Interactive Motion Browse and Synthesis from Unorganized Motion Data Set

Yoshihiro Yasumuro[1], Taku Shirazawa[2], Muneyuki Sakata[3],
Masataka Imura[4], Yoshitsugu Manabe[4] and Kunihiro Chihara[4]

[1]Faculty of Environmental and Urban Engineering, Kansai University
3-3-35, Yamate-cho, Suita, 564-8680, Japan

[2]Sony Digital Network Applications, Inc.
[3]Positron Medical Center, Tokyo Metropolitan Institute of Gerontology
[4]Grad School of Information Science, Nara Institute of Science and Technology

Abstract

ABSTRACT
The motion capture technique is gathering more and more attention because of its powerful potential for providing life-like motions in computer graphics (CG) animation via sample-based motion creating techniques. Since motion data is multi-dimensional and spatio-temporal data that is difficult to edit as desired, an effective scheme for reusing captured motion sets to create new motion is advantageous. Reusing a motion data set requires effective browsing and extraction techniques that enable the user to look up and capture the relations among the motion contents in the motion data set. We propose a new framework for a motion editing tool by focusing on the connectivity between motions and utilizing it as a filter to extract the desired motion contents from a motion database. The proposed system uses a tree structure for expressing possible connective motion paths in the motion database. The motion connective tree can be a useful user interface to browse and select a motion scenario by exploring the existing motion data sets in the database. Our prototype system demonstrates an easy-to-understand interface to explore and quickly edit a motion data set by selecting icons of the motion tree nodes.

1 Introduction

In computer-generated animation, human-like characters play major roles in presenting the contents and drawing the audience into the action. Preparation of the actions to be performed by the characters is a key issue in terms of both cost and quality. Designing and editing the motion data for a realistic human-like character is a costly task requiring interactive refining of sequences of poses. Motion capture is a powerful method for acquiring motion data directly from actual human actions. Life-like physical motion, including subtle behaviors, can be recorded and replayed on digital characters. While motion capture has an established market in the visual production field, the system has a high cost for installation and operation, and the quality of the motion data depends on the physical actions of the actors. Therefore, motion capturing is often employed for specific and professional uses, such as movie productions, with elaborate scenarios not only for the story but also for the behavior, action and camera work. However, there is a growing trend to distribute motion data for smaller projects or personal use [1, 2]. To provide a significant technique for reusing motion data for varied applications, we must consider that the motion data resources could be commonly distributed as multimedia data such as images, video clips, 3D models and so on. While standalone personal computer systems have become capable of computing high-quality graphics synthesis, human motion processing still depends on posing-based key-frame software. This research focuses on enhanced reusability of distributed motion capture data for any scale of use.

2 Related Work

For motion editing with sampled data, various techniques for modification and adaptation of existing motion data or for the synthesis of new motions from example motions have been developed [4, 5, 7, 8]. Prior to reusing and processing motion capture material, it is necessary to identify and extract suitable motion clips from the database. CMU Graphics Lab Motion Capture Library is a web-based database for captured motion data, and academic and commercial projects have permission to access its data sets [1].

978-0-7695-3456-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DICTA.2008.67

IEEE computer society

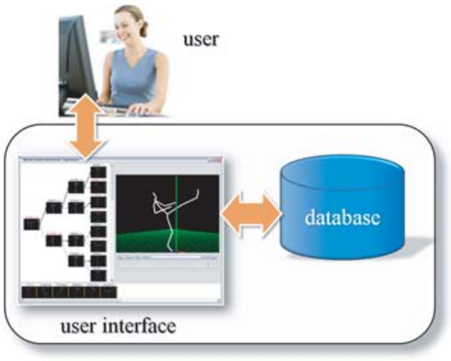


Figure 1. System concept

There is a total of 2605 captured motions. Each set of data has a short description indicating the feature of the motion contents. Features are categorized into 6 major and 23 sub groups. The user may search a motion by either inputting a keyword of the motion feature or choosing one of the categories and looking up the listed motion data contained in it. Users may describe the motion clips to be retrieved in various ways at different semantic levels. Another search mode would involve a short query motion clip. In this case, the task would be to retrieve all clips in the database containing parts or aspects similar to the query. This kind of problem is commonly referred to as content-based retrieval.

Kuriyama et al. constructed a web-based database system, ToMoLow, for distributing motion data [2]. Like the CMU database, each motion data has a notation of its feature categorized into 4 major and 12 sub groups. Both a category search and posture image search are implemented by the ToMoLow search system, Motion Map [11]. Motion Map uses a Self-Organizing map (SOM) for mapping posture data sets into 2D space. A posture data set is generated by splitting motion data into time steps. In 2D space, posture images are mapped for visualizing the motion distribution. The user can select one of the data by clicking the iconized images of the posture on a graphical user interface (GUI). This approach has a goal similar to ours. The system automatically organizes the motion data contained in the database and shows it to the user for intuitive selection. The GUI functions for ToMoLow are focused on visualizing motion data sets and is totally separate from editing applications.

3 Proposed Approach

3.1 Motion Connectivity Tree (MCT)

To make the most use of existing motion samples, a useful visual interface is required for checking the contents by setting the timeline of the motion as well as browsing the indexes of the motion clips effectively. The similarity of skeletal motions is a clear criterion for categorizing motion data. Kovar used similar motions as a query to retrieve motion from a motion database [9]. We use similar motion as a node that may transit to other motions. There may be multiple nodes in a motion data that segment the motion data. The user can explore a motion database by tracing the nodes and examining possible variations from one motion segment to other segments in different motion data.

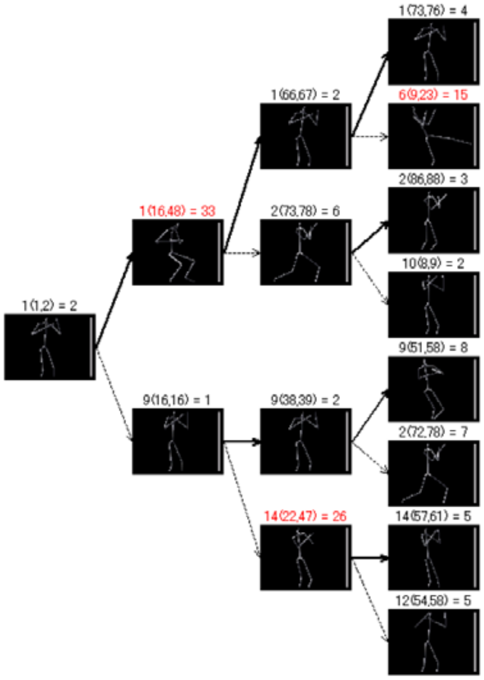


Figure 2. Motion Connective Tree

By finding similar articulated human postures between different motion frames, connective frames can be set. The connection of multiple motion segments can be expressed by a digraph structure. However, a digraph may often contain looped routes that make it difficult to understand the timeline for the user. To display several possible digraphs, a large screen area is needed. A tree structure is a compact method for expressing multiple diverse cases of connective motion segments without looped shapes. Moreover, the parent-to-children connection can be sorted as the time flow direction to replay a connected motion sequence. This allows the user to see motion sequences intuitively and to examine multiple cases of possible motion transitions. We call the tree structure an MCT (motion connective tree), and the motion editing tool software we propose uses a GUI with MCT representation. The root of MCT is set on the node currently selected by a user. The leaves are possible connective motion data within the database. Figure 2 shows an MCT example with a depth of three layers. A motion is selected for the starting point and set as the root. Consecutive connective motion segments are set as the nodes. A characteristic frame from the motion segment is put on the node by the system. Solid lines connect the same motion divided into segments and dashed lines express the transition between different motions.

3.2 MCT Construction

To construct an MCT, not only articulate posture information but also connectivity information is necessary. We define connectivity as a pair of time periods that can be connected to one another as continuous and natural motions. First of all, the distance between two different motion segments is examined. The distance is defined as the summation of the absolute differences of the joint angles of the two frames. The distance is computed for each different pair of frames from the two motions Figure 3 (left) shows the connectivity of a pair of two motion segments. The vertical and horizontal axes denote the frame numbers of each motion segment. The distance between frame x in a motion and frame y in the other motion is plotted at (x,y); thus, every pair of the two motions is depicted as a matrix, as in Motion Graph [10]. Consecutive frame pairs with lower distances and higher similarity are picked up as connective regions. The dark color expresses similarities in the figure. Longer connective regions are picked up as larger square shapes in the matrix (white squares in Fig. 3) (right)). This process is executed for each pair of motion data in the database beforehand and the list of connectivity regions are stored. MCT is dynamically constructed by searching the connective points from the list, according to the user's selection of the connective nodes on the GUI.

3.3 Motion Database with MCT

A database is a suitable framework for handling a great amount of motion data and for accepting more data sets. Moreover, adding a user interface with MCT in the front-end of the database improves management and reusability of the motions. Therefore, we design the database to have a connectivity index as well as motion data. The database structure is shown in Fig. 4. Our database consists of a motion information table, motion data and a connectivity tree table. The motion information table contains serial numbers of the motion data, filenames and degrees of freedom of the motion. The motion data stores the time series of motion data for each joint in a file. Connectivity is expressed by a description of the index of two files which can be connected easily, that is, the starting and ending frame of the two files.

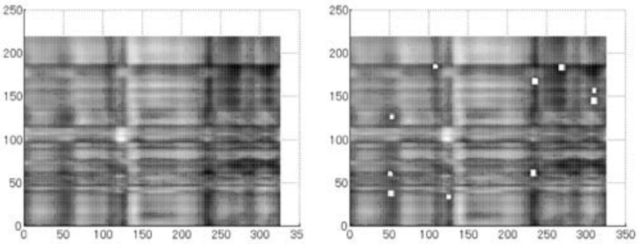
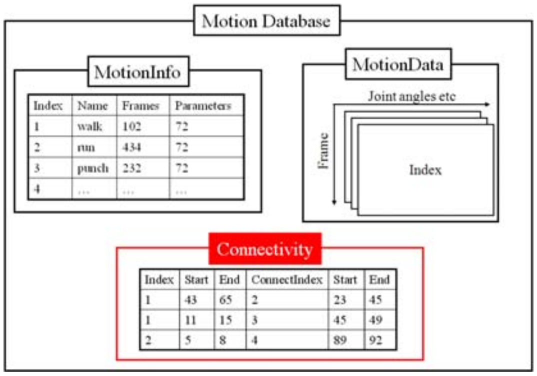


Figure 3. Visualized connectivity between different motions



Index	Name	Frames	Parameters
1	walk	102	72
2	run	434	72
3	punch	232	72
4	...	...	...

Index	Start	End	ConnectIndex	Start	End
1	43	65	2	23	45
1	11	15	3	45	49
2	5	8	4	89	92

Figure 4. Database structure

This database structure enables users to access the possible connecting points of any motion data files. By recursively referring to the connecting points, an MCT is constructed, and an arbitrary motion file can easily be set as the root.

In the operation of editing motion, we focus on the capability of selecting and connecting existing motion data. We consider that this operation is a minimal task in designing motion; however, quick pre-processing for prototyping makes the most use of the data set.

Effective browsing of the data set allows the user to acquire new data by knowing what data is already missing in the database.

Effective scheme for brousing the data set helps to plan for acquring new data by making clear what kind of data is missed in the database.

4 Motion Blending

Here, we employ a fundamental blending method to connect selected motion clips without much alteration of the original motion contents. Motion blending basically generates a new posture sequence that is located intermediately between two existing sequences. The ratio for blending smoothly varies from 1:0 to 0:1, which is equivalent to the weight of one motion on the other. In this paper, a cubic function is used for this transition. The connecting region to be blended has F frames, out of which the weight value α of frame x is expressed as:

$$\alpha(x) = 2(\frac{x}{F+1})^3 - 3(\frac{x}{F+1})^2 + 1 (1 \leq x \leq F). \quad (1)$$

Articulate postures in motion data frames are described based on a tree structure whose root is set at the hip. Since the hip position generally varies from frame to frame, the hip position needs to be consistent at the starting and ending points of the blending frames. The starting hip position of the successive motion is connected to the next successive hip position for the hip position blending result. Thus, connected motions are expressed relative to the preceding ones with successive offset of the hip position.

5 Experiments

5.1 Setups

First of all, we prepared a motion data set, which included 14 characteristic motions of a fighting sport, such as punching, kicking, and stepping motion. Each motion was captured by an MX camera system (ViconPeaks, Inc.) Fifteen markers were used for tracking the major body parts according to a standard marker setup suitable for the BVH (Biovision hierarchical) motion data format. The sampling rate was 30 Hz. We used MotionBuilder (AutoDesk inc.) software for converting the raw motion data expressed in a time sequence of the 3D positions of the 15 markers into a BVH motion data standard format. BVH data describes the time sequence of joint angles based on a tree structure to express articulated human body structure. The root of the tree is the hip, whose children expand toward the hands, toes and head.

Second, we executed a connectivity analysis for every pair of motion data in the data set and constructed a table listing possible connectivity periods for every motion data. The total number of listed connectivity periods was 517. The minimum length of the connective period and the threshold of connective similarity was determined in an ad-hoc manner. The motion database was implemented by MySQL [12], which is a multi-threaded SQL database framework with reasonable speed and robustness for multi-user access; also, MySQL is suitable for use over a network.

The user interface for the prototype application tool for editing motion sequences is shown in Fig. 6. This GUI consists of three major panels: motion selecting interface (top left), preview monitor (top right), and motion sequencer (bottom). The motion selecting interface shows the possible route to connecting motion clips in the database from the one the user picked or to the preceding clip selected. This view shows MCT, which is dynamically recreated for renewing the tree root. When the user selects a new motion segment, the selected motion segment is set as a new root of MCT, and possible successive motion segments across several layers are displayed. A preview monitor allows the user to check the listed candidate motions one by one, as well as track the motions already selected and connected as a single sequence.

In this implementation, the MCT was designed as a binary tree structure. The user interface displays four layers of the tree at a time for easier browsing. To start the experiments, an initial root node of the motion clip and period was set. The icons to show the articulated posture as the content of each connecting region of the motion were created by selecting the middle frame of the region. When the user selects a connective region of a motion, the system connects the motion data from the starting motion to the newest one and lists them on the motion sequencer at the bottom of the screen. The last region of the motion is set as the root and renews the MCT to show the next possible connective motions. The connecting process for interpolating two motion data was achieved by using a cubic function, as in Motion Graph.

Figure 5. Marker setup for Mocap

5.2 Experimental Scenario

To evaluate the usability of the proposed method, we conducted an experiment for selecting and connecting a motion as desired with our system. Though the desired motion is supposed to be achieved by trials reflecting the user's intent, this experiment set a clear target work of motion sequence as a goal. The target motion was composed of six selected types of motion, whose lengths were all different. This task was able to test the efficiency of examining the database and the reproducibility of searching materials in a

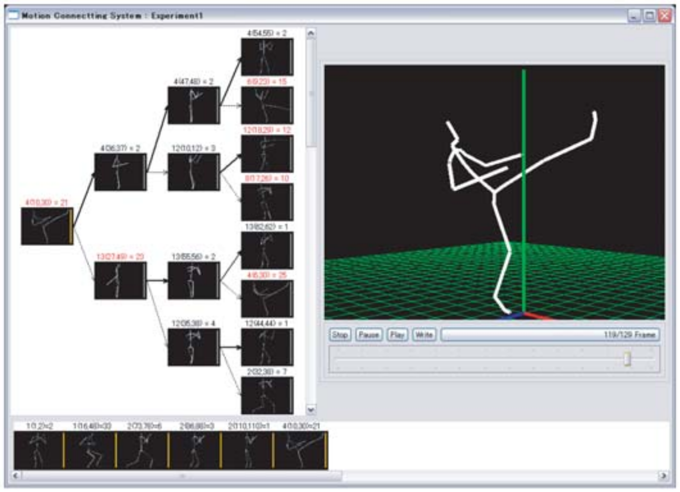


Figure 6. GUI for prototype system

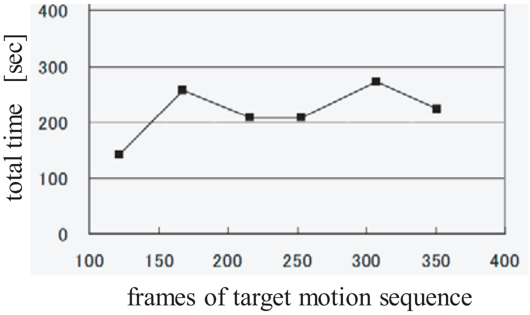


Figure 7. The time spent for the task

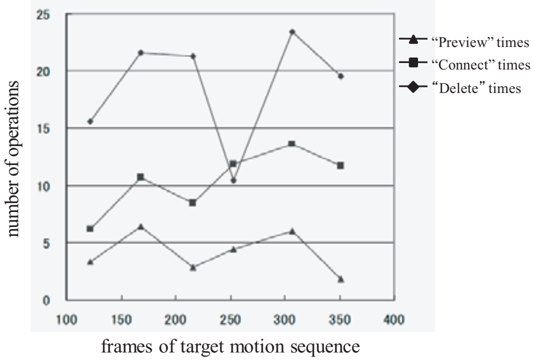


Figure 8. Number of operations

huge motion data source.

Another window in the GUI showed the target motion sequence by playing it as a movie file so that the user could check the target motion which is supposed to be the desired animation. The total time spent for exploring the database and creating the same motion data as the target was counted for each subject. The number of operations was measured by counting the number of mouse clicks for selecting, undoing, previewing and checking the MCT node contents. The subjects in this experiment were seven students, mid-twenties in age, who were enrolled in the graduate school of information science. They were allowed to become equally familiar with the proposed system, since they completed exercises of a similar task beforehand as practice.

5.3 Results

Eventually, all of the subjects successfully created the same target motion sequence. This shows the significant potential of our approach for reusing existing motion data sets. There are number of frames within a single motion that can be connected with other motions. Only the reasonable connective frames are listed in the MCT, and the system shows a motion connectivity map where the frames the user is now focused on is cantered. The time and operations for the subjects to achieve the task are shown in Figures 7 and 8. The average time for editing 100 to 360 frame lengths (3 to 12 seconds each) motion was 3 to 5 minutes. Browsing and selecting the motion clips seemed easy for the users, since the length of the target sequence correlates only slightly with total spent time for searching and editing. Rather than the number of frames to edit, the contents of the motion for selection affected the difficulty of the editing task. As seen by time-consuming result, the displayed motion on the MCT nodes was not clear enough to understand the kind of actions contained in the clip. The user had to spend time selecting and deleting more than once to exactly select the contents of the candidate motion.

Another problematic case was caused by motion data with many connective frames, such that the user could not find the next key feature scene within four layers, which we set as the depth of the MCT to display at one time. This might force the user to take a longer time to search for a motion that is comparatively far from the current node in the MCT. The visibility of the MCT nodes depends on the number of frames similar to other motion data in the database. We need to cluster the similarities between the connective regions and define the node's distance not only based on the time interval but also on the motion contents contained in the intervals. Showing the contents of the motion features, which are important clues to selecting the next motion candidates, can be improved by motion analysis for extracting unique actions and postures contained in a clip.

6 Conclusion

This paper proposed a tree structure approach for storing and mapping motion data sets with the goal of enhancing

Figure 9. Example of target motions

the usability of a number of existing motion data. The proposed framework is based on a combination of similarity-based segmentation and data visualization by the tree structure, which can be an intuitive GUI for exploring the motion database. Using a normal relational database structure, an interactive system was implemented for browsing and selecting the desired motion from a data set. Experimental results showed good reproducibility for browsing by actual motion selection and editing tasks.

Assuming the use of a larger scale for the motion database, our next step will focus on a hierarchical similarity computation of motion data for an effectively structured MCT by taking global and local similarities of the motion into account.

Acknowledgement This research is partly supported by the Japanese Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B) 19700116.

References

[1] Carnegie Mellon University Graphics Lab: CMU Graphics Lab Motion Capture Database, http://mocap.cs.cmu.edu/

[2] ToMoLoW: http://tomolow.vcl.ics.tut.ac.jp/

[3] Michael Gleicher: Retargetting Motion to New Characters, SIGGRAPH98, pp.33-42, 1998.

[4] A.Bruderlin and L.Williams: Motion Signal Processing, Proc. ACM SIGGRAPH95, pp.97-108, 1995.

[5] Andrew Witkin and Zoran Popovic: Motion Warping, Proceedings of SIGGRAPH95, pp.105-108, 1995.

[6] Rose, C., B. Bodenheimer, and M. Cohen: Verbs and Adverbs: Multidimensional motion interpolation using radial basis functions, IEEE CGAA, 1998.

[7] Pullen, K., and Bregler, C.: Motion capture assisted animation: Texturing and synthesis. ACM Transactions on Graphics, pp.501-508, 2002.

[8] Kovar, L., and Gleicher, M. : Flexible automatic motion blending with registration curves, In Proc. SIGGRAPH 2003 / Eurographics Symposium on Computer Animation, Eurographics Association, pp.214-224, 2003.

[9] Kovar, L., and Gleicher, M. : Automated extraction and parameterization of motions in large data sets. ACM Transactions on Graphics 23, 3, pp.559-568, 2004.

[10] L. Kovar, M. Gleicher, F. Pighin: Motion Graphs, Proc. SIGGRAPH 2002, pp.473-482, 2002.

[11] Y. Sakamoto, S. Kuriyama and T. Kaneko: Motion Map:Image-based Retrieval and Segmentation of Motion Data, Eurographics/SIGGRAPH2004 Symposium on Computer Animation, pp.259-266, 2004.

[12] http://www.mysql.com/

Table 1. Motion data set

Motion data	Number of frames
left punch	47
right punch	58
left upper punch	93
right upper punch	88
left low kick	73
right low kick	78
left middle kick	47
right middle kick	104
left high kick	60
right high kick	98
step forward (left)	115
step forward (right)	112
squat	87
walk back and forth	168

Table 2. Target motion

Index	Number of frames	Number of segments
1	122	8
2	168	10
3	216	13
4	253	14
5	307	19
6	351	20