Partial Motion Blending and Assembly for Interactive Motion Synthesis

Gwo-Hao Juang[†] Yan-Ju Chen [†] Jen-Yu Peng [†]
midas@caig.cs.nctu.edu.tw slimsatan.cs94g@nctu.edu.tw alvin@caig.cs.nctu.edu.tw

I-Chen Lin[†] Jui-Hsiang Chao[‡]
ichenlin@cs.nctu.edu.tw rickyzao@iii.org.tw

[†]Department of Computer Science, National Chiao Tung University, Taiwan
[‡]Network and Multimedia Institute for Information Industry, Taiwan

ABSTRACT

Interactively controlling or editing motion capture data is an intriguing topic in games or animation prototyping. However, data-driven approaches require a large data set to synthesize motion of high variety. In the first part of this paper, we propose novel partial motion synthesis to extend the control parameter space or variety of limited motion data. We extend the attack range of kicking or punching by blending body parts respectively and then reassembling. Users can simply assign a target position in the extended parameter space and will get a motion that hit the desired target. In the second application, we propose applying partial motion assembly for motion editing. Given a sequence of key postures, our system will retrieve resembling partial figures from data sets. While reassembling the different parts of character motion and adjusting the motion variance according to the query motion, our novel synthetic motions can preserve original style and naturalness.

Keywords

motion synthesis, partial motion blending, partial motion assembly

1. INTRODUCTION

Character animation has been widely used in entertainment industry. Two major animation processes have been developed to generate character animations: simulation-based and data-driven. Simulation-based motion synthesis generates human motions by applying kinematics, physics and biomechanical models. Those approaches provide a flexible representation of motion. Nevertheless, it is difficult to construct natural-looking motions without appropriate constraints, especially for those low-energy motions, such as locomotion. Data-driven approaches mostly use motion capture techniques (MoCap) to acquire real motion data of the performer. This technique can easily retain subtle details and styles of human motions to produce complex and realistic motions. In this paper, we employed it to accomplish our systems. However, the acquisition of motion capture data is always a time-consuming and labor-intensive job. Besides, the huge storage size is another problem. Therefore, how to efficiently reuse a compact motion database to enrich the motion variety is the focus of our work.

In this paper, we propose two approaches and their applications about partial body part synthesis. We first separate the human body into several parts and assembly these different partial figures to create novel and natural-looking motion.

The remainder of this paper is organized as follows. In the next section, we review several related literatures. Section 3 provides the process of parameterized partial motion blending. Section 4 describes the application of partial figure retrieval and assembly. Section 5 shows our experiments and results. And Section 6 concludes this paper.

2. RELATED WORKS

Reusing the existing motions to synthesize new ones has become an important issue in motion synthesis. Most related literatures in the last decade can be categorized into two techniques: blending and transition. The first one blends a set of motion samples with different weights to create novel but similar motions [Kov03a, Kov04a] while the other generates continuous motions by assembling several small clips [Kov02a]. Heck and Gleicher [Hec07a] proposed a graph structure combined with both techniques to synthesize a continuous motion while preserving highly control over the character.

However, those above-mentioned synthesized motions are constructed for the whole degrees of freedom (DOFs) in a human skeleton which causes the motion database with high redundancies.

Therefore, we concerned about assembling different partial motions of the human body. Al-Ghreimil and Hahn [Al03a] estimated the detailed motion information for parts of the articulated figure. They first collected the base motions with the same style of partial motion such as reach when sitting, reach when running and etc. They then computed the difference trajectories and stored the averaged one. Finally, they added all computed and selected trajectories to their corresponding ones in the base motion by a simple vector addition for each frame. Ikemoto and Forsyth [Ike04a] enriched motion database by transplanting limbs. They then used a radial basis function kernel SVM to determine the naturalness of synthesized motions. Besides, Heck et al. [Hec06a] treated human motion as a combination of upper-body action and lower-body locomotion. Therefore, the whole body is divided into two parts. They first aligned the two target motions in time domain, and then computed a rotation transformation for the splice joint to keep its natural balance

In addition to reduce required motion database while extending the variety of motion, the frequent human interventions should be avoided. Efficient motion retrieval can help us to achieve this goal. Arikan et al. [Ari03a] introduced a framework that allows the user to synthesize a novel human motion through rearranging the annotations of motion. First, the user annotates a large collection of motions and then paints a timeline with annotations from a defined vocabulary. This system assembles frames from motion database and demonstrates the results which perform the desired actions at specified times.

Another key for retrieval is the logical similarity. Instead of using numerical features, the geometric features are preferred to compare the similarity between different motion clips. Muller et al [Mul05a] introduced a class of Boolean features to express geometric relations between certain body points of pose. Besides, they also simplified the notation of the motion clips to provide users a flexible retrieval process. Users can concentrate on specified parts of motion instead of the full body motion. Our proposed retrieval method is inspired by this concept. In the next year, they propose a further concept called motion template captured the spatial-temporal characteristics in the matrix representation [Mul06a]. The inconsistent and variable aspects of motions can be learned to automatically find appropriate boolean features.

3. PARAMETERIZED PARTIAL MOTION BLENDING

In this section, we introduce the partial motion blending improved from motion parameterization. Our goal is to overcome the problem of defective parameter space caused by not-well-captured motion data. Figure 1 shows the framework.

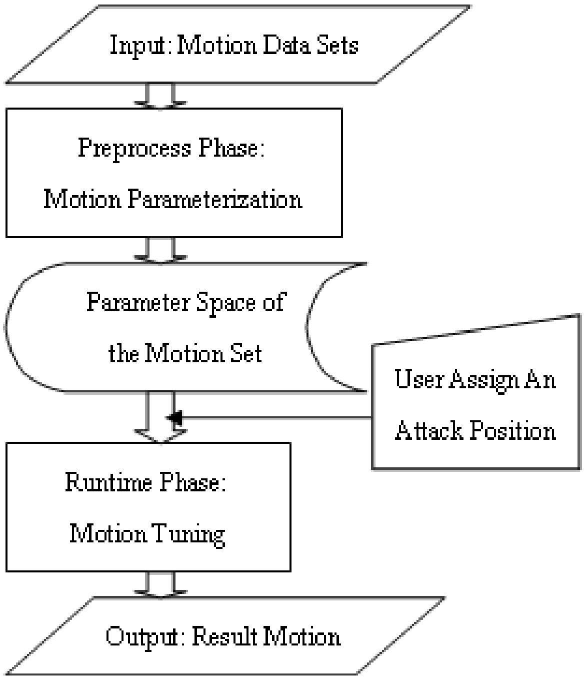


Figure 1. The framework of partial blending.

3.1 Motion Parameterization

The first step is to associate the control parameter with input motion sets. Since the MoCap performer is not a precise robot, the captured motions can’t be perfectly time-aligned, we need to find the optimal time alignment for all motions. After retiming, we can play them simultaneously. The time-alignment method is referred to Kovar and Gleicher [Kov04a].

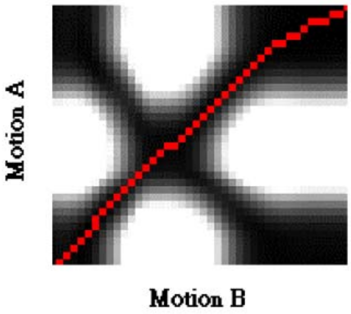
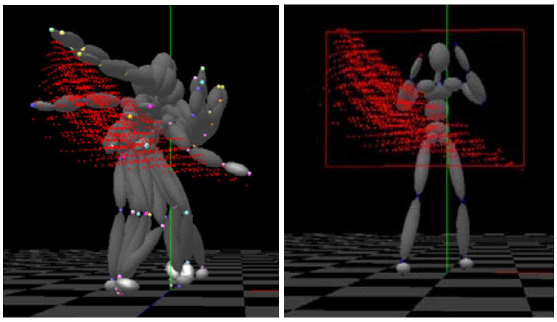


Figure 2. Distant Grid: The darker cells represent higher similarity, and the path in red shows time alignment between the two motions.

First, similar to other dynamic time warping algorithms, a distant-grid map is evaluated according to the similarity of each pair of frame data. We use the sum of square distance between each corresponding joint pair as our distance metric. While searching for the minimum cost path using the dynamic programming from the lower left cell to the upper right one, we can estimate the optimal time alignment, as shown in Figure 2.

After all sample motions are aligned together in time and starting orientation, we pick out a desired feature. For punching motions, the farthest stretched position of end-effector in right arm is an appropriate choice as the hit point. Then a denser sampling is applied to fulfill the parameter space by spherical linear interpolation over the sample motions to generate new blended motions and extract their parameters. In Figure 3(a), each red dot represents the hit position of one generated motion.

Obviously, if the sample motions are not well designed and captured, the affine-sum parameter space will be deficient. But with our runtime partial blending approach, we are able to extend the attack region to the red bounding box as shown in Figure 3(b).



(a) **(b)**

Figure 3. Denser sampling and extension of attack region.

3.2 Runtime Motion Tuning

Based on our observations, we found that a simple attack is the combination of two behaviors: the waist turns for the proper attack direction plus the precise hitting at a given height. Therefore, we separate the whole body as several synthesis units and apply blending respectively. After that, we reassemble them and adjust the orientations of some joints.

As in Figure 4, we define several adjustment joints and separate human body into segments. Different partial segments are represented in different colors where hollow joints are the adjustment joints.

When the user requests a parameter not included the defective parameter space, we query two parameters from the original space and derive the correlations between those queried motions to estimate proper adjustments of joint angle.

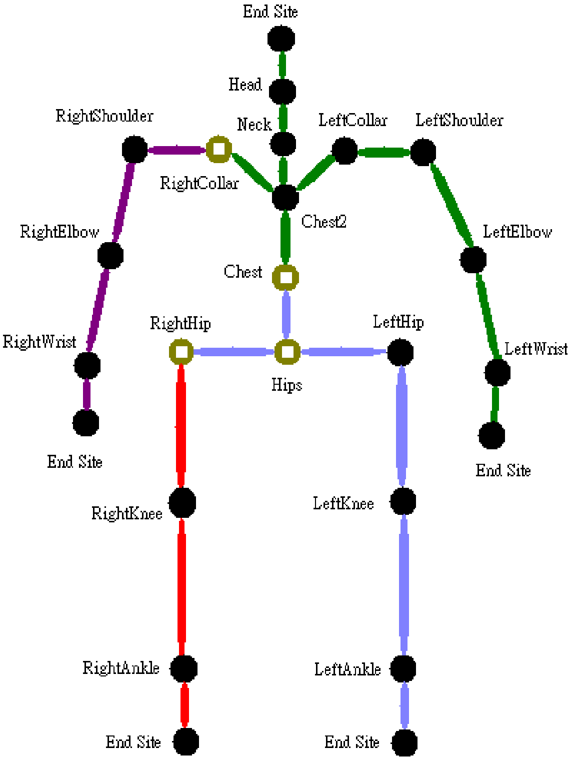


Figure 4. The hierarchical human skeleton.

For example in Figure 5, each red dot represents the hit position of blended motions. For the simplicity, we projected all of them on an x-z plane. We can see the constructed parameter space is deficient in four corners. Given a desired parameter P_T which indicates the target attack position excluded in this space, we first find one parameter P_H that hits the same height as P_T and is closest to P_T in horizontal direction. Then we query another parameter P_V which is closest to P_T in vertical direction and has similar waist-turning style as target motion.

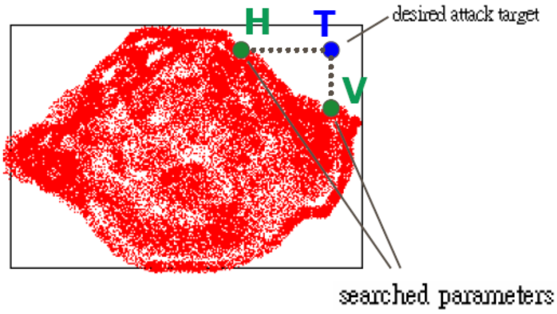


Figure 5. Parameter query for motion synthesis.

Due to the significant differences in actions between punching and kicking motion, we divide our refinement method into two cases. In punching motion, we generate desired motion using the blending weights of P_H, and adjust the predefined joints according to the information of motion M_V. Therefore, its attack direction is adjusted properly.

We then apply hierarchical adjustments to “Hips”, “Chest”, and “RightCollar” to get the final result.

In kicking motion examples, we splice the right leg of motion M_H and the rest parts of motion M_V for proper facing direction. We first rotate “RightHip” in y direction for direction adjustment of right leg, and rotate “RightHip” in x direction for the adjustment of kicking height.

Finally, the tuned motion is shortened and time-warped to get a smoother result.

4. PARTIAL FIGURE RETRIEVAL AND ASSEMBLY

To retain the variety of example-based motion editing, it is usually unavoidable to utilize a considerable set of motion samples or else a time-consuming optimization has to be applied. In this section, we introduce how we utilize the partial motion synthesis for motion editing with a compact data set.

First, an intuitive key-framing animation generator is provided for users to shape their desired motion (as shown in Figure 6). Our system will then retrieve motion clips that are most similar to the designated one. However, since only a compact data set is utilized, the retrieved motion may beyond users’ expectation.

To live up to users’ expectation, our system will perform similar motion retrieval for each body part respectively, and then reassemble the partial figures for the final results.

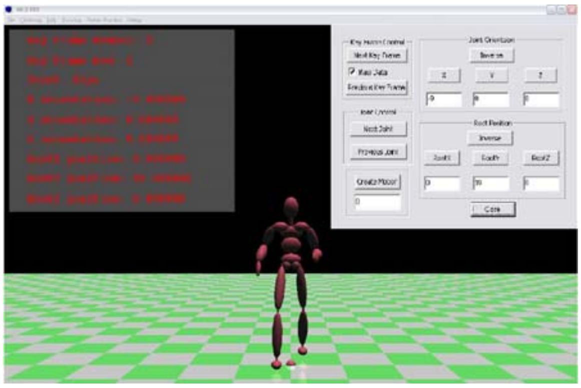


Figure 6. Key-framing animation generator.

4.1 Partial Figure Retrieval

The first phase handles the work of motion retrieval. We adopt the concept of relational motion features proposed by Muller et al. [Mul05a] and define several geometric features (as shown in Table 1). For efficiency of on-line retrieval, our system first pre-compute the Boolean feature values of all poses in database. Once the user inputs a query motion, we match the geometric features between query motion and those in database to retrieve the similar partial figure of character motion. In fact, our retrieval is a simplified version compared to Muller’s because we only work on the partial figures and mainly focus on locomotion and punching

Features	Description
F1	Is right foot in front? 1: in front, 0:behind
F2	Is left foot in front? 1: in front, 0:behind
F3	Does right elbow bend? 1: bend, 0:stretch
F4	Does left elbow bend? 1: bend, 0:stretch

Table 1. The features for locomotion and punches

4.2 Partial Figure Assembly

In the above steps, we have retrieved the partial figures similar to users’ query. Here, we explicate the details about the partial motion assembly. Several issues have to be settled.

4.2.1 Segment Alignment

The first issue is the velocity problem in different motion clips. We take a punch motion as an example. The query motion and retrieved motion do not have the same length. To solve this problem, we align the feature poses estimated in the retrieval step. As shown in Figure 7, the top motion sequence shows the 2th, 5th and 18th frames and the corresponding frames in the bottom one are the 7th, 9th, and 16th frames.

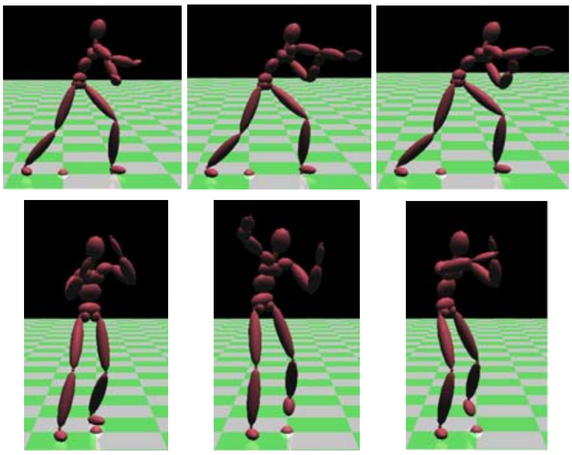


Figure 7. Punch motions with different velocities.

Such a solution according to logical similarity can avoid time-consuming dynamic programming alignment. After the alignment of feature poses, a Bezier curve interpolation is established for mapping between non-feature frames.

After this step, the partial figure motion would have the same speed with the query motion while retaining its style properties.

4.2.2 Different Type Adjustment

The second issue is the adjustment for natural-looking. In the previous section, we have extracted logically similar partial figures from different motion clips. However, directly assembling different parts of motion would cause strange results when the retrieved parts are not numerically analogue to the query motion. Figure 8 shows a case of punching motion. It might retrieve the arm from a kicking motion because they have logically similar Boolean value variation of arms.

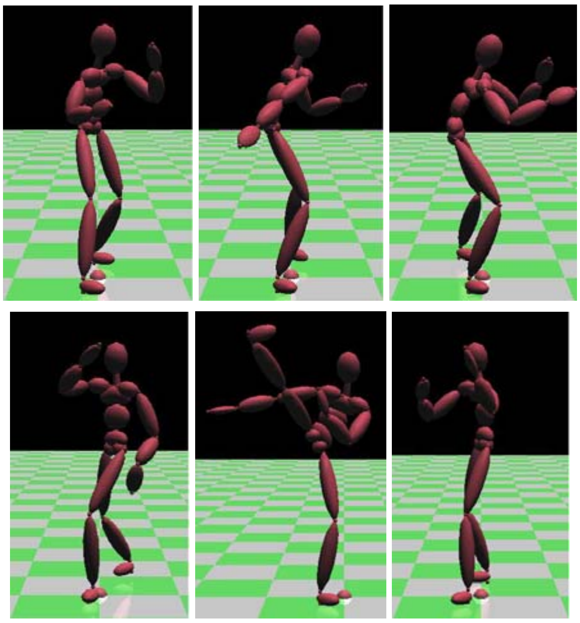


Figure 8. Two different motions with similar boolean value variation in arms.

From our observation, we noticed that improper angles of specific joints will degrade naturalness of the assembled results. Moreover, we found that the juncture is usually the critical joint.

For the sake of naturalness of assembled motion, we retain the statistical distributions of query motion to enhance the reliability of the assembled motion. The adjustment of critical joint is performed as the following equation:

$$S_i = \frac{R_i - M_R}{\sigma_R} * \sigma_Q + M_Q \qquad \text{for } 1 < i < m \qquad (1)$$

where R_i is the euler angle of critical joint in the i_{th} frame of retrieval motion. Here, we process the x-, y- and z-axis respectively. *M* is the size of the anticipative motion. M_R and M_Q are the means of the retrieved and query motion clip while σ_R and σ_Q are the standard deviations. The adjusted angles are stored in S_i

After adjustment, the critical joint can maintain the local variation of the retrieved motion and also conform to the global distribution of the query motion. We show the angle value of z-axis in Figure 9. After the statistical tuning, the local distribution of assembled motion is similar to the retrieved motion but the mean and standard deviation is close to the query motion. You can see the trajectory of assembled motion is similar to retrieved motion but the value is obviously different. With the hypothesis, we avoid the complex computation of physical simulation but still obtain quality assembled results for online editing.

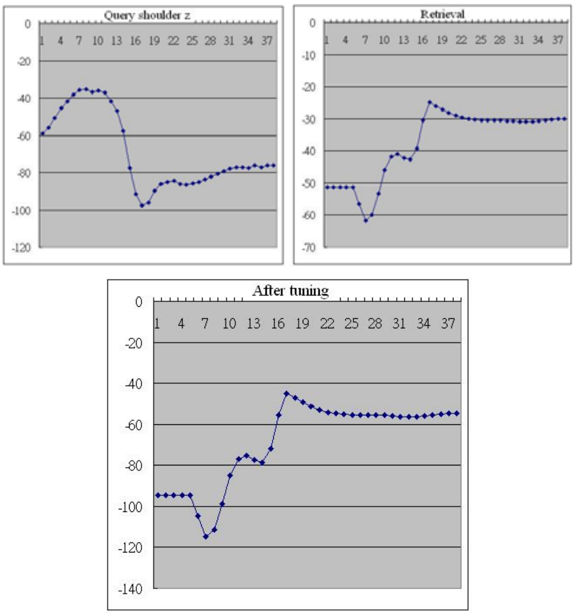


Figure 9. Critical joint adjustment. Query motion (top left), retrieved motion (top right) and assembled motion (bottom).

The Figure 10 shows a comparison between direct assembly and adjusted assembly.

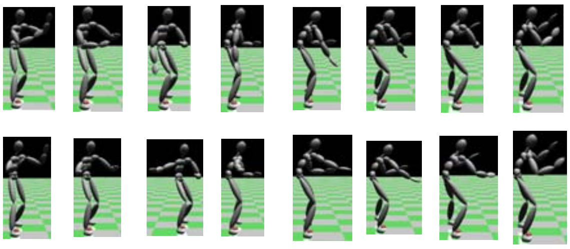


Figure 10. The direct assembly (top) and the assembly with statistical adjustment (bottom).

5. EXPERIMENTAL RESULTS

5.1 Environment and Results

Our experiments were performed on a desktop with a 3.20G Hz processor, 1.5 GB main memory, and a NVIDIA GeForce 6600 graphics card.

Two punching and two kicking motion sets are used in parameterized partial blending. There are straight punches in 7 directions, hook punches in 9 directions, straight kicks in 8 directions, and side-kicks in 8 directions.

The database used in partial figure retrieval and reassembly is listed in Table 2. It contains 22437 frames in 226 motions.

Type	#	Type	#	Type	#
Punch	40	Kick	25	Walk	10
Run	10	Jump	18	Shadowbox	40
Defense	6	Reach	7	Dodge	7
Push	16	Beaten	18	Pull	7
Combination	22	Total	226		

Table 2. Motion categories

Several results are shown as follows. First, an interactive attack with hitting position control is introduced as shown in Figure 11. A user can simply assign any desired hit point within the extended region and our system will synthesize the corresponding motion.

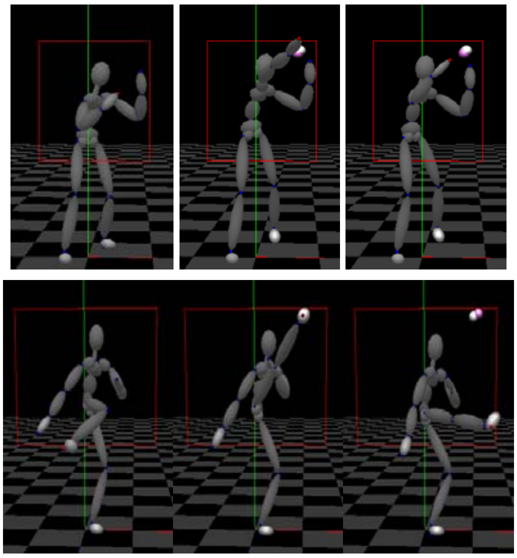


Figure 11. The parameterized partial motion blending attacks closely to the desired target.

In the other application, with our motion assembly technique, users can simply shape the desired motion by key-framing. Our system will first retrieve the most similar motion by matching the feature vectors of the whole body. If users are not satisfied with the preliminary result, our system will perform partial motion retrieval to replace designated parts and apply motion adjustment to retain its naturalness. Figure 12 and Figure 13 show the comparisons of punch and walk examples respectively.

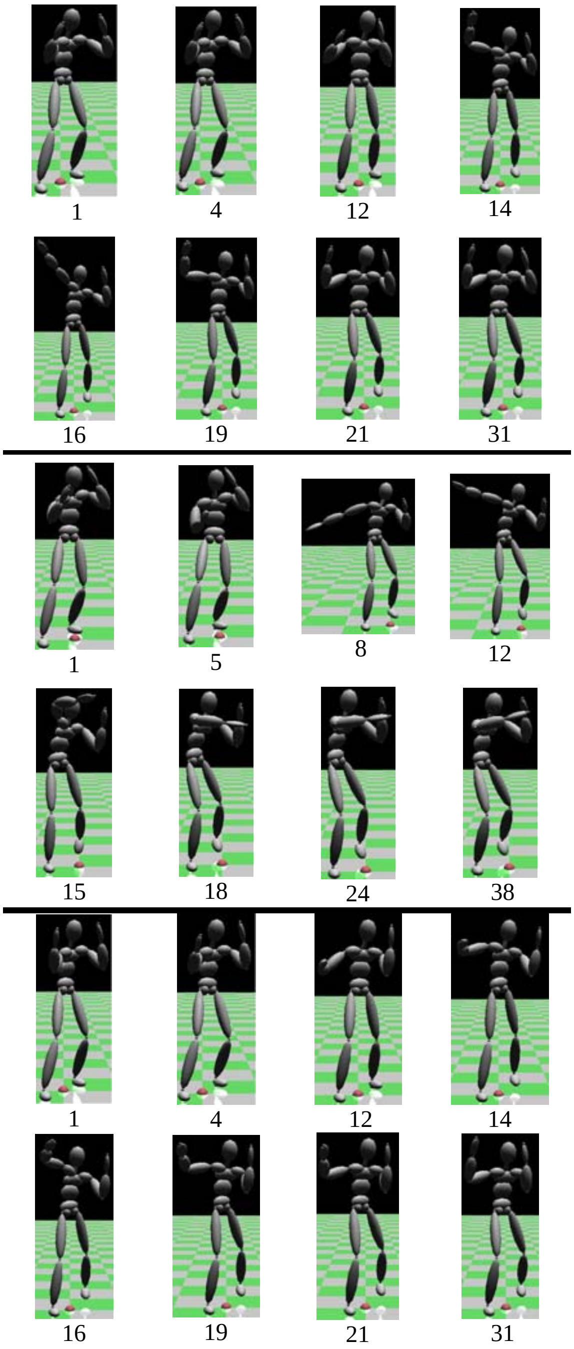


Figure 12. Punch: query motion(top), retrieved motion(middle) and assembled motion(bottom).

5.2 Experimental Analysis

In order to prove the utility of parameterized partial blending, we adopt the cross validation analysis by removing one sample motion and forming another parameter space smaller than the original one. Then we assign the same attack position which is within the original space but outside of newly formed shrunk space. Thus, we can compare the accuracy and joint trajectory of regular blending with the synthesized motion by our method.

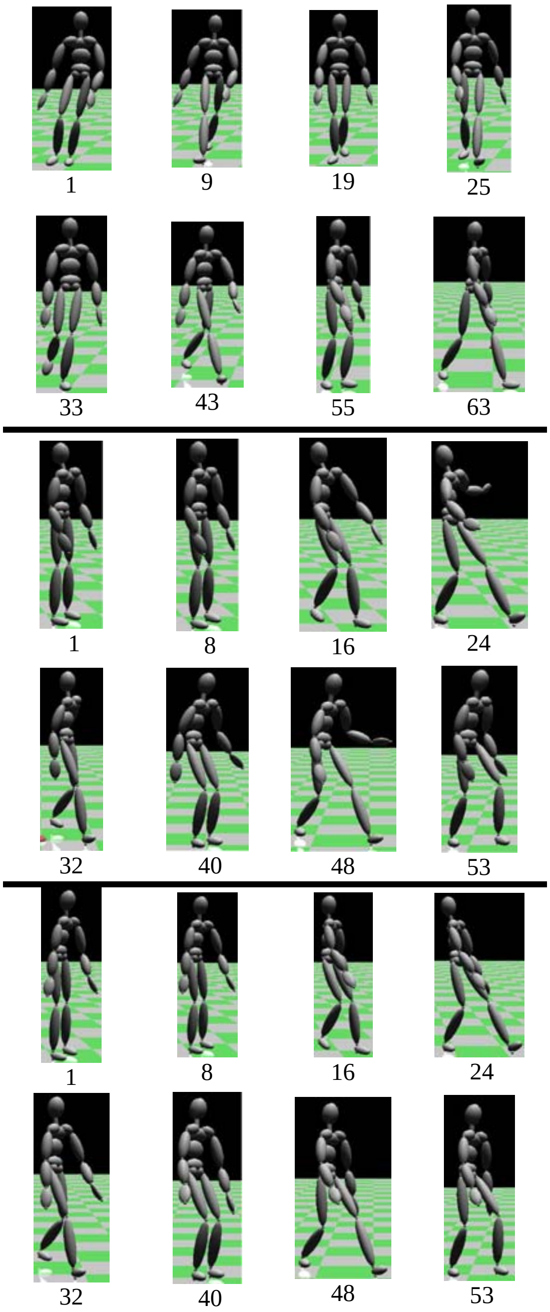
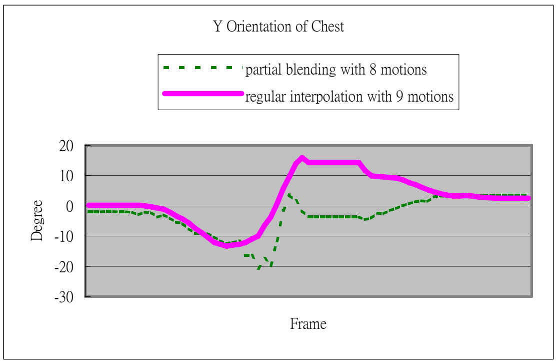


Figure 13. Walk: query motion(top), retrieved motion(middle) and assembled motion(bottom).

In punching motion examples as shown in Figure 14, the temporal trends of the medium part of y orientations of “Chest” are in opposite direction. The main reason is that we focus on adjusting the upper body. Since there are only slightly adjustments at “Hips”, the insufficiency of “Hips” orientation is mostly compensated by “Chest”, thus more right turns are taken in that temporal fragment when using partial blending. The purple point is the desired attacking position and the green and white points are the blended attacking point. This shows that the accuracy of our partial blending is as higher as regular blending, even with fewer motion data.



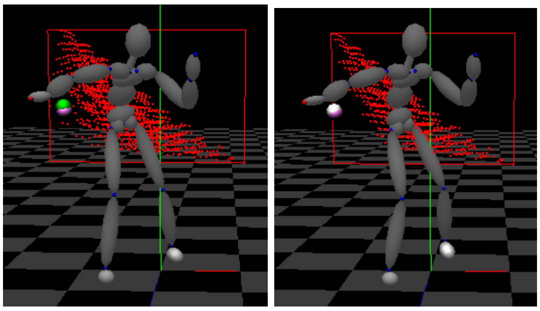
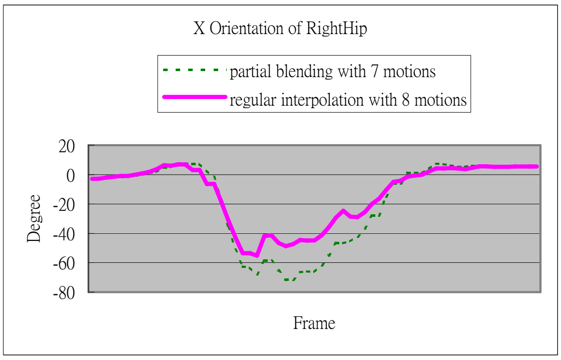


Figure 14. Hook punch. Left: Interpolate with 9 motions. Right: Partial blending with 8 motions

In kicking motion examples as shown in Figure 15, it is apparent that the x orientations of “RightHip” are shifted down to lift up the right leg additionally.



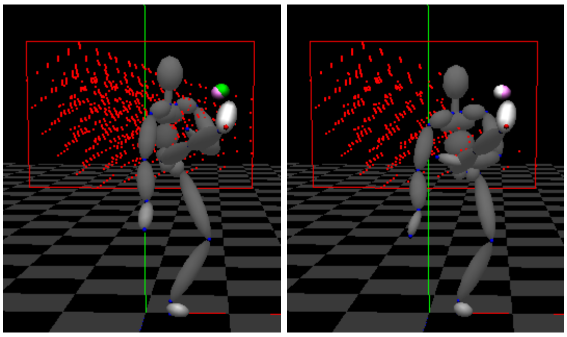


Figure 15. Straight kick. Left: Interpolate with 8 motions. Right: Partial blending with 7 motions

On the other hand, in order to verify the naturalness of our synthesized results in partial motion retrieval and assembly, we requested four experienced subjects (3 males and 1 female) to view the synthesized results and give grades.

We tested the punch and walk style motions generated by our system. The report show that 63% ~ 82% of assembled punching motions and 55%~66% of assembled walking motions are considered natural.

6. CONCLUSION

How to efficiently exploit motion capture data has been a critical issue for real-time synthesis. Based on the concept of partial motion synthesis, we propose two novel approaches for interactive motion control and motion editing. We extend the existing techniques to partial motions and provide the suitable adjustment frameworks to refine the assembly results.

Given a limited or defective motion data set, our proposed partial motion blending can extend the feasible attack range in real time. We demonstrate the utility by the analysis of cross validation. On the other hand, our novel motion editing system requires only assigning key postures and the retrieved result will be improved by partial motion assembly and adjustment.

With only a compact motion data set, our partial motion synthesis can increase the variety of synthetic motions dramatically. At present, our works are designed for interactive fighting games but will be extended for more general applications. Furthermore, physics-based adjustment will be included in the near future. Besides, we will conduct further user evaluation with more subjects and more detailed analysis.

7. ACKNOWLEDGMENTS

We would like to thank Ming-Shang Hsu and Ming-Han Tsai for their help of video recording. This research was supported by the III Innovative and Prospective Technologies Project of Institute for Information Industry and sponsored by MOEA, Taiwan. This research is also partially supported by National Science Council, Taiwan, with grant no. NSC-95-2221-E-009-164-MY3.

8. REFERENCES

[Al03a] Al-Ghreimil, N., Hahn, J. K. Combined partial motion clips. In WSCG. SHORT PAPERS proceedings, The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003.

[Ari02a] Arikan, O. and Forsyth, D. Interactive motion generation from examples. ACM Transactions on Graphics, 21, 3, 483-490, 2002.

[Ari03a] Arikan, O., Forsyth, D. A., O’Brien, J. Motion synthesis from annotations. ACM Transactions on Graphics, 22, 3, 402–408, 2003.

[Hec06a] Heck, R., Kovar, L., and Gleicher, M. Splicing upper-body actions with locomotion. Computer Graphics Forum (Eurographics 2006 Proceedings)25, 3, 459-466, 2006.

[Hec07a] Heck, R., Gleicher, M. Parametric motion graphs. In ACM Symposium on Interactive 3D Graphics and Games, 129-136, 2007.

[Hod98a] Hodgins, J. K., O'Brien, J. F., and Tumblin, J. Perception of human motion with different geometric models. IEEE Transactions on Visualization and Computer Graphics 4, 4 (October), 307-316, 1998.

[Ike04a] Ikemoto, L., and Forsyth, D. A. Enriching a motion collection by transplanting limbs. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 99-108, 2004.

[Kov02a] Kovar, L., Gleicher, M., Pighin, F. Motion graphs. ACM Transactions on Graphics, 21, 3, 473-482, 2002.

[Kov03a] Kovar, L., and Gleicher, M. Flexible automatic motion blending with registration curves. In Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Eurographics Association, 214–224, 2003.

[Kov04a] Kovar, L., and Gleicher, M. Automated extraction and parameterization of motions in large data sets. ACM Transactions on Graphics 23, 3, 559–568, 2004.

[Mul05a] Muller M., Roder T., Clausen M. Efficient content-based retrieval of motion capture data. ACM Transactions on Graphics 24, 3, 677–685, 2005.

[Mul06a] Muller M., Roder T. Motion Templates for Automatic Classification and Retrieval of Motion Capture Data. In Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Eurographics Association, 137–146, 2006.

[Yam04a] Yamane, K., Kuffner, J. J., and Hodgins, J. K. Synthesizing animations of human manipulation tasks. ACM Transactions on Graphics, 23, 3, 532-539, 20