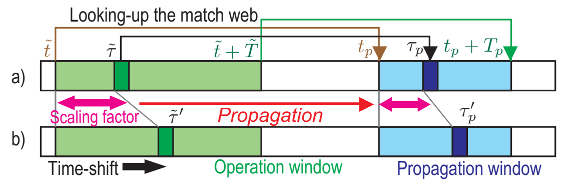

Figure 5: *Edit propagation of motion retiming. a) Temporally corresponding frames are detected by looking-up the match web. b) The manual time-shift of the intermediate frame is propagated from the operation window to the propagation windows by transferring the scaling factor of the preceding segment while maintaining the overall durations.*

Propagation Each propagation window is temporally aligned with the operation window using the DTW-based time alignment by looking-up the match web.

5.3. Temporal Coordination

Human movement involves the systematic coordination of multiple joints. Temporal coordination is especially important to dominate the impression and intention of the motion, which is represented by the phase relation of motion curve among different body parts. For example, the pitching motion indicates a temporal coordination among lower body movement, rotation of trunk, and right arm swing, and such systematic coordination often gives the impression of a pitching style. The phase relation is easily edited by temporally shifting the motion curves of several joints while maintaining the overall duration, which can be achieved by applying the motion retiming only for those joints. However, manual operations often lose the naturalness of complex movements because the phase shift of a motion curve affects the coordination with multiple body parts. Therefore, our system provides an example-based editing of the temporal coordination by inheriting the phase relation between two motion curves from the reference motion. We introduce a simple algorithm that temporally shifts the motion curve of one joint (free joint) while fixing that of the other joint (fixed joint).

Coordination between two joints The simplest temporal coordination is defined between two joints as illustrated in Figure 6 (a). The transferring algorithm consists of three time alignments. First, the source motion is temporally aligned with the reference motion on the basis of the motion curve of the fixed joint (Figure 6 (b)). The motion curve of the free joint of the source motion is then aligned with that of the reference (Figure 6 (c)). Finally, the time-warp of the first step is canceled using its inverse function (Figure 6 (d)).

Coordination between joint sets The above approach is extended for the temporal coordination between two joint sets. The motion curves of either joint set are temporally shifted while fixing those of the other set. The three-step time alignment is similarly used by computing the time correspondence among multidimensional motion curves of the joint set [HKG06].

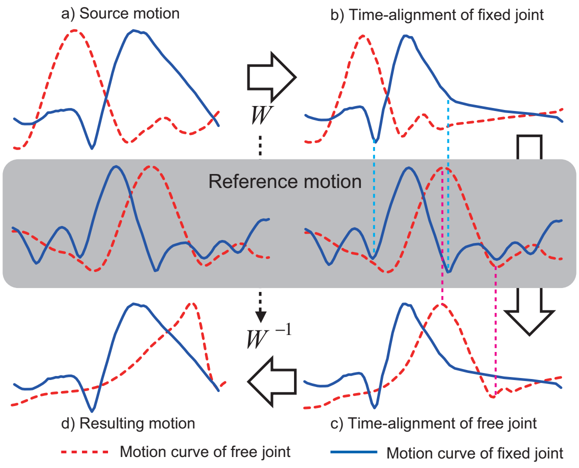

Figure 6: *Three-step process of temporal coordination transfer. The source motion has a phase relation by which the motion curve of fixed joint reaches the peak later than the free joint, and the reference motion has the opposite relation. The temporal coordination of the source motion can be aligned with the reference by temporally shifting forward the motion curve of the free joint.*

Coordination with end-effectors The movements of hands and feet often give most of the motion style rather than the joint rotation. Our system provides the tool for editing the temporal coordination between joint set and the end-effectors. This tool is also an extension of the basic transferring algorithm with three-step time alignment. First, the source motion is temporally aligned with the reference motion based on the movements of fixed joint sets [HKG06]. Second, the moving velocity of the end-effector is aligned with the reference motion while maintaining the overall shape of the spatial trajectory. Third, the timeseries of the end-effector position is time-warped using the inverse time-warp function of the first step. Finally, the limb posture at every frame is modified to satisfy the new trajectory using the limb inverse kinematics (IK) technique [TGB00]. The whole-body IK techniques may produce a more natural result, and we will incorporate such an advanced IK solver in the future. Notice that the temporal coordination between end-effectors can be edited in a similar way.

Propagation The transferred temporal coordination is simultaneously applied to the propagation windows. This is achieved in a straightforward manner by repeating the transferring operation from the reference segment to the operation and propagation windows.

5.4. Kinematical Features

Our system involves the existing techniques of motion style editing based on signal processing. The filtering method [WMZ*06] modifies the mean and variance of joint rotations, where the rotation variance can be regarded as the intensity of the joint movement. We extend this method with the time domain filter [LS02] for improving the computational accuracy. The edited joint rotation is blended with the original motion using the ease-in and ease-out interpolation so that the joint rotations around the time boundaries of the editing segment are smoothly connected to the neighboring frames.

The spatial trajectories of hands and feet are transferred between motion segments. First, the trajectory of an end-effector of the reference motion segment is temporally aligned with the source motion as described in Section 5.2. Next, the trajectory of the reference motion is spatially aligned with the source motion using the coordination alignment method [KG03]. Third, the trajectory of aligned reference motion and source one are blended using the ease-in and ease-out interpolation. Finally, the limb posture is modified to satisfy the new trajectory at every frame using the limb IK technique.

Propagation These kinematical features are simultaneously applied to the propagation windows by simply repeating the transferring operation from the reference segment to the operation and propagation windows.

6. Applications

6.1. Editing of Mutual Motions of Two Characters

Figure 7 demonstrates the editing of the mutual motions of two fighting characters. The characters begin avoidance behavior right after a hit by one character. The motion sequence of one character is temporally controlled to avoid all punching and kicking. To ease the collision avoidance by timing control, the pose-timeline is extended to visualize the hitting timing and location by highlighting the colliding body parts of the pose-icon with vivid color. The key-pose extraction scheme is modified to display the colliding poses in preference to non-colliding ones. After mapping all poses in the low-dimensional space by PCA (see Section 3.1), a large offset is added to the colliding pose. Consequently, the colliding poses and their neighboring frames are preferentially selected as pose-icons because the key-pose extraction algorithm detects the extreme of the motion sequence in a low-dimensional space. The timing of motion can be changed by referring to the color of the pose-icon which is interactively updated according to the body collision. The number of pose-icons within the operation window is reduced after all collisions are avoided as shown in Figure 7 (b).

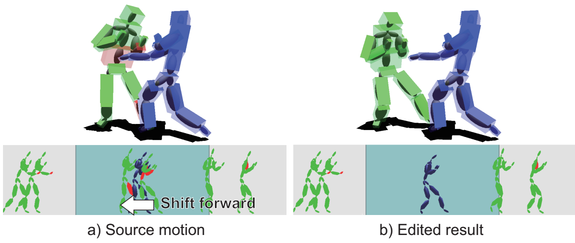

Figure 7: *Retiming of mutual fighting motions. The green-colored character avoids the hit of the blue one by the timing control so that the green character will engage in avoidance behavior earlier. The pose-icons in the operation window are removed because the timeline is reconstructed to preferentially display the colliding poses.*

6.2. Synchronization with Music Sequence

The timeline representation is well suited to compare the motion sequence with another type of sequential data. Figure 8 shows the synchronization of gait motion with a music sequence. The curved locomotion, which indicates a different duration and timing at every cycle, is edited to align the timing of foot contact with the rhythm of the music. The movements of the lower limbs are emphasized by transferring the spatial feature from the marching motion. The editing operation mainly consists of the following four steps: 1) The trajectories of feet are transferred from the marching motion to the source gait motion. 2) The operation window is specified over two gait steps, and other similar steps are automatically selected as the propagation windows. 3) All windows are temporally aligned. 4) The duration of the operation window is changed to be in rhythm with the music by referring to the wave shape of the sound signal displayed on the timeline. Most propagation windows are automatically aligned with the rhythm pattern due to the temporal alignment, but some manual adjustments are required for the irregular intervals of the music sequence because the propagated operation uniformly modifies all segments. Nevertheless, the whole editing process is usually done within a minute by the trained user.

The existing motion synchronization methods enable automatic editing of the cyclic motion by analyzing the beat component of motion curves [KPS03, LL05]. These methods can automatically control the timing of the motion cycles if the extrema of motion curves are robustly detected. However, they assume that every motion cycle has the same styles in terms of timing adjustment. The manual editing is still unavoidable because some impressive expressions must be added to the segments using intentional retiming and deformations of spatial features for each different style. Our side-by-side visualization of music signals and motions provides a direct and flexible editing interface without losing the efficiency in controlling periodical events.

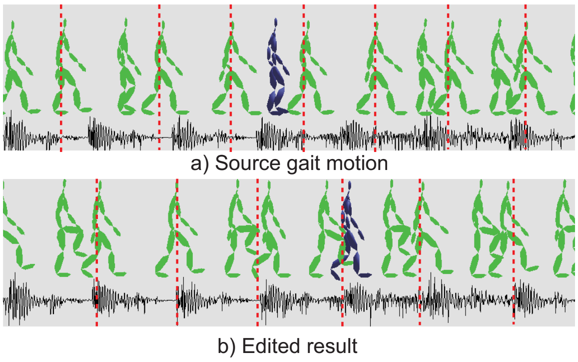

Figure 8: *Motion synchronization with music sequence. The trajectory of both feet is transferred from marching motion, and contact timing of every step is synchronized with the rhythm of the music sequence. Manual editing operation is only applied to a unit walking cycle.*

6.3. Usability Evaluation

We performed a user study to evaluate the effectiveness of our prototype system. The subjects were five students who are familiar with 3D animation systems. After a short instruction on usage, we asked each subject to edit the gait motion to synchronize with a music sequence and to change the walking style as demonstrated in Section 6.2. Three types of display and editing capability were given to all subjects; the first scenario disabled both the display of pose-icon and propagation function, the second disabled only the propagation function, and the third enabled the full functions. Table 2 shows the statistics of operation time to achieve the acceptable result. Displaying pose-icons on the timeline had little effect on a reduction of time-wasting animation playback to specify the operation window; t-test showed no significant difference in operation time (p = 0.1751 between Trial 1 and 2). In contrast, the propagation function reduced the repetitive operation (p = 0.0120 between Trial 2 and 3). Enabling both functions significantly reduced the operation time (p = 0.0017 between Trial 1 and 3), which showed displaying pose-icons reduced the time to preview the resulting animation of propagation windows.

Some subjects required that the pose-timeline should provide more information about each action. For example, the timing of ground contact should be highlighted for accurately synchronizing the gait cycle with the music sequence. The visualization function of such kinematical and dynamical features like [BZOP07] can be integrated in our pose-timeline, which has the potential to further increase the efficiency of our system.

7. Conclusions

We have proposed an easy-to-use, timeline-based motion editing interface that allows edit propagation of the timing and kinematical features of motion data. We can intuitively change the duration and timing of arbitrary segments by dragging the pose-icon displayed on the timeline. The painting and drag-and-drop interface is introduced for easy adjustment of the temporal coordination among joints or between joints and end-effectors. The editing operation on one segment is automatically propagated to the other segments of similar appearance, thereby reducing the cost of repetitive manual operations. The edit propagation can be triggered by dropping the menu-item of the extracted kinematical features onto the entire sequence or locally selected subsequence. Various interaction styles can be integrated into a common interface based on the pose-timeline, which gives essentially the same interaction framework as the existing keyframe-based animation systems. For this reason, most animators can easily guess its function because of their familiarity with such a framework.

Our new system can edit a short motion segment which is explicitly specified by the user. However, actual motion editing often requires per-frame operations; for example, the imposition of kinematical constraints at a single frame. Our future work includes the extension of our propagation function and a relevant interface to the per-frame editing scheme. The pose-timeline is created using the key-pose extraction technique based on the detection of the extrema of motion sequence. Although this strategy works well for displaying the content of the motion with only a few pose-icons, the selected key-poses are sometimes unusable to edit the motions. For instance, few pose-icons are extracted from motion segments where only slight movements appear. Our prototype system allows users to control the number of pose-icons, which requires some trials to find the optimal sequence, and more intuitive and direct controls of key-frame extraction should be developed.

Our prototype system shows only the basic design of the possible graphical interfaces that can handle fundamental functions based on our framework of propagation of motion edit. More elaborate and sophisticated designs could enhance the usability of our system, and more functions could be added in our editing framework while keeping operation styles consistent. A usability test with professional animators is also among our forthcoming projects.

Table 2: *The statistics of the user study, where the operation time is measured in seconds. Both the pose-icons and edit propagation are disabled in Trial 1. Trial 2 enabled the pose-icons, and Trial 3 enabled both functions.*

Subject	A	B	C	D	E	avg.	s.d.
Trial 1	222	227	236	306	244	247.0	34.0
Trial 2	249	249	176	185	215	214.8	34.4
Trial 3	140	145	179	145	156	153.0	15.7

Acknowledgements

The authors would like to thank Hiroshi Yasuda for detailed discussions, and anonymous reviewers for their helpful comments. This work was supported by the MEXT Grant-in-Aid for Young Scientists (B) 19700090 and Scientific Research (B) 18300068.

References

[ACCO05] Assa J., Caspi Y., Cohen-Or D.: Action synopsis: Pose selection and illustration. *ACM Transactions on Graphics 24*, 3 (2005), 667–676.

[ACOYL08] Assa J., Cohen-Or D., Yeh I.-C., Lee T.-Y.: Motion overview of human actions. *ACM Transactions on Graphics 27*, 5 (2008), Article 115.

[BW95] Bruderlin A., Williams L.: Motion signal processing. In *Proc. of SIGGRAPH 95* (1995), pp. 97–104.

[BZOP07] Bouvier-Zappa S., Ostromoukhov V., Poulin P.: Motion cues for illustration of skeletal motion capture data. In *Proc. of Non-Photorealistic Animation and Rendering 2007* (2007), pp. 133–140.

[CCW*04] Chiu C.-Y., Chao S.-P., Wu M.-Y., Yang S.-N., Lin H.-C.: Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation 15*, 3 (2004), 446–466.

[CVB*03] Cardle M., Vlachos M., Brooks S., Keogh E., Gunopulos D.: Fast motion capture matching with replicated motion editing. ACM SIGGRAPH 2003 Sketch and Applications, 2003.

[DGL09] Deng Z., Gu Q., Li Q.: Perceptually consistent example-based human motion retrieval. In *Proc. of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2009* (2009), pp. 191–198.

[DYP03] Dontcheva M., Yngve G., Popović Z.: Layered acting for character animation. *ACM Transactions on Graphics 22*, 3 (2003), 409–416.

[HdSP07] Hsu E., da Silva M., Popović J.: Guided time warping for motion editing. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007* (2007), pp. 45–52.

[HKG06] Heck R., Kovar L., Gleicher M.: Splicing upper-body actions with locomotion. *Computer Graphics Forum 17*, 3-4 (2006), 219–227.

[HPP05] Hsu E., Pulli K., Popović J.: Style translation for human motion. *ACM Transactions on Graphics 23*, 3 (2005), 1082–1089.

[IAF09] Ikemoto L., Arikan O., Forsyth D.: Generalizing motion edits with gaussian processes. *ACM Transactions on Graphics 28*, 1 (2009), Article 1.

[KG03] Kovar L., Gleicher M.: Flexible automatic motion blending with registration curves. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2003* (2003), pp. 214–224.

[KG04] Kovar L., Gleicher M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (2004), 559–568.

[KHKL09] Kim M., Hyun K., Kim J., Lee J.: Synchronized multi-character motion editing. *ACM Transactions on Graphics 28*, 3 (2009).

[KPS03] Kim T.-H., Park S. I., Shin S. Y.: Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics 22*, 3 (2003), 392–401.

[LHP06] Liu C. K., Hertzmann A., Popović Z.: Composition of complex optimal multi-character motions. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006* (2006), pp. 215–222.

[LL05] Lee H.-C., Lee I.-K.: Automatic synchronization of background music and motion in computer animation. *Computer Graphics Forum 24*, 3 (2005), 353–362.

[LLW04] Levin A., Linschinski D., Weiss Y.: Colorization using optimization. *ACM Transactions on Graphics 23*, 3 (2004), 689–694.

[LS02] Lee J., Shin S. Y.: General construction of time-domain filters for orientation data. *IEEE Transactions of Visualization and Computer Graphics 8*, 2 (2002), 119–128.

[MPS06] McCann J., Pollard N. S., Srinivasa S.: Physics-based motion retiming. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006* (2006), pp. 205–214.

[MRC05] Müller M., Röder T., Clausen M.: Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics 24*, 3 (2005), 677–685.

[MZF06] Majkowska A., Zordan V. B., Faloutsos P.: Automatic splicing for hand and body animations. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006* (2006), pp. 309–316.

[Osh08] Oshita M.: Smart motion synthesis. *Computer Graphics Forum 27*, 7 (2008), 1909–1918.

[PL07] Pellacini F., Lawrence J.: AppWand: Editing measured materials using appearance-driven optimization. *ACM Transactions on Graphics 26*, 3 (2007), Article 54.

[RBC98] Rose C., Bodenheimer B., Cohen M. F.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5 (1998), 32–40.

[SCF06] Shapiro A., Cao Y., Faloutsos P.: Style components. In *Proc. of Graphics Interface 2006* (2006), pp. 33–39.

[Sni95] Snibbe S. S.: A direct manipulation interface for 3d computer animation. *Computer Graphics Forum 14*, 3 (1995), 271–283.

[TBvdP04] Thorne M., Burke D., van de Panne M.: Motion doodles: An interface for sketching character motion. *ACM Transactions on Graphics 23*, 3 (2004), 424–431.

[TGB00] Tolani D., Goswami A., Badler N. I.: Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models 62*, 5 (2000), 353–368.

[TM07] Terra S. C. L., Metoyer R. A.: A performance-based technique for timing keyframe animations. *Graphical Models 69*, 2 (2007), 89–105.

[UAT95] Unuma M., Anjyo K., Takeuchi R.: Fourier principles for emotion-based human figure animation. In *Proc. of SIGGRAPH 95* (1995), pp. 91–96.

[UGB*04] Urtasun R., Glardon P., Boulic R., Thalmann D., Fua P.: Style-based motion synthesis. *Computer Graphics Forum 23*, 4 (2004), 799–812.

[WMZ*06] Wu X., Ma L., Zheng C., Chen Y., Huang K.-S.: Online motion style transfer. In *Proc. of International Conference on Entertainment Computing 2006* (2006), pp. 268–279.

[WP95] Witkin A., Popović Z.: Motion warping. In *Proc. of ACM SIGGRAPH 95* (1995), pp. 105–108.

[YKSN07] Yasuda H., Kaihara R., Saito S., Nakajima M.: Motion belts. ACM SIGGRAPH 2007 Sketches, 2007.