International Conference on Computer Animation and Social Agents 2006 (CASA 06), Geneva, Switzerland, July 2006.

Motion Style Transformation by Extracting and Applying Motion Features

Takuya Terasaki
Kyushu Institute of Technology
terasaki@cg.ces.kyutech.ac.jp

Masaki Oshita
Kyushu Institute of Technology
oshita@ces.kyutech.ac.jp

Abstract

In this paper, we present a method for the transformation of motion style. We extract style features from differences between two example motions. We then apply the style features to an input motion to transform it to a styled motion. For example, we extract style features that represent "tired" from a walking motion and a tired walking motion. By applying the style features to a running motion, for example, we can transform the running motion to a tired running motion. We represent style features as the difference between two motions in an abstract form. The style features can be applied to a wide range of input motions whose motion directions or global translations are different from the example motions. The style features consist of postural and temporal differences between two example motions. To represent the postural differences, we use the relative positions of primary body parts instead of joint angles. The movements of the pelvis are divided into relative and absolute positions and orientations. The movements of the end effectors are represented in local coordinates computed from motion directions. We present some experimental results and discuss the effectiveness and limitations of our proposed method.

Keywords: motion transformation, motion feature, character animation.

1. Introduction

There is a great demand for transformation of human motion data. By being able to modify motion data to match the context, existing motion data can be effectively reused. Some commercially available animation systems provide motion editing interfaces for animators to change motion data in direct ways; for example, by changing the joint angles or the position of end effectors on a keyframe. However, when animators seek to transform a motion clip, typically they want to change the style of motion. For example, they may want to make a walking motion look happy, tired or excited. It is difficult to realize these kinds of transformations of motion styles with existing animation systems since animators have to repeat small direct modifications to achieve their goal. Moreover, they are also required to understand how to change human motions in order to have them take on the new style such as happy, tired or excited. Therefore, these kinds of style transformations are very difficult and time consuming tasks even for professional animators.

In this paper, we propose a new motion transformation technique (Figure 1). We extract style features between two existing motion data and apply the obtained style features to another motion. For example, we extract style features represent "tired" from a walking motion and a tired walking motion. By applying the style features to a running motion, for example, we can transform the running motion to a tired running motion.

Motion blending [16][20][14][8] is a popular method to generate a new motion from some example motions; however, it cannot solve the problem we noted above since it simply blends joint angles of example motions and cannot handle spatial and orientational differences of example motions. For example, a strong

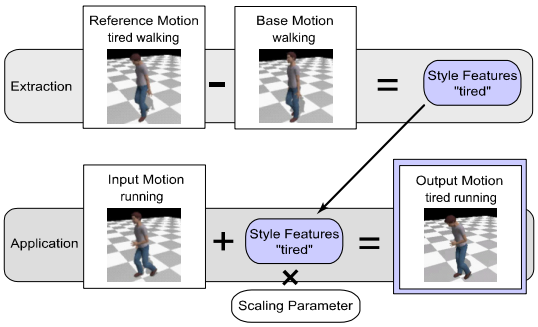

Figure 1: Overview of style transformation.

uppercut motion cannot be generated by blending a punch motion, a strong punch motion, and an uppercut motion because the directions of the hand's movements are different between the punch motions and the uppercut motion. In this case, the differences between the punch motion and the strong punch motion are mainly the hand's horizontal movements. These features cannot be applied to an uppercut motion with a motion blending method because the hand moves vertically in the uppercut motion. If we blend the strong punch motion into the uppercut motion, the hand does not move vertically but horizontally in the blended motion. Further, a happy running motion cannot be generated from a walking, happy walking and running motion. Blending the happy walking into the running motion makes the running motion into one close to a walking motion because not only the postural differences but also global translations are blended if using a motion blending method.

We solve these problems by representing style features as the differences between two motions in an abstracted form. The style features can be applied to a wide range of input motions whose motion directions or global translations are different from the example motions. The style features consist of the postural and temporal differences between two example motions. For representing postural differences, we use relative positions of primary body parts instead of joint angles. The movements of the pelvis are divided into relative and absolute positions and orientations. The movements of the end effectors are represented in local coordinates that are computed from the motion directions.

Using our system, a user can first extract style features from two example motions, e.g. a walking motion and a tired walking motion (Figure 1). The user then can generate a new motion by applying the style features to another motion, for example, a tired running motion by applying the "tired" style features to a running motion. The user also can control the degree of the application of the style features using a scaling parameter. In essence, our system generates one new motion from three example motions. By storing a number of extracted style features in a database, we expect that in the future users of our system will be able to transform any motion to a styled motion such as happy, sad or tired, etc., by selecting appropriate style features from the database and applying them.

The rest of this paper is organized as follows. Section 2 introduces related works and compares their approaches with ours. Section 3 presents an overview of style features. Sections 4 and 5 describe representations of temporal and postural features, respectively. Section 6 summarizes the extraction and application processes of style features. In Section 7, we show experimental results. Section 8 discusses the effectiveness and limitations of our method. Finally, section 9 concludes.

2. Related Work

Motion transformation has been an important animation technique and many methods have been developed. Forward and inverse kinematics are the most common basic methods. By changing joint angles or positions of body parts, animators can transform a posture on a motion clip. Signal based transformation of the angular trajectories of joints [21] is another basic method. However, these transformations are kinds of direct modifications. In order to realize a style transformation, animators have to repeat these direct modifications many times and need to know how to change human motion styles.

There are also more advanced techniques for motion transformation such as motion parameterization and motion blending, and feature extraction and application. Some researchers parameterize motions in order to control motion styles. Bruderlin et al. [4] apply multi-resolution filtering to input motions and control the set of scaling parameters for each band of the joint signals for changing the style of motions. This technique allows users to

interactively change motion styles. Neff et al. [13] use scaling parameters of joint angles or hand positions to exaggerate motion data. These methods are aimed at helping interactive motion transformation. Using these methods, animators can edit motion data through intuitive parameters. However, these methods limited the type of motions and transformations to simple ones.

Liu et al. [11] used nonlinear inverse optimization to extract control parameters as a kind of physics-based style feature from example motions and use them with optimization to synthesis new motions. Their method is suitable for motion style transformations caused by differences of muscle strength or usage.

Motion blending [16][20][14][8][12][1] has been used by many researchers. By parameterizing example motions, blend weights are computed from a given set of parameters. A new motion is then generated by blending the example motions with the blend weights. However, as explained in Section 1, motion blending is applicable only when all example motions are similar ones that have the same motion directions and global translations. Motion blending works well for interpolation but not for extrapolation. Therefore, to add a style to an input motion, we need to prepare an example motion that is very close to the input motion and which has the required style. Motion blending is useful only when an animator already has the styled motion and just wants to control the degree of the style.

Style machines [3] or motion graphs [7][10] can generate new and continuous motions by connecting small pieces of the example motions so that the generated motions preserve the style of the original ones. However, they cannot be used to change the style of other existing motions.

Unuma et al. [19] transform joint trajectories of motion data into a Fourier series and blend the coefficients of example motions to generate a new motion. Using Fourier coefficients as blending parameters, they can efficiently transform the style of the motion data. However, their method is specialized to cyclic motions since they use Fourier series expression. Moreover, the types of styles that can be realized with their method are limited.

Kawasaki et. al. [6] extracted features between golf swing motions and applied them to other golf swing motions of different persons. The purpose of their work is similar to ours. However, they used more specific features such as scaling of joint angles on keyframes, and the duration between keyframes; therefore, their method cannot be applied to a wide range of motions.

Recently Shapiro et al. [17] employed Independent Component Analysis (ICA) to extract the style components between two motions and to apply them to another motion. Hsu et al. [5] proposed a motion translation method that learns the relationship between a base motion and a reference motion using a linear time-invariant model (LTI). However, since their method simply learns the translation between two example motions, example motions basically have to cover all postures that appear in input motions. These two methods [17][5] have the advantage that input motions are not needed to be aligned to the example motions. However, since they use angular trajectories of joints, they cannot handle variations of motion directions between example and input motions. For the same reason, they cannot distinguish global and local movements of the pelvis. Since they compute the global translation of the pelvis by accumulating local translations, the global translation (e.g. feet step distance or moving speed) may be affected by local movements of the pelvis. Moreover, additional constraints on the end effectors are required to prevent foot sliding and unnatural motions.

Our work has similarity with [17][19][6] in that they too extract style features between example motions and apply them to another one. However, these methods use joint angles for style features; therefore, they cannot handle the directional changes of motions or mixtures of global and local movements. We represent style features in an abstracted form and apply them to a wide range of target motions.

3. Overview of Style Features

We extract style features between a base motion and a reference one as shown in Figure 1. We suppose that the base motion does not have any particular style and the reference motion is the same motion as the base motion but has a style. The style features are then applied to another motion utilizing a scaling

parameter to generate a styled motion. We assume all the input motions are short and of the same type of action as the base motion. If users want to transform a long and complex motion, they will need to divide them into a set of short and single actions.

To apply our method, a set of keytimes has to be assigned to all the motions in advance. We basically divide motions into three segments: initial segment, main segment, and terminal segment, and give them keytimes that specify the transition point between serial segments, as shown in Figure 2. We also give additional keytimes to the effective points in the main segment (e.g. highest point of a jump motion, contact time of a punch motion, etc.). A user can specify a number of additional keytimes as long as they match other motions. Currently, users have to manually set the keytimes; however, keytimes are usually obvious and it is easy for users to specify them. It takes only a few minutes for each motion, and users do not have to specify all the keytimes if some do not exist in the motion. Only specified keytimes and segments are used in the extraction and application processes.

As shown in Figure 3, style features SF consist of temporal and postural differences as follows,

$$SF = \begin{Bmatrix} T_1(t) \cdots T_{n-1}(t), \Delta\mathbf{P}_1 \cdots \Delta\mathbf{P}_n, \\ \Delta\mathbf{Q}_1(t) \cdots \Delta\mathbf{Q}_{n-1}(t) \end{Bmatrix} \quad (1)$$

where $T_i(t)$ is the temporal feature for the i-th segment and $\Delta\mathbf{P}_j$ is the postural feature for the j-th keyframe, and $\Delta\mathbf{Q}_i(t)$ are is the postural feature for the i-th segment. $T_i(t)$ is a time-warping function from the base motion to the reference motion for each segment. $\Delta\mathbf{P}_j$ is a 39 dimensional vector and represents postural differences of the human figure in an abstracted form. $\Delta\mathbf{Q}_i(t)$ is a time-varying function of the 39 dimensional vector and represent the trajectories of the postural differences in each segment.

In the following sections, we explain in detail temporal and postural features, and the extraction and application processes.

4. Temporal Features

Temporal features are time-warping functions for each segment of the base and reference motions,

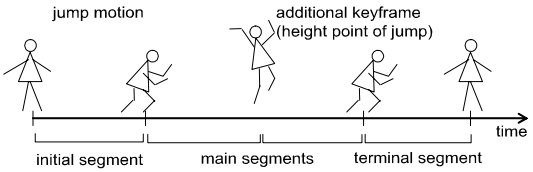

Figure 2: An example of keytimes.

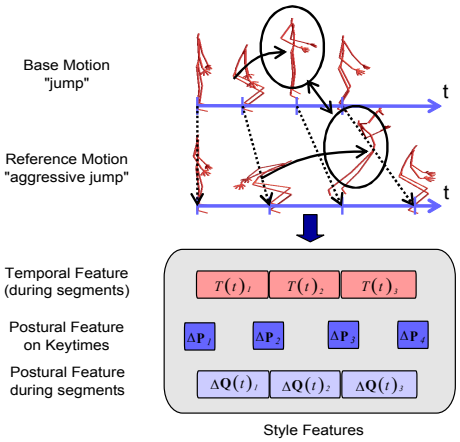

Figure 3: Style features.

$$t_r = T_i(t_b) \quad (2)$$

where t_b is the normalized time in the i-th segment of the base motion and t_r is the normalized time in the i-th segment of the reference motion. We represent a time-warping function by a series of sets of corresponding times between two motion segments.

To construct a time-warping function, we have to find two corresponding frames in the same segments of two motions. Searching for similar postures [10][8] has also been used for this purpose in previous works. However, this approach works only when the styles of the two motions are very close to each other. Therefore, instead of searching similar postures, we use a normalized postural progress that represents how far the current posture has progressed between the initial and terminal postures of the segment. We take this approach since a set of keytimes are given to two motions in our method, and so we only have to find corresponding frames in the corresponding segments of the two motions. Although some motion blending methods use the same approach and simply use linear interpolation [16][14][1] or parametric curves [12] to compute the corresponding frames, our method extracts and applies more specific time variations.

We compute the postural progress for each motion as follows,

$$S(t)=\frac{\sum_{d=1}^{tM}\sum_{j=1}^{D}\left|\mathbf{x}_j(d/M)-\mathbf{x}_j((d-1)/M)\right|}{\sum_{d=1}^{M}\sum_{j=1}^{D}\left|\mathbf{x}_j(d/M)-\mathbf{x}_j((d-1)/M)\right|} \quad (3)$$

where $\mathbf{x}_j(t)$ is the position of the *j*-th joint, *D* is the number of joints, and *M* is the number of frames on the segment. We accumulate the displacements of joint positions to determine the postural progress. $S(t)$ represents the postural progress at the normalize time *t* on the motion segment. $S(t)$ continuously increases from 0 to 1. Using the $S(t)$ of each frame in the corresponding segments of the two motions, a time-warping function for the segments is computed.

5. Postural Features

In this section, we first explain the proposed posture representation for capturing postural features in an abstract from. We also describe in detail how to represent the positions and orientations of the end effectors and the pelvis. We then describe how to extract and apply postural differences using the posture representation.

5.1. Postural Representation

We use a 39 dimensional vector $\mathbf{P}$ to represent postural information (Figure 4). Postural features $\Delta\mathbf{P}_j, \Delta\mathbf{Q}_i(t)$ in equation (1) are represented using this form by computing the differences between the corresponding postures of two motions. We basically use the positions and orientations of primary body parts in this representation.

To represent postures of each limb, we use the relative positions and orientations of the end effectors $\mathbf{e}_i, \mathbf{r}_i$ that are represented in the local coordinates of their base body part (pelvis for feet and shoulder for hands). Let's suppose that $\mathbf{e}_{\text{global},i}, \mathbf{r}_{\text{global},i}$ are the global position and orientation of the *i*-th end effector and $\mathbf{R}_i$ represents the local coordinates of the base body part (pelvis or shoulder) of the *i*-th end effector. The local position and orientation are computed by transforming global ones $\mathbf{e}_{\text{global},i}, \mathbf{r}_{\text{global},i}$ to local ones $\mathbf{e}_i, \mathbf{r}_i$ as follows,

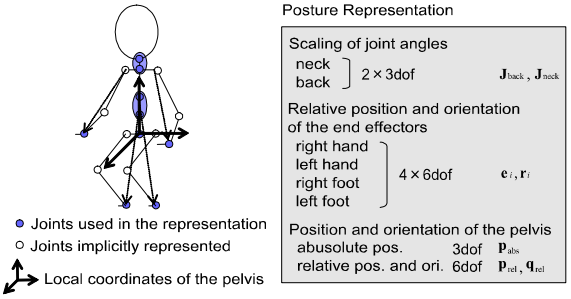

Figure 4: Posture representation.

$$\mathbf{e}_i=\mathbf{R}_i^{-1}\mathbf{e}_{\text{global},i} \quad (4)$$

$$\mathbf{r}_i=\mathbf{R}_i^{-1}\mathbf{r}_{\text{global},i} \quad (5)$$

We use quaternions to represent rotations, $\mathbf{r}_i$. To handle directional differences on the motions in the extraction and application stages, we represent the spatial and directional differences of the end effectors using local coordinates computed from the moving direction of the motion data (Section 5.3).

To isolate the absolute and relative movements of the pelvis, we compute the absolute position, relative position and relative orientation $\mathbf{p}_{\text{abs}}, \mathbf{p}_{\text{rel}}, \mathbf{q}_{\text{rel}}$, and store them independently (Section 5.4).

The rotation of the back and neck $\mathbf{J}_{\text{back}}, \mathbf{J}_{\text{neck}}$, respectively, are also quaternions and represent the average rotations of multiple joints on the back and neck.

5.2. Extraction of Postural Features

For postural features we use the postural difference on each keyframe $\Delta\mathbf{P}_j$ and the function of the postural difference for each motion segment $\Delta\mathbf{Q}_i(t)$.

For each segment, postural features are applied to the motion as follows

$$\mathbf{P}_i(t)' = \mathbf{P}_i(t) + \left((1-t)\Delta\mathbf{P}_i + t\Delta\mathbf{P}_{i+1} + \Delta\mathbf{Q}_i(t)\right)s \quad (6)$$

where *t* is the normalized time on the *i*-th segment, s is the scaling parameter, and $\mathbf{P}_i(t), \mathbf{P}_i(t)'$ are the postures of input and output motions, respectively. Each frame of an input motion is represented in the abstract representation $\mathbf{P}_i(t)$ and $\Delta\mathbf{P}_j, \Delta\mathbf{Q}_i(t)$ are applied to it using the scaling parameter *s* to compute the output posture $\mathbf{P}_i(t)'$.

The postures of all frames of base and reference motions are converted into the

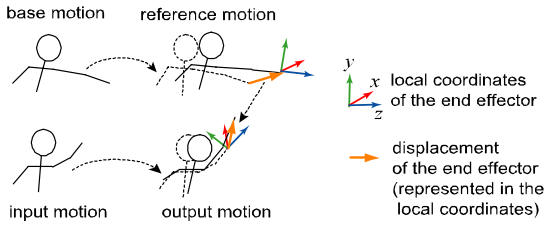

Figure 5: Displacements of the end effector in the local coordinates computed from the motion direction.

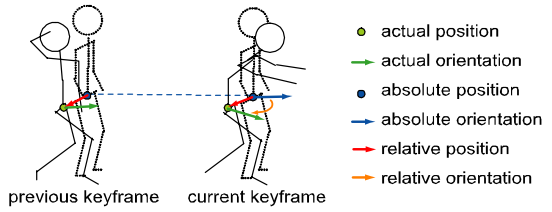

Figure 6: Absolute and relative pelvis position and orientation.

proposed representation $\mathbf{P}_{\text{base}}$, $\mathbf{P}_{\text{reference}}$, respectively. The differences in keytimes are then computed as follows,

$$\Delta\mathbf{P}_j = \mathbf{P}_{\text{reference},j} - \mathbf{P}_{\text{base},j} \tag{7}$$

where $\mathbf{P}_{\text{reference},j}$, $\mathbf{P}_{\text{base},j}$ are the postures of the base and reference motions, respectively, at bthe *j*-th keyframe.

Based on the computed postural features of keytimes $\Delta\mathbf{P}_1 \cdots \Delta\mathbf{P}_n$, the postural differences for all frames on each segment $\Delta\mathbf{Q}_i(t)$ are then computed based on the difference between the reference and the base motions after equation (6) has been applied as follows.

$$\begin{aligned}\Delta\mathbf{Q}_i(t) = \mathbf{P}_{\text{reference},i}(t) - \mathbf{P}_{\text{base},i}(t) \\ -(1-t)\Delta\mathbf{P}_i - t\Delta\mathbf{P}_{i+1}\end{aligned} \tag{8}$$

As result, $\Delta\mathbf{P}_j$ for all keyframes and $\Delta\mathbf{Q}_i(t)$ for all segments are computed and stored as postural features.

5.3. Representation of the Displacements of the End Effectors

We represent the differences of the positions and directions of the end effectors in local coordinates that are computed from the movements of the end effectors as shown in Figure 5. For example, if the position of the hand moves forward between the base and reference straight punch motions (Figure 5 above), our method transforms the displacement to the local coordinates of the hand of the uppercut motion and moves the hand forward (upward in the world coordinates) (Figure 5 below).

First, we compute the motion direction of each end effector on each segment *i* from the positions of the previous frame $\mathbf{e}_{\text{base},i}$ and the next keyframes $\mathbf{e}_{\text{base},i+1}$ as follows,

$$\mathbf{d}_{\text{base},i} = \mathbf{e}_{\text{base},i+1} - \mathbf{e}_{\text{base},i} \tag{9}$$

The direction $\mathbf{d}_{\text{base},i}$ is used as the Z-axis of the local coordinates $\mathbf{M}_{\text{base}}$, a 3×3 matrix. Second, we use the up-directional vector that crosses the Z-axis as the Y-axis. Then the Y-axis is computed from the cross product of the X- and Z-axes. After constructing the local coordinates $\mathbf{M}_{\text{base}}$, the difference of the position and orientations between the base motion and the reference motion are represented in local coordinates.

$$\Delta\mathbf{e} = \mathbf{e}_{\text{reference}} - \mathbf{e}_{\text{base}} \tag{10}$$

$$\Delta\mathbf{e}' = \mathbf{M}_{\text{base}}{}^{-1}\Delta\mathbf{e} \tag{11}$$

Equation (10) is included in equation (7) and equation (8) since $\mathbf{e}$ is a part of $\mathbf{P}$. Equation (11) is then applied to $\Delta\mathbf{P}_j$, $\Delta\mathbf{Q}_i(t)$.

When these features are applied to an input motion, the local coordinates of the input motion $\mathbf{M}_{\text{input}}$ are computed and the vectors are applied in the local coordinates,

$$\Delta\mathbf{e}'' = \mathbf{M}_{\text{input}}\Delta\mathbf{e}' \tag{12}$$

$$\mathbf{e}_{\text{output}} = \mathbf{e}_{\text{input}} + \Delta\mathbf{e}'' \tag{13}$$

As result, even when the directions of the end effectors' moments are different between the base and input motions, appropriate transformations are realized.

5.4. Computing Absolute and Relative Positions and Orientations of the Pelvis

We isolate the absolute and the relative position and orientation of the pelvis in each frame to apply their transformations. We consider that the absolute position and orientation $\mathbf{p}_{\text{abs}}$, $\mathbf{q}_{\text{abs}}$, respectively, are neutral positions and the orientations when a human figure stands without local displacements of the pelvis (Figure 6). The relative positions and orientations $\mathbf{p}_{\text{rel}}$, $\mathbf{q}_{\text{rel}}$ are considered to be the differences between the actual positions and orientations of the pelvis $\mathbf{p}$, $\mathbf{q}$ and its absolute positions and orientations $\mathbf{p}_{\text{abs}}$, $\mathbf{q}_{\text{abs}}$, respectively.

First, we compute the absolute position $\mathbf{p}_{\text{abs}}$ from the foot positions by supposing the human figure stands straight with no relative translations and rotations. In frames where both feet contact the ground, the horizontal absolute position $\left(\mathbf{p}_{\text{abs,x}}, \mathbf{p}_{\text{abs,z}}\right)$ is computed as the center position between the feet. In addition, the height of the pelvis in the rest posture is used for the vertical position $\mathbf{p}_{\text{abs,y}}$. When both feet are not contacting the ground, the absolute position is computed by linearly interpolating the absolute positions of the previous and next frames where both feet do contact the ground.

Second, the absolute orientation $\mathbf{q}_{\text{abs}}$ on each keyframe is computed from the absolute positions of the keyframe $\mathbf{p}_{\text{abs},j+1}$ and the previous keyframe $\mathbf{p}_{\text{abs},j}$. Since we make $\mathbf{q}_{\text{abs}}$ include only the horizontal rotation, $\mathbf{q}_{\text{abs}}$ is computed from the horizontal component of $\left(\mathbf{p}_{\text{abs},j+1} - \mathbf{p}_{\text{abs},j}\right)$. For absolute orientations during the motion segments, the orientations of the previous and the next keyframes are interpolated using spherical liner interpolation (SLERP) [15].

Once the absolute position and orientation $\mathbf{p}_{\text{abs}}$, $\mathbf{q}_{\text{abs}}$, respectively, are computed, the local coordinates of the pelvis $\mathbf{M}_{\text{base}}$ is computed from the absolute orientation $\mathbf{q}_{\text{abs}}$ in the same way with the local coordinates of end effectors. The relative position and orientation $\mathbf{p}_{\text{rel}}$, $\mathbf{q}_{\text{rel}}$, respectively, are then computed as follows,

$$\mathbf{p}_{\text{rel}} = \mathbf{M}_{\text{base}}{}^{-1}\left(\mathbf{p} - \mathbf{p}_{\text{abs}}\right) \quad (14)$$

$$\mathbf{q}_{\text{rel}} = \mathbf{M}_{\text{base}}{}^{-1}\mathbf{q} \quad (15)$$

As result, $\left(\mathbf{p}_{\text{abs}}, \mathbf{p}_{\text{rel}}, \mathbf{q}_{\text{rel}}\right)$ is stored to represent the posture of the pelvis in the frame and to compute the postural features. $\mathbf{q}_{\text{abs}}$ does not have to be stored since it is used only for commuting the local coordinates $\mathbf{M}_{\text{base}}$.

6. Extraction and Application of Style Features

When extracting style features, the system first computes temporal features $T_i(t)$, and then postural features at keytimes $\Delta\mathbf{P}_j$, and finally the postural features during segments $\Delta\mathbf{Q}_i(t)$ because we need to find frames that correspond between base and reference motions; first, for computing postural features, and then to apply postural features at the keytimes to the base motion to compute the postural features during the segments using equation (8).

Application of style features is processed in the same order. The scaling parameter specified by the user is multiplied with all postural features. Using the scaling parameter, the user can control the degree of exaggeration of the style application. During transformation, the posture on each frame of an input motion is converted to the abstract representation $\mathbf{P}(t)$. Then postural features $\Delta\mathbf{P}_j$, $\Delta\mathbf{Q}_i(t)$ are applied to them using equation (6). Based on the modified parameters, $\mathbf{P}'(t)$, the transformed posture is computed.

Since we simply add the extracted subtractions to the positions and orientations, this may cause unnatural postures. For example, if a transformation moves down the pelvis on an input motion in which the position of the pelvis is already low, the transformed position of the pelvis may be too much low. To avoid such unnatural postures, we limit the relative positions and orientations of the pelvis and the end effectors to within a feasible range. The joint rotations of input postures on the limbs are modified based on the position and orientation of the end effectors using an analytical inverse kinematics method [9].

7. Experimental Results

This section shows some experimental results generated using our method.

All motion clips in our experiments are 30 Hz motion and stored in a BVH format. The keytimes for each motion have been specified manually and extraction and application done automatically. The computational time for extraction takes a few seconds (3 to 5 seconds for the example motions on our experiments) on a standard PC. We apply style features in each frame during animation on the fly. Therefore, the application in each frame also takes very little computational time. In the following experiments, we used 1.0 for the scaling parameter for transformation.

7.1. Result 1: Walking and Running Motion

The first experiment relates to cyclic motions: walking and running. We extracted three kinds of features from walking motions (tired, happy, and drunken) and then applied them to a running and a backward-walking motion.

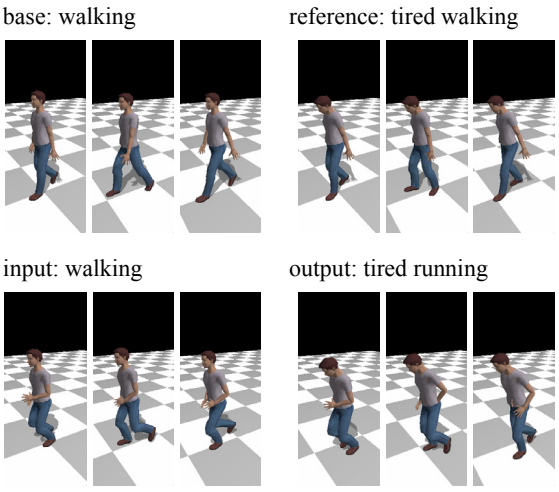

Figure 7: Tired running motion.

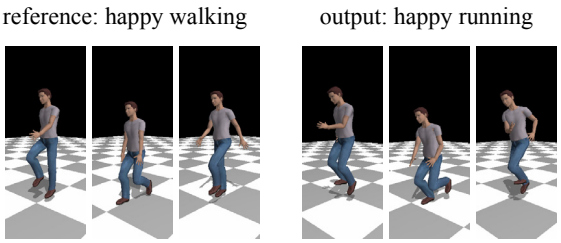

Figure 8: Happy running motion.

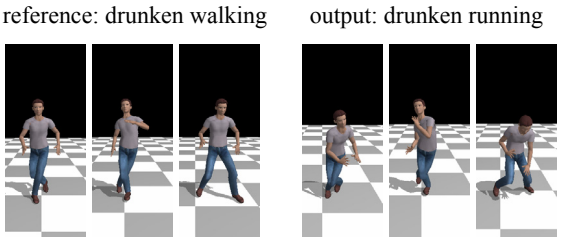

Figure 9: Drunken running motion.

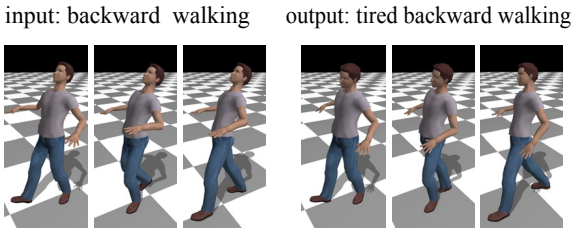

Figure 10: Tired backward walking motion.

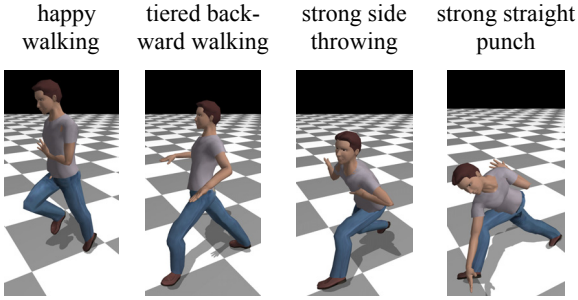

Figure 11: Results of motion blending.

We used one cycle of walking and running motions and set four keytimes for each motion. Figure 7 shows the results of a tired running motion. The "tired" features are extracted from the walking motion and the tired walking motion. When the running and tired running motions are compared, the results of motion transformation such as deceleration of running, tilting the upper body forward, and swinging the upper body laterally can be observed.

These styles have been successfully extracted from the tried walking motion and applied to the ruing motion. The tired running motion still has longer translations compared to the walking motion because the absolute movements of the pelvis in the walking motion are isolated and only relative movements are applied to the running motion.

Figure 8 is the result of a happy running motion. The applied features such as deceleration, vertical pitching of the pelvis and exaggeration of the swing of the hands and feet are observed. Figure 9 is a drunken running motion. Some irregular variations of motion velocities and directions are applied.

We also applied the same style features to a backward-walking motion which has different global translation from the walking motion (Figure 10). The same types of transformation with Figure 7 are realized successfully.

For comparison, we have tried to apply a motion blending method [14] to generate the same kinds of results (Figure 11). It showed that the blending method did not work well on these combinations of motions whose motion directions and global translations are different to each other as discussed in Section 2.

7.2. Result 2: Over Throwing and Side Throwing Motion

The second experiment is on non-cyclic motions: throwing and punching. We set five keytimes for each motion. We first extracted style features from an over throwing motion and a strong over throwing motion. We then applied the style features to a side throwing (Figure 12) and a straight punch (Figure 13) whose motion directions of the right hand are different from the over throwing motions. Thanks to being able to represent the differences of the movements of the right hand in local coordinates, an increase of the right hand's movements along the moving direction is successful applied.

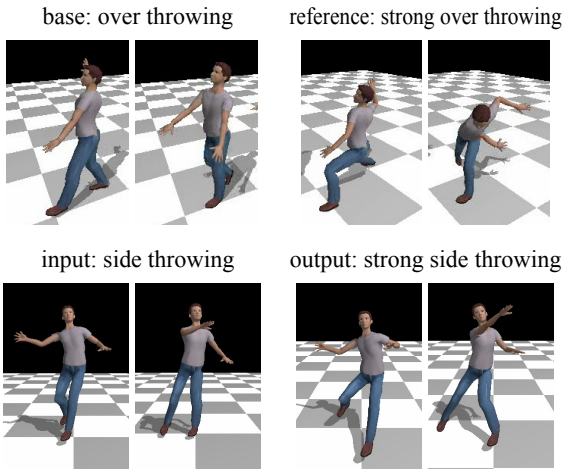

Figure 12: Strong side throwing.

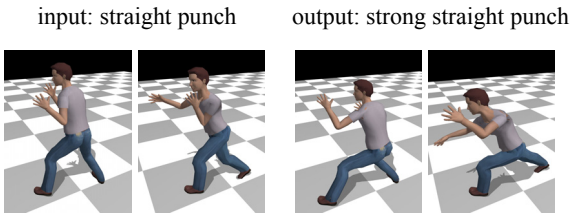

Figure 13: Strong straight punch motion.

7.3. Interactive Control Style Features

We also developed an interface that a user can use to control the scaling parameter and to see the results interactively. Since we simply apply extracted subtraction to the postural parameters, using a large scaling parameter (about 1.0 or larger) sometimes causes motions that are too exaggerated and unnatural. By utilizing the interactive controlling interface, a user can easily choose an appropriate scaling parameter.

8. Discussion

Using our method, the extracted style features are applied to a wide range of input motions. As we intended, the extracted style features can be applicable to input motions whose global translation is different from the example motions (Section 7.1) and in which the motion direction of the end effectors is different from the example motions (Section 7.2). However, the kinds of the motions have to be basically the same. For example, style features from a walking motion would be difficult to apply to a throwing motion (cyclic to non-cyclic motion). Although currently our method does not allow this, we could apply a style feature extracted from an end effector to another end effector. For example, with some adjustments a style feature of the right hand extracted from punch motions could be applied to movements of the left foot in a kicking motion.

Our method relies on given sets of keytimes. To effectively capture the temporal and postural features, keytimes have to specify the exact transition points between two serial motion segments. Specifying such keytimes is sometimes difficult since the movements of body parts are not always synchronized and appropriate keytimes can vary for body parts. For example, the transition timings of the hands and the feet of a walking motion are slightly different to each other. Currently we use one set of keytimes for the whole body and so the style features of some body parts may not be well captured. Considering these problems, finding effective keytimes can be difficult for novice users. Combining an automatic keyframe detection algorithm [2] with our method would be helpful.

Our motion transformation is purely kinematic and for now the physical aspect is not considered; therefore, it is possible that physically unnatural motions are generated. For example, unbalance postures might be generated during the motion, or the laws of the gravity might be distorted because of time-warping. Moreover, self collisions can occur. Applying a motion filter [18] to generated motions after motion transformation could be a solution to such problems.

As mentioned in Section 1, as a future work, we plan to develop a motion transformation system in which a number of extracted features are stored and can be used by users to apply various styles to input motions. To realize such a system, we need to find an appropriate style feature from the database based on the specified style and the input motion. To make a motion transformation work well, the selected style feature should be extracted from example motions that are close to the input motion. We need to establish criteria for the element of closeness between motions, such as the type of motions (cyclic or non-cyclic), primary body parts and motion direction. In cases where the input and example motions are not close to each other, applying only an abstract feature such as $T_i(t)$, $\Delta\mathbf{P}_j$ but $\Delta\mathbf{Q}_i(t)$ is an option to might generate better results.

9. Conclusion

In this paper, we presented a method that applies style features extracted from base and reference motions to an input motion and so transforms the input motion into a styled motion. We represent the difference between two motions in an abstract form in order to apply them to a wide range of motions. The experimental results show that our method works successfully.

Future work will include the application of style features to different end effectors, consideration of variations in body lengths, better clarification of the applicable range for our method, and the development of a database and transformation system.

References

[1] Ashraf G, Wong KC. Dynamic Time Warp Based Framespace Interpolation for Motion Editing, Proc. of Graphics Interface 2000. 45-52, 2000.

[2] Assa J, Caspi Y, Cohen-Or D. Action synopsis: pose selection and illustration. ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005), 24(3), 2005.

[3] Brand M, Hertzmann A. Style Machines. Proc. of SIGGRAPH 2000, 183-192, 2000.

[4] Bruderlin A, Williams L. Motion signal processing, Proc. of SIGGRAPH 95, 97-104 (1995).

[5] Hsu E. Pulli K, Popovic J. Style Translation for Human Motion. ACM Transactions of Graphics (Proc. of SIGGRAPH 2005), 24(3), 1082-1089, 2005.

[6] Kawasaki R, Kitamura Y, Kishino F. Extraction of Motion Individuality in Sports and Its Application to Motion of Characters with Different Figures. Proc. of Computer Graphics International 2003.

[7] Kovar L, Gleicher M, Pighin F. Motion Graphs. ACM Transactions on Graphics (Proc. of SIGGRAPH 2002), 21(3), 473-482, 2002.

[8] Kovar L, Gleicher M. Automated Extraction and Parameterization of Motions in Large Data Sets. ACM Transactions on Graphics (Proc. of SIGGRAPH 2004), 23(3), 559-568, 2004.

[9] Lee J, Shin S Y. A Hierarchical Approach to Interactive Motion Editing for Human-like Characters. Proc. of SIGGRAPH 99, 39-48, 1999.

[10] Lee J, Chai J, Reistma P, Hodgins J, Pollard N. Interactive Control of Avatars Animated with Human Motion Data. ACM Transactions of Graphics (Proc. of SIGGRAPH 2002), 22(3), 491-500, 2002.

[11] Liu CK, Hertzmann A, Popovic Z. Learning Physics-based Motion Style with Nonlinear Inverse Optimization. ACM Transactions on Graphics (Proc. of SIGGRAPH 2005), 24(3), 2005.

[12] Ménardais S, Kulpa R, Multon F, Arnaldi B. Synchronization for Dynamic Blending of Motions. In Proceedings of the 2004 ACM Siggraph/Eurographics Symposium on Computer Animation 2004, 2004.

[13] Neff M, Fiume E. Aesthetic Edits for Character Animation. Eurographics/SIGGRAPH Symposium on Computer Animation 2003.

[14] Park S L, Shin H J, Kim T H, Shin S Y. On-Line Motion Blending for Real-time Locomotion Generation. Computer Animation and Virtual Worlds, 15(4), 125-128, 2004.

[15] Pletinckx D. Quaternion calculus as a basic tool in computer graphics. The Visual Computer, 5(2), 1989.

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

[17] Shapiro A, Cao Y, Faloutsos P. Style Components. Proc. of Graphics Interface 2006.

[18] Tak S, Song O Y, Ko H S. Spacetime Sweeping: An Interactive Dynsmic Constraints Solver. Proc. of Computer Animation 2002, 2002.

[19] Unuma M, Anjyo K, Takeuchi R. Fourier Princeples for Emotion-based Human Figure Animation. Proc. of SIGGRAPH 95, 91-96, 1995.

[20] Wiley D J, Hahn J K. Interpolation Synthesis of Articulated Figure Motion. IEEE Computer Graphics and Applications, 17(6), 39-45, 1999.

[21] Witkin A, Popovic Z. Motion Warping. In Proc. of SIGGRAPH 1995, 105-108, 1995.