Augmented Motion History Volume for Spatiotemporal Editing of 3D Video in Multi-party Interaction Scenes

Qun Shi Shohei Nobuhara Takashi Matsuyama

Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo, Kyoto, 6068501, Japan
seki@vision.kuee.kyoto-u.ac.jp {nob, tm}@i.kyoto-u.ac.jp

Abstract

In this paper we present a novel method that performs spatiotemporal editing of 3D multi-party interaction scenes for free-viewpoint browsing, from separately captured data. The main idea is to first propose the augmented Motion History Volume (aMHV) for motion representation. Then by modeling the correlations between different aMHVs we can define a multi-party interaction dictionary, describing the spatiotemporal constraints for different types of multi-party interaction events. Finally, constraint satisfaction and global optimization methods are proposed to synthesize natural and continuous multi-party interaction scenes. Experiments with real data illustrate the effectiveness of our method.

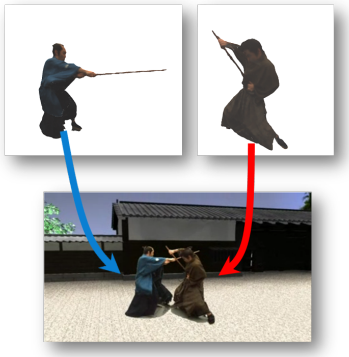


Figure 1. Multi-party interaction scenes editing from separately captured data.

1. Introduction

This paper is aimed at presenting a novel method to synthesize spatiotemporally synchronized 3D multi-party interaction scenes from separately captured data, while preserving the original motion dynamics of each object as much as possible. In the literature of virtual character motion editing, most conventional methods assume to have kinematic models of the objects. Such methods are known to work robustly for diverse motion editing tasks, but the modeling of multi-party interactions is still an open problem. In addition, these kinematic model-based methods are not directly applicable for certain kind of 3D data without a unified mesh model. In computer vision and computer graphics fields, in order to avoid occlusion problems or to increase the reusability of data, multi-party interaction scenes could be created by first capturing each object independently and then, synthesizing them together. Since separate capture will inevitably result in spatial and temporal mismatches, the synthesis of multi-party interaction scenes becomes a non-trivial task and requires large amount of manual work. Our research proposes to introduce the idea of augmented Motion History Volume (aMHV) to represent both the single object motion and multi-party interaction event. This approach allows the editor to realize natural 3D editing of multi-party interaction scenes from separately captured data with spatial and temporal mismatches semi-automatically, no matter kinematic models are available or not.

The technical problem we address for this research is how we can perform spatiotemporal alignments on separately captured data, which is reconstructed into temporal sequences of 3D meshes. Generally, the word interaction has different levels of meanings, so in this research work we narrow our focus onto the explicit interaction event conducted by spatiotemporally correlated body motions. The ideas behind our method are as follows. Basically, an interaction event can be considered as a pair of spatiotemporally synchronized motions performed by different objects. By representing the motions with time recorded volume data, we will be able to introduce various spatiotemporal con-

straints for multiple objects, based on which a multi-party interaction dictionary can be defined. Then for each interaction scene, rather than computing an individual frame, the aMHV considers a duration of motion simultaneously in computation. This makes it possible to consider spatiotemporal constraints on the entire motion, and to have objective criteria that describe entire motions. Thus for each interaction scene we can acquire a solution group with multiple reasonable relative locations of the two interacting objects by performing aMHV-based constraint satisfaction, while preserving the original motion of each object.

The overall processing scheme of the proposed method is as follows. Given two separately captured motion sequences, we first perform data segmentation and temporal alignment. Then we build up the aMHV for each motion segment in a multi-party interaction scene. Next, we decide the spatiotemporal constraints for each interactive motion segment pair and compute the solution group to realize the editing of each single interaction scene. Finally all the edited interaction scenes will be integrated together under a global optimization strategy. Figure 2 illustrates the computational processes of our method. It should be noted that step 3, 5, 6, 7 can be computed automatically using the proposed method, while step 4 requires some manual work from the editor. The two pre-processing steps are out of the scope of this article, and they could be done using conventional methods.

The main contributions of this paper consist of (1) we have augmented the original Motion History Volume [15] by encoding more temporal information and assigning labels onto its surfaces, (2) we have designed a novel method that models multi-party interaction events into an interaction dictionary, based on the aMHV. The possible applications include the making and editing of movies, computer animations and video games.

The rest of this paper is organized as follows. First, we review related studies to clarify the novelty of this work in Section 2. We then introduce our aMHV-based multi-party interaction representation in Section 3, and the intra-key segment editing and inter-key segment optimization algorithm in Section 4. Evaluation of our method with real data is given in Section 5. Finally, we summarize the proposed method in Section 6.

2. Related Work

2.1. Motion editing work for computer animation

The proliferation of various techniques for creating motions for visual media products, coupled with the needs to best utilize these artistic assets, has made motion editing a rapidly developing area of research and development. The past few years have witnessed an explosion of new techniques for various motion editing tasks, as well as deployment of commercial tools.

For editing a single object, motion warping [1, 2] has been widely applied to preserve the original features of input motion data while satisfying a new set of constraints as much as possible. Using structurally similar motion database, Mukai *et al*. [3] have proposed a motion interpolation approach to obtain continuous spans of high quality motion. Some other researchers combined an interpolation technique with motion graphs to allow parametric interpolation of motion while traversing the graph [4, 5]. While these works are proved to work robustly for editing single object motions, they did not provide any constraints for editing multi-party interaction events.

Besides, Zordan *et al*. [6] have introduced a technique for incorporating unexpected impacts into a motion capture-driven animation system through the combination of a physical simulation and a specialized search routine. Ye *et al*. [7] have proposed a fully automatic method that learns a nonlinear probabilistic model of dynamic responses from very few perturbed sequences. Their model is able to synthesize responses and recovery motions under new perturbations different from those in the training examples. Although these methods can somehow, work for very short interactive events, they are performing the editing by changing the original motion dynamics instead of keeping it, and they are not applicable for editing long and continuous multi-party interaction scenes either.

In this paper, we are presenting a novel spatiotemporal editing framework that synthesize well synchronized multi-party interaction scenes from separately captured data, in which objects are performing interactive motions that should match with each other, respectively.

2.2. Motion representation approaches

In computer graphics and computer vision area, researchers have invented various approaches for describing motions.

Positions and velocities of human body parts have been used by Green *et al*. [8] for human movement recognition. Optical flows [9], motion templates [10] and space-time volumes [11, 12] are also widely used for solving tracking and recognition problems. Such methods are mainly used for describing the objects motion features for recognition tasks, and they do not offer any controllable factors for motion editing task. Besides, kinematic models [6, 13] are widely used in robotics, motion capture system and computer animation editing. They can represent variety kind of motions, and as well they provide easily controllable factors for the editors. However, the kinematic models are lacking in the ability of representing multi-party interaction events. Few spatiotemporal constraints can be directly defined between multiple objects based on them. Not to mention that for certain type of unstructured data [14] without a unified

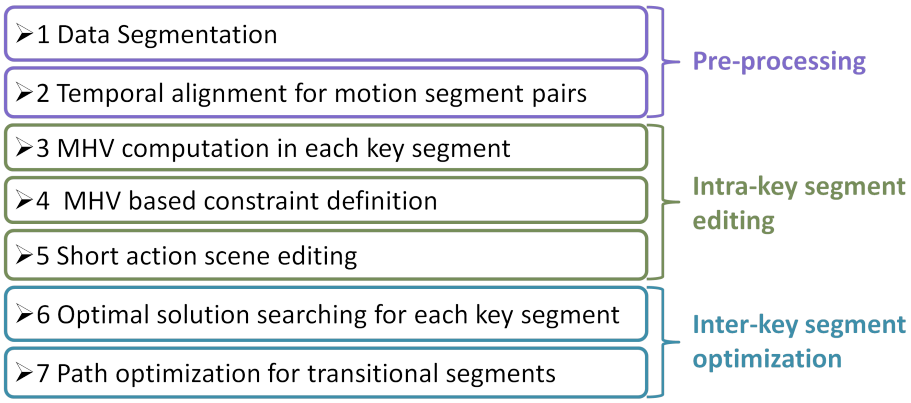


Figure 2. Computational processes for multi-party interaction editing.

mesh model, kinematic structures are not applicable at all.

On the other hand, the idea of Motion History Volume has been proposed by Weinland [15], for solving free viewpoint action recognition task. It carries out a concept of considering the entire motion as a whole instead of describing it for each single frame, by encoding the history of motion occurrences in the volume. Voxels are therefore multiple-values encoding how recently motion occurred at a voxel. Mathematically, consider the binary-valued function $D(x, y, z, t)$ indicating motion at time t and location (x, y, z), then their MHV function is defined as follows:

$$v_\tau(x,y,z,t) = \begin{cases} \tau & if D(x,y,z,t) \\ max(0, v_\tau(x,y,z,t-1)-1) & otherwise \end{cases} \quad (1)$$

where τ is the maximum duration a motion is stored.

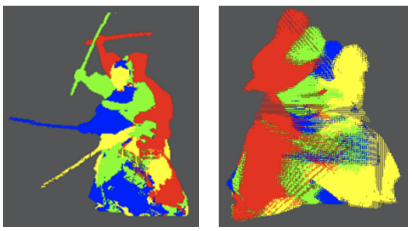
Our work proposed an augmentation of the original MHV by recording precisely the full temporal information of motion, making it more suitable for multi-party interaction editing tasks.

3. Definition of aMHV and multi-party interaction dictionary

In this section, we first propose an augmentation of the original Motion History Volume by adding in full temporal information of the motion. Then an aMHV based multi-party interaction representation method will be given as well.

3.1. aMHV of individual motion

As is written in Section 2.2, the idea of motion history volume was invented for action recognition, and has been proved to be effective. However, the original definition of MHV only cares about the occurrence of the motion, which may not be enough for performing multi-party interaction editing. Since we need to do spatial and temporal alignments of multiple separately captured motion sequences,



(a) aMHV of the attacker. (b) aMHV of the dodger.

Figure 3. Examples of aMHV.

full temporal information including both the starting and ending moments of the motion is necessary for representing the spatiotemporal correlations between multiple objects motions. Therefore, we extend the definition of MHV by adding in more temporal information of the motion as follows:

$$v_\tau(x,y,z,t) = \begin{cases} (t_{start}, t_{end}) & if \bigcup_{i=0}^{\tau-1} D(x,y,z,i) \\ Null & otherwise \end{cases} \quad (2)$$

where t_{start} and t_{end} are the starting and ending time of the motion. And we denote the center of gravity of aMHV $v^A_{\tau_i}(x, y, z, t)$ as $C^A_i(x, y, z)$, which is considered as the root node of an aMHV.

Figure 3 illustrates the aMHVs of the attacker and the dodger in a fighting scene, respectively. Here different colors represent different ending time of the motion on the voxels.

It should be noted that aMHV needs not necessarily contain the motion of the entire body. Instead it can be simplified by only counting in the partial volume of interest on the objects body. For example, in a sword fighting scene we may only care about the weapon of the attacker, so that an aMHV of the sword would be enough for further editing work.

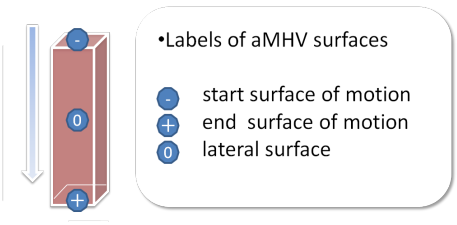


Figure 4. Lables of aMHV surfaces.

3.2. Multi-party interaction dictionary using aMHVs

Using the aMHV as is described in Section 3.1, we can represent the spatiotemporal structure of a single objects motion. However, in order to perform effective editing of multi-party interaction events, we still need to find a way to model the spatiotemporal correlations of multiple objects motions (which is represented with aMHV). To realize that, we assign certain labels onto the aMHV surfaces according to the motion directions to help describe the spatiotemporal relationship of multiple aMHVs. As is shown in Figure 4, we denote the starting surface of aMHV with ⊕, the ending surface with ⊖, and the lateral surface with ⓪. Then, let a pair of labels denote the contacting relationship of two aMHVs. For example, ⊕ & ⊕ means the two objects aMHVs contact with each other at the ending surfaces.

Based on the combination of those labels, we can describe all the possible spatiotemporal correlations of multiple aMHVs, as is illustrated in Table 1. We name the sum up of them a multi-party interaction dictionary, which represents all the possible multi-party interaction types that could be modeled using aMHV based representation.

In details, the spatiotemporal constraints for each interaction type are described as follows:

First of all, for all interaction types there is a spatial constraint that the 3D volume of two aMHVs should contact with each other:

$$V_\tau^A(x,y,z) \cap V_\tau^B(x,y,z) \neq \phi \tag{3}$$

here $V_\tau^A(x,y,z)$ represents a collection of voxels in $v_\tau^A(x,y,z,t)$. Note that the two paired aMHVs should have the same maximum duration τ, we will explain how we ensure this in Section 4. As well, the recorded timing should be scaled into the segment oriented timing by subtracting its real frame numbers from the original motion sequence.

Second, each interaction type has its own temporal constraint on all the contacting voxels as follows:

⊕ & ⊖:

$$t_{end}^A = T_{end}, t_{start}^B = T_{start} \tag{4}$$

⓪ & ⓪:

(a) If interest volumes contact with each other at every moment within τ

$$\bigcap_{t=0}^{\tau-1} (V_t'^A(x,y,z) \cap V_t'^B(x,y,z) \neq \phi) \tag{5}$$

(b) If interest volumes contact with each other at certain moments within τ

$$\bigcup_{t=0}^{\tau-1} (V_t'^A(x,y,z) \cap V_t'^B(x,y,z) \neq \phi) \tag{6}$$

(c) If interest volumes have no contacts throughout τ

$$t_{start}^A > t_{end}^B \tag{7}$$

⊕ & ⊕:

$$t_{end}^A = t_{end}^B = T_{end} \tag{8}$$

⊖ & ⊖:

$$t_{start}^A = t_{start}^B = T_{start} \tag{9}$$

⊕ & ⓪:

$$t_{end}^A = T_{end} \tag{10}$$

⊖ & ⓪:

$$t_{start}^A = T_{start} \tag{11}$$

here $T_{start} = 0$ and $T_{end} = \tau$. $V_t'^A(x,y,z)$ represents a subset of $V_\tau^A(x,y,z)$ at time t.

This multi-party interaction dictionary provides us various spatiotemporal constraints for performing the editing work. For example, an attack and guard scene in sword fighting can be counted as type ⊕ & ⊕. Then on the interested part of the two aMHVs, we require $t_{end}^A = t_{end}^B = T_{end}$, meaning that the two swords contact and only contact at the end of the whole motion duration.

4. Spatiotemporal 3D editing algorithm

In this section we present the spatiotemporal 3D editing algorithm based on aMHV.

4.1. Overview and definition of terms

In this research work, the expected input data is a series of 3D mesh surfaces of the objects, be that a unified mesh model or frame-wise unstructured meshes. And we suppose a motion sequence can be considered as a collection of key segments and transitional segments. For each short action scene formed up by a pair of key segments, there should exist multiple reasonable solutions. While a unique optimal solution for the entire interaction event could finally be found in the integration throughout the whole sequence.

Generally, our multi-party interaction editing method consists of three steps as (1) data segmentation and temporal alignment, (2) intra-key segment editing by aMHV-based

Table 1. Interaction Dictionary with examples.

Interaction Dictionary	⊕ & ⊖	⓪ & ⓪	⊕ & ⊕	⊖ & ⊖	⊕ & ⓪	⊖ & ⓪
Hand shaking		Shake Hands	Raise Arm	Arm Down		
Fighting	Push&Pull	Attack&Dodge	Attack&Guard	After A&G	Punch	After Punch

constraint satisfaction and (3) inter-key segment global optimization. In order to make it more understandable, we first give clear definitions of these editing-related terms as follows:

1. Motion sequence: The whole sequence of the original captured objects, in the form of 3D surface meshes. Let $M = \{V, E\}$ denote the 3D mesh, then a motion sequence can be denoted as $S_i = \{M_k | k = 1, \ldots, n_k\}$. For our multi-party interaction editing, motion sequences of different objects will serve as the input.
2. Key segment: In each motion sequence, frames where interactive motions ($e.g.$ handshaking, fighting, ...) happen are considered to be key segment $K_i = \{M_k | k = 1, \ldots, n_k\}$. Since the objects are supposed to perform motions that match with each other, key segments should appear in pairs from different motion sequences.
3. Transitional segment: Intermediate frames between key segments in the motion sequence, in which no interactive motion is stored. We use $T_i = \{M_k | k = 1, \ldots, n_k\}$ to denote transitional segments, and it should be noted that there can be no transitional segment between two key segments.
4. Action scene: Spatiotemporally synchronized short multi-party interaction scene synthesized from a single pair of key segments.
5. Interaction sequence: Spatiotemporally synchronized long multi-party interaction sequence synthesized by integrating all the key segment pairs and transitional segments in the original motion sequences.

4.2. Data segmentation and temporal alignment

In this research work, since our main focus is on editing multi-party interaction scenes instead of recognizing them, the data segmentation is performed manually by the editor. Below are three criteria for selecting key segments:

(1) Interactive motions should happen in each key segment, so that spatial and temporal constraints as defined in the interaction dictionary exist within each key segment pair.

(2) The motion trend of the interested volume should be unidirectional, which will ensure the simplicity of aMHV in each key segment.

(3) Multiple key segments in one motion sequence should have no overlaps, and they need not to be adjacent.

On the other hand, since the potential interactively corresponding motions in the original captured data would inevitably have temporal mismatches, the candidates in each selected motion segment pair may also have different lengths. Therefore, a temporal normalization is needed for building up comparable aMHVs as follows:

For a pair of key segment K_i^A and K_i^B,
if $\tau_i^A > \tau_i^B$,

$$M'^A_{i_m} = M^A_{i_n} \quad (0 < m < \tau_i^B, n = \left\lfloor m \times \frac{\tau_i^A}{\tau_i^B} \right\rfloor) \quad (12)$$

else,

$$M'^B_{i_m} = M^B_{i_n} \quad (0 < m < \tau_i^A, n = \left\lfloor m \times \frac{\tau_i^A}{\tau_i^B} \right\rfloor) \quad (13)$$

where τ_i^A and τ_i^B represent the duration of K_i^A and K_i^B, $M^A_{i_n}$ represents the nth frame in K_i^A, and $M'^A_{i_m}$ represents the mth frame of the normalized K_i^A. $\lfloor x \rfloor$ is the floor function that returns the largest integer not greater than x. Note that the same processing will be applied to the transitional segment pairs as well.

4.3. Intra-key segment editing

4.3.1 Editor defined constraints

After data segmentation and temporal alignment, the interactive motions of each object are organized into key segment pairs, and inside each pair the two motions are scaled into the same length. Then the aMHVs of each object in each segment pair can be computed as $v^A_{\tau_i}(x, y, z, t)$ and $v^B_{\tau_i}(x, y, z, t)$. For each interaction event inside a key segment pair, the two interacting objects should follow certain spatiotemporal constraints based on (1) interaction type, (2) mutual visibility and (3) fidelity requirement. First for each key segment pair the interaction type should be decided by the user based on his/her editing intension, producing the spatiotemporal constraint as is described in Section 3.2.

Second, for multi-party interaction scenes, naturally the objects should be visible for each other. We use the method of Shi *et al.* [16] to compute the objects average gazing direction during the editor-assigned frames. Then a visual cone is generated for each object as $VS_i^A(x, y, z)$ and $VS_i^B(x, y, z)$, looking from the average 3D position of the center of the eyes into the average gazing direction. The detailed parameters (e.g. the cone angles) can be adjusted

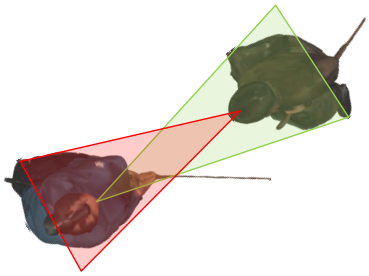


Figure 5. Mutual visibility constraint.

by the editor as appropriate. Then the mutual visibility constraint can be described as follows:

$$(VS_i^A(x,y,z) \cap V_{\tau_i}^B(x,y,z)) \cap (VS_i^B(x,y,z) \cap V_{\tau_i}^A(x,y,z)) \neq \phi \quad (14)$$

It should be noted that the visibility constraint is an optional constraint for the editor, meaning that it is not always required for each moment in the key segment pair.

Third, the fidelity requirement avoids the unnatural fakeness, like two object merge into each other's body, from happening. Let $nv_{\tau_i}^A(x,y,z,t)$ and $nv_{\tau_i}^B(x,y,z,t)$ denote the aMHVs of the objects that should not contact in the interaction event, then the fidelity constraint can be defined as follows:

$$\bigcup_{t_i=0}^{\tau_i-1} (nV_{t_i}'^A(x,y,z) \cap nV_{t_i}'^B(x,y,z)) = \phi \quad (15)$$

4.3.2 Action scene editing using constraint satisfaction

With the editor defined constraints, we compute the proper relative position of the objects by constraint satisfaction.

First, the two objects are put into the same world coordinate system. We fix the position of $v_{\tau_i}^A(x,y,z,y,t)$ and then, sample the xy plane into grid points, and each grid point will be further sampled by surrounding angles.

Next, translate and rotate $v_{\tau_i}^B(x,y,z,t)$ onto each sampled position and direction, then test with the editor defined constraints. If for a sampled position and direction all the constraints are fulfilled, then the distance between the two objects $L_i^{AB} = |\overrightarrow{C_i^A C_i'^B}|$ with the two reference angles θ_i^A and θ_i^B can be counted as one solution for this key segment pair. Here $C_i'^B$ denotes the relocated position of C_i^B in one sampled position, and θ_i^A, θ_i^B represents the included angles between the average viewing direction of each aMHV and line $C_i^A C_i'^B$, respectively. By testing all the sampled positions and directions, for each key segment pair a solution group $R_i^{AB} = (L_{i_j}^{AB}, \theta_{i_j}^A, \theta_{i_j}^B)$ will be computed.

4.4. Inter-key segment optimization

Having computed the solution groups for each key segment pair, we will combine them together with the transitional segments to synthesize a complete multi-party interaction sequence, by (1) finding the optimal solution in each key segment pair and (2) adding in the transitional segments and performing path optimization.

4.4.1 Optimal solution searching for key segments

Before editing, suppose the positions of the aMHVs' root node in the world coordinate system are $C_1^A, C_2^A, \ldots, C_I^A, C_1^B, C_2^B, \ldots, C_I^B$, and the facing directions of the aMHVs in the world coordinate system are $\overrightarrow{F_1^A}, \overrightarrow{F_2^A}, \ldots, \overrightarrow{F_I^A}, \overrightarrow{F_1^B}, \overrightarrow{F_2^B}, \ldots, \overrightarrow{F_I^B}$. In order to make natural and smooth editing, we should maintain the vectors $\overrightarrow{C_i^A C_{i+1}^A}$ and $\overrightarrow{C_i^B C_{i+1}^B}$ as much as possible to minimize the artificial offsets. As well, the facing directions $\overrightarrow{F_i^A}$, $\overrightarrow{F_i^B}$ should also be preserved. If we denote the edited ideal positions of the aMHVs' root node as $\bar{C_1^A}, \bar{C_2^A}, \ldots, \bar{C_I^A}, \bar{C_1^B}, \bar{C_2^B}, \ldots, \bar{C_I^B}$, the edited facing directions of the aMHVs as $\overrightarrow{\bar{F_1^A}}, \overrightarrow{\bar{F_2^A}}, \ldots, \overrightarrow{\bar{F_I^A}}, \overrightarrow{\bar{F_1^B}}, \overrightarrow{\bar{F_2^B}}, \ldots, \overrightarrow{\bar{F_I^B}}$, and the angles between $\overrightarrow{\bar{F_i^A}}$ and $\overrightarrow{\bar{C_i^A}\bar{C_i^B}}$ as $\bar{\theta_i^A}$, then we should search for the optimal solution for each key segment pair by fulfilling:

$$min \bigcup_{i=1}^{I-1} [|\overrightarrow{C_i^A C_{i+1}^A} - \overrightarrow{\bar{C_i^A}\bar{C_{i+1}^A}}|^2 + |\overrightarrow{C_i^B C_{i+1}^B} - \overrightarrow{\bar{C_i^B}\bar{C_{i+1}^B}}|^2 + \lambda(|\overrightarrow{F_i^A} - \overrightarrow{\bar{F_i^A}}|^2 + |\overrightarrow{F_i^B} - \overrightarrow{\bar{F_i^B}}|^2)] \quad (16)$$

and

$$|\overrightarrow{\bar{C_i^A}\bar{C_i^B}}| = L_i^{\bar{A}\bar{B}} = L_{i_j}^A B = |\overrightarrow{C_{i_j}^A C_{i_j}'^B}| \quad (17)$$

$$\bar{\theta_i^A} = \theta_i^A, \bar{\theta_i^B} = \theta_i^B \quad (18)$$

here λ is a weighting factor to balance the translation and rotation transformations. The optimal solution for all the key segments can be computed by solving function (16), (17) and (18) using dynamic programming approach.

4.4.2 Path optimization for transitional segments

Having found the optimal solutions for all the key segments, we then add in the transitional segments and perform path optimization. Suppose inside a transitional segment, the original positions of the object's center of masses for all the frames are $C_1, C_2, \ldots, C_{m-1}$ and the edited positions are denoted as $\bar{C_1}, \bar{C_2}, \ldots, \bar{C_{m-1}}$. While original and edited facing directions are denoted respectively as

$\overrightarrow{F_1}, \overrightarrow{F_2}, \ldots, \overrightarrow{F_{m-1}}$ and $\overrightarrow{\bar{F}_1}, \overrightarrow{\bar{F}_2}, \ldots, \overrightarrow{\bar{F}_{m-1}}$. Then we define our objective function for performing path optimization as follows:

$$f = \sum_{i=1}^{m-1} [\delta f_a(\bar{C}_i) + (1-\delta)\omega_i^C f_v(\bar{C}_i) + \lambda(\delta f_a(\overrightarrow{\bar{F}_i}) + (1-\delta)\omega_i^F f_v(\overrightarrow{\bar{F}_i})] \quad (19)$$

Specifically, $f_a(\bar{C}_i)$ is defined as follows to preserve the original accelerations for each frame:

$$f_a(\bar{C}_i) = |(C_{i-1} - 2C_i + C_{i+1}) - (\bar{C}_{i-1} - 2\bar{C}_i + \bar{C}_{i+1})|^2 \quad (20)$$

and $f_v(\bar{C}_i)$ is defined as follows to preserve the original speed for each frame:

$$f_v(\bar{C}_i) = |\frac{C_{i-1} - C_{i+1}}{2} - \frac{\bar{C}_{i-1} - \bar{C}_{i+1}}{2}|^2 \quad (21)$$

Besides,

$$\omega_i^C = \frac{1}{1 + exp\{\alpha(v_i - v_k)\}} \quad (22)$$

$$\omega_i^F = \frac{1}{1 + exp\{\alpha(r_i - r_k)\}} \quad (23)$$

$$v_i = \frac{|C_{i-1} - C_i| + |C_i - C_{i+1}|}{2} \quad (24)$$

$$r_i = \frac{|\overrightarrow{F_{i-1}} - \overrightarrow{\bar{F}_i}| + |\overrightarrow{\bar{F}_i} - \overrightarrow{F_{i+1}}|}{2} \quad (25)$$

here v_k and r_k are constant parameters to be set by editors. By minimizing this objective function we can perform path optimization onto all the transitional segments and finally, acquire a natural looking spatiotemporally synchronized multi-party interaction 3D Video sequence for free-viewpoint browsing.

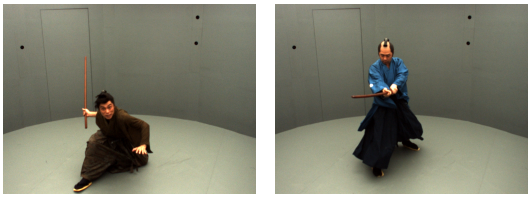
5. Experiment and evaluation

In this section we present an evaluation of the proposed multi-party interaction editing method with real data.

5.1. Experiment setup and pre-processing

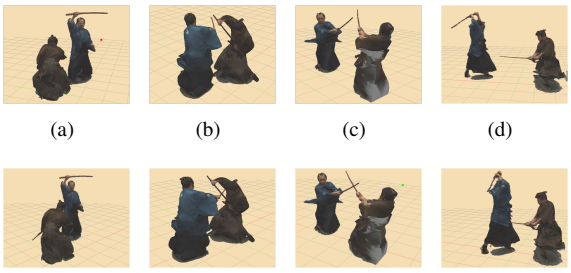
To prove the effectiveness of our method, we prepared the 3D video data of two professional actors, separately captured by 15 calibrated XGA cameras running at 25 Hz with 1 msec shutter speed, and reconstructed with frame-wise 3D shape reconstruction method. In the data, the two separately captured actors are performing a pre-designed sword fighting motion sequence, respectively. Note that the reconstructed 3D shapes do not have a unified 3D mesh model. Body part segmentation and kinematic models are not available either.

As a pre-processing, the data is manually segmented into 9 key-segment pairs and inside each pair the motions of the two objects are aligned into the same temporal length. After the pre-processing, the lengths of both motion sequences are resampled into 425 frames.



(a) Image data of object A. (b) Image data of object B.

Figure 6. Separately captured image data of two objects.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. Motion editing results for short action scenes.

5.2. Multi-party interaction editing results

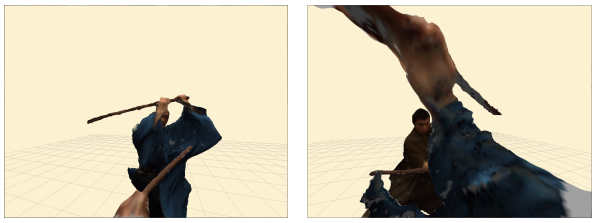
With the separately captured data, if we simply put them together into the same coordinate system and match up their beginning frames manually, in the following motion sequences various spatiotemporal mismatches will occur, as is shown in Figure 7(a)-(d). On the other hand, the editing results using the proposed method are illustrated in Figure 7(e)-(h). It can be clarified that in each synthesized interaction scene, the relative location of the two objects looks reasonable and they well qualify the editor defined spatiotemporal constraints.

In addition, Figure 8 illustrate the first-person-view images rendered from the editing results, using Shi *et al.*'s method[16]. It can be seen clearly that the two objects are inside each other's view, fulfilling the mutual visibility constraint.

5.3. Processing cost of the proposed method

The experiment is conducted using a PC with 2 Intel(R) Core(TM)2 Duo CPU, E6850 @ 3.00GHz. Under the sampling resolution of 10mm, 2 degree, the average computation time for each automatic process is as follows:

1. aMHV computation for each object in a single segment: 1 min
2. Constraint satisfaction for each short action scene: 5 mins



(a) First-person-view image of object A. (b) First-person-view image of object B.

Figure 8. First-person-view images rendered from the editing result.

3. Optimal solution searching for each key segment pair: 98 mins
4. Path optimization for each object in a single transitional segment: 7 mins
5. Total time cost: 287 mins

Note that the computations for each segment can be conducted parallelly inside process 1, 2, 4.

6. Conclusion

In this paper we presented a novel motion editing method that synthesizes spatiotemporally synchronized multi-party interaction 3D Video scenes from separately captured data. By proposing the idea of aMHV we can effectively model the interaction events of multiple objects as well as representing the motions of a single object, based on which the constraint satisfaction method can be applied to perform intra-key segment editing. An optimal solution search and path optimization scheme is also designed to minimize the artifacts generated in the editing and maintain the original motion dynamics as much as possible. Experiments with real data proved the effectiveness and robustness of the proposed method.

In our work, the data being used is the unstructured 3D Video data. However, since our method only requires a sequence of 3D meshes as input, it will work for (Motion Capture Driven) CG data, which has a unified mesh model, as well. For further studies, we are planning to perform the same editing job using CG data with unified mesh models, to examine how we can combine our spatiotemporal editing method with conventional kinematic structure based methods together to acquire better results.

Acknowledgement

This study is supported by JSPS Ayame project ”Visual Gesture Perception”.

References

[1] Gleicher, M.. Retargetting motion to new characters. SIGGRAPH 98: Proceedings of the 25th annual conferenceon Computer graphics and interactive techniques (1998), ACM, pp. 3342. 2.

[2] Lee, J., Shin, S.Y.. A hierarchical approach to interactive motion editing for human-like figures. Proceedings of SIGGRAPH 99 (1999), pp. 3948. 2.

[3] Mukai, T., Kuriyama, S.. Geostatistical motion interpolation. ACM Transactions on Graphics 24, 3 (2005), 10621070. 2.

[4] Heck, R., Gleicher, M.. Parametric motion graphs. Proceedings of Symposium on Interactive 3D Graphics and Games (2007), pp. 129136. 2.

[5] Safonova, A., Hodgins, J.K.. Construction and optimal search of interpolated motion graphs. ACM Transactions on Graphics 26, 3 (2007), 106. 2.

[6] Zordan, V.B.. Dynamic response for motion capture animation. ACM Trans. Graph., 24, 3, pp. 697-701(2005).

[7] Ye, Y. and Liu, C.K.. Synthesis of responsive motion using a dynamic model. Comput. Graph. Forum, 29, 2, pp. 555-562 (2010).

[8] Green, R. and Guan, L.. Quantifying and recognizing human movement patterns from monocular video images. IEEE transaction on Circuits and Systems for Video Technology, 14, February 2004.

[9] Alexei, A.E., Alexander, B., Greg, M., and Jitendra, M.. Recognizing action at a distance. ICCV, 2003.

[10] Aaron, F.B. and James, W.D.. The recognition of human movement using temporal templates. IEEE Trans. Pattern Anal. Mach. Intell., 23(3):257-267, 2001.

[11] Syeda-Mahmood, T.F., Vasilescu, M.A.O. and Sethi, S.. Recognition action events from multiple viewpoints. EventVideo01, pages 6472, 2001.

[12] Yilmaz, A. and Shah, M.. Actions sketch: A novel action representation. CVPR05, pages I: 984.989, 2005.

[13] Hilton, A.. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. International Journal on Computer Vision and Image Understanding, Vol.104, No.2-3 :90-127, 2006.

[14] Nobuhara, S. and Matsuyama, T.. Deformable Mesh Model for Complex Multi-Object 3D Motion Estimation from Multi-Viewpoint Video. The Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT2006), 2006.6.

[15] Weinland, D., Ronfard, R. and Boyer E.. Free viewpoint action recognition using motion history volumes. COMPUTER VISION AND IMAGE UNDERSTANDING (2006).

[16] Shi, Q., Nobuhara, S. and Matsuyama, T.. 3D Face Reconstruction and Gaze Estimation from Multi-view Video using Symmetry Prior. IPSJ Transactions on Computer Vision and Applications, Vol.4, pp.149-160, 2012.10.1.