Interaction-based Human Activity Comparison

Yijun Shen, Longzhi Yang, Edmond S. L. Ho, and Hubert P. H. Shum

Abstract—Traditional methods for motion comparison consider features from individual characters. However, the semantic meaning of many human activities is usually defined by the interaction between them, such as a high-five interaction of two characters. There is little success in adapting interaction-based features in activity comparison, as they either do not have a fixed topology or are in high dimensional. In this paper, we propose a unified framework for activity comparison from the interaction point of view. Our new metric evaluates the similarity of interaction by adapting the Earth Mover's Distance onto a customized geometric mesh structure that represents spatial-temporal interactions. This allows us to compare different classes of interactions and discover their intrinsic semantic similarity. We created five interaction databases of different natures, covering both two-characters (synthetic and real-people) and character-object interactions, which are open for public uses. We demonstrate how the proposed metric aligns well with the semantic meaning of the interaction. We also apply the metric in interaction retrieval and show how it outperforms existing ones. The proposed method can be used for unsupervised activity detection in monitoring systems and activity retrieval in smart animation systems.

Index Terms—Activity Comparison, Interaction, Human Motion Analysis, Distance Metric, Earth Mover's Distance

1 Introduction

Comparing human activities is a core problem in areas such as sports sciences, rehabilitation and monitoring. Applications in these areas typically require the user to perform a set of pre-defined activities and evaluate the correctness/quality by comparing the performed activities with given exemplars. On one hand, traditional motion analysis methods typically require the type of the activities to be known in advance in order to apply the right criteria for evaluations, and can only evaluate the similarity of activities belonging to the same class. On the other hand, traditional motion classification methods work well in identifying different classes of activities, but fall short in analyzing the subtle difference for those belonging to the same class. This paper aims at proposing a new unified metric that accurately evaluates both intra-class and inter-class similarity.

Existing research in the field mainly analyzes the motion of individual characters only, without considering the interaction among characters and that between the character and the environment. The geometric features extracted from individual characters are limited in modelling the semantic meaning of complex movements such as boxing and dancing. They cannot distinguish semantically dissimilar interactions that have similar geometrically features. For example, a high-five interaction between two characters is similar to a waving interaction if we look at the features of the individual characters. Similarly, they cannot identify the similar semantic meaning from geometrically different interactions, such as a right punch having some level of similarity to a left punch when they both hit the opponent.

We observed that high-level activities are usually defined based on character-character or character-environment interactions, such as punching an opponent, sitting on a chair and jumping over a fence. The contextual meaning of the activity depends heavily on the interaction instead of individual movement [1]. For example, a punching movement that hits is semantically different from the same punch that misses. This motivates us to research on a metric that evaluates the similarity of activities based on the concept of interaction.

Interaction-based features are therefore considered to solve the problem. However, many of them suffer from various limitations. While relative kinematic features such as the joint relative distance are used to model movement between characters [2], the number of feature increases exponentially with the number of considered joints, and it becomes inefficient to use a high dimensional feature vector for representing the interaction involving two characters. A feature selection pre-process can be introduced, but there is a side effect that the optimally selected features depend on the types of interactions. Logical filters are efficient in indexing and modelling the motion of character using multiple manually defined logical rules [1]. However, for two or more characters, there will be an exponential number of possible logical rules, and manually defining the optimal rules requires domain experts' knowledge. The Gauss Linkage Integral (GLI) that represents the degree of twisting between two

- *Y. Shen is with the Faculty of Engineering and Environment, Northumbria University, UK. Email: yi.shen@northumbria.ac.uk*
- *L. Yang is with the Faculty of Engineering and Environment, Northumbria University, UK. Email: longzhi.yang@northumbria.ac.uk*
- *E. S. L. Ho is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. Email: edmond@comp.hkbu.edu.hk*
- *H. P. H. Shum, the corresponding author, is with the Faculty of Engineering and Environment, Northumbria University, UK. Email: hubert.shum@northumbria.ac.uk*

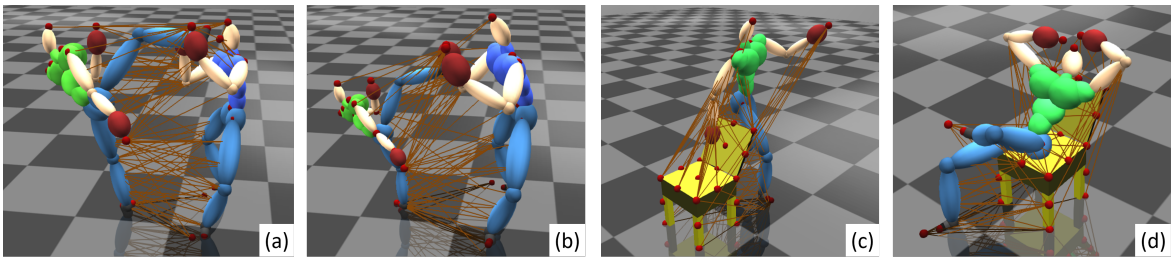

Fig. 1: Our interaction model (brown lines) effectively represents the information accounting for the high-level semantic meaning of human activities, including (a) character-character interaction, (b) character-character interaction with different body sizes, (c) non-contacting and (d) contact-based human-object interaction.

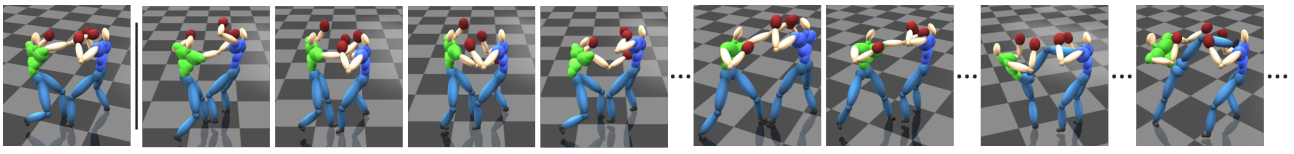

Fig. 2: Our unified framework can compare different classes of interactions and discover their intrinsic similarity. Using a "right punch + being hit" interaction as a reference, the most similar interactions are variations of "right punch + being hit" (rank 1-4). Notice that the punch in rank 3 and 4 hit the upper body instead of the head as in the reference, resulted in a slightly higher difference. "left punch + being hit" (rank 14-15) are ranked lower, which have a similar semantic meaning of punches that hit. "kick + being hit" (rank 31 and 40) appears in even lower ranks, which have a similar semantic meaning of attacks that hit, while the right kicks is more similar than the left ones.

strings can be applied in analyzing two characters interactions [3]. Since it models human body as simplified strings and twisting degree indicates close-body interactions, it cannot effectively represent non-contacting interactions. This group of interactions is important as it covers a large range of interactions, such as one character avoid a punch from another, a character walking around another, and two characters talking. Overall, these interaction-based features either suffer from the problem of exponentially growing dimension size or perform optimally only for limited types of interactions.

In this paper, we propose a new metric for evaluating the degree of similarity between interactions by adapting Earth Mover's Distance [4] onto a customized interaction mesh structure [5] that represents spatial-temporal interactions. Given a scenario, we extract feature points from the characters and/or the environment objects and construct an interaction mesh structure using Delaunay Tetrahedralization [6]. Such a three-dimensional mesh can be robustly applied in different human activities, as shown in Fig. 1, and effectively samples spatial proximity into a lower dimensional space for efficient processing. We establish correspondences between topologically different structures from different interactions, such that we can evaluate the similarity of interactions between different classes. This is facilitated by the Earth Mover's Distance (EMD) [4], which has shown great success in corresponding meshes of different objects [7]. Instead of using a distance function of simple vertex coordinates that cannot capture directional information, or discrete topological distance that cannot produce continuous distance values [8], we propose a distance function consisting of direction and position for effective interaction comparison, and demonstrate that it works very well with the correspondence produced by EMD.

A strength of our system is that it can compare different classes of interactions and evaluate the subtle similarity. This is particularly important in activity analysis applications in which we do not have full prior knowledge of the activity classes. Our proposed metric is continuous, meaning that we can compare any two interactions and evaluate their similarity. Also, our proposed metric aligns well with the semantic meaning of the interactions comparing with existing ones. As shown in Fig. 2, "right punch + being hit" (reference) and "right kick + being hit" (rank 31) are usually considered to be different classes of interactions, but they are similar as they are both "attack + being hit" interactions. Our system can assess the level of similarity between them and discover their intrinsic correspondence. As a result, we can arrange different classes of interactions on a continuous scale of similarity, and perform content-based interaction retrieval.

While our system perform very well in inter-class

comparison, its true value is the accuracy in intra-class activity comparison. Comparing the subtle difference among activities in the same class is essential in areas such as sports sciences and rehabilitation, in which the user performed activity is being compared with a set of exemplars in order to mimic the correct moves. Notice that many activities involve interaction with opponents (e.g. dancing and boxing) or environment objects (e.g. hurdling and ball games). Again, as shown in Fig. 2, our system can tell the subtle difference between a punch that hit the head (reference, rank 1 and 2) and a punch that hit the upper body (rank 3 and 4) using the interaction features.

Good inter and intra-class interaction comparisons facilitate many applications. For example, in a monitoring system, listing all possible types of interactions that can happen is typically difficult. With inter-class comparison, the system can deduce that a pushing interaction is somewhat similar to a fighting one, thereby understanding it as a dangerous activity. Intra-class comparison enables it to assess the potential injury level of a fighting interaction based on the annotated exemplars in the database. Similarly, the inter-class comparison allows a smart animation system to suggest a "hook punch and hit" interaction to the animator, when the query "straight punch and hit" is not available in the database. The intra-class comparison facilitates finding out the best matches.

We have three major contributions in this paper:

- We propose a new framework to compare human interactions including human-human and human-object interactions. We demonstrate that the evaluation of both intra-class and inter-class distance aligns well with the semantic similarity.
- Utilizing the proposed interaction-based metric, we implement an interaction comparison system and a content-based interaction retrieval system. Such systems perform robustly in different types of human activities.
- We construct five interaction databases that are open to the research community for benchmarking. This is one of the first comprehensive databases containing different classes of human-human and human-object interaction in the community.

The rest of the paper is organized as follows. Section 2 reviews the related research of this project. Section 3 details the construction of five interaction databases of different natures. Section 4 explains our framework to compare interactions. Section 5 shows our experimental results. Section 6 concludes the paper and provides discussions.

2 Related Work

In this section, we first review traditional human-centered representations for human motions and discuss their major weaknesses. Then, we review interaction-based representations and point out the difficulties of applying them in motion retrieval and analysis.

2.1 Human-centered Systems

There is a large body of research about analyzing and identifying human motion using human-centered features of body movement. In the early research of human motion retrieval, traditional approaches utilize kinematic features such as joints position [9] and joint angles-based distance [10] to evaluate different types of motion. Dynamic features such as forces produced by specific joints provide another mean to identify human movement [11]. Derived dynamic features such as center of pressure can enhance body stability analysis [12].

Although it is possible to analyze individual kinematic and dynamic features, understanding the logical significance of a motion requires the meaningful combination of them. Logical rules based on combined kinematic features can be used as the motion features in motion retrieval [1]. By exploiting the body hierarchy, kinematic features concerning body parts can provide a higher level of evaluation [13]. Movement notation language known as the Laban notation can abstract a short duration movement [14].

To better reflect the semantic meaning of human motion and minimize the tedious manual design, machine learning algorithms based on joint-pair relationship features are introduced to train classification systems that recognize different types of motion [15], [16]. Learning a distance metric based on a set of single-character posture feature improves motion recognition accuracy [17]. Neural networks can automatically learn a manifold to represent a motion [18]. Deep learning algorithms such as the convolutional autoencoders can learn effectively from a large amount of data [19], [20].

While these human-centered representations have been effective for interpreting basic movement, they fall short in representing scenarios involving multiple interacting characters, which is one of the key components in daily activities.

2.2 Interaction-based Systems

Recently, there has been a significant increase in research to analyze the interaction between multiple characters. Long durations of human-environment interactions can be segmented as patches [21]. This motivates us to focus our evaluation on shorter duration of interactions. An event graph structure is proposed to represent multi-character interactions for motion synthesis [22]. However, such an abstract model does not carry enough low-level information for detailed activity comparisons. An even more abstracted model, the motion grammar, is more human-understandable for representing high-level activities [23]. Instead of

an abstract, human-friendly interaction representation for human-computer interfaces, we design a metric that better reflects the semantic meaning of the interactions. The Laplacian space is used to maintain the spatial-temporal constraint when editing multi-character interactions [24]. We employ a similar technique to synthesize more interactions in our CRC database.

Relative kinematic features such as relative joint distance are proposed to represent the interaction between two characters [2]. The concept of kinematic-based logical rules can also be extended to represent inter-character kinematic features [1]. However, the feature dimension increases exponentially when considering multiple characters. While feature selection [2], [15] can be used to maintain a reasonable feature dimension, the optimal set of the feature depends on the type of interactions. It is difficult to find a global optimal set of low dimensional feature to represent all interaction types.

By considering the skeleton hierarchy of the interacting character as a number of strings, the *Gauss Linking Integral* is used to represent how these strings wrap around each other, thereby representing the interaction of the two characters [3]. Such a representation can be used to synthesize movement by considering close interaction [25], as well as motion indexing and retrieval [26]. However, the representation cannot effectively represent non-contacting interactions such as one character avoiding an attack from another.

The *interaction mesh* has been shown to be a robust interaction representation [5]. It considers the joints of the interacting characters, and applies Delaunay Tetrahedralization [6] to generate a mesh structure that indicates spatial proximity. Using the interaction mesh, interaction among characters can be adapted according to the user-defined criteria or environment changes [5], [27]. The structure is used in robotics to represent the interaction between a robot and the environment for movement adoption [12] and control [28].

There are some attempts to apply interaction mesh in interaction retrieval [29], [30], [31]. However, the results are not satisfying. The major difficulties are that the topology and dimension of the interaction mesh depend on the postures of the interacting characters, and therefore changes across different classes of interaction and across frames, making it difficult to compute the difference between two interaction meshes. Previous works attempt to solve the problem by dividing the distance function into two parts. For the edges that co-exist in two interaction meshes, a traditional geometry-based distance function is applied. For those that do not co-exist due to the topological difference, [30] assumes zero distance, while [29] simply counts the total number. Since the two parts of the distance function have different natures, forcing them together generates inconsistent results. [31] utilizes an affinity matrix calculated based on a heuristic to extract the active joint pairs, but the heuristic requires domain knowledge and is likely dependent on the types of interaction.

Database	Nature	No. of Interactions	Duration (sec)
2C	Synthetic	95	206
CRC	Synthetic	60	130
HOI	Real-People	30	200
2PB	Real-People	44	103
2PD	Real-People	29	170

TABLE 1: Details of the databases constructed.

In this project, we propose a new unified framework for interaction-based activity comparison. We adapt the interaction mesh structure due to its robustness, and we correspond two topologically different interaction meshes with the Earth Mover's Distance [4]. Our method can discover the intrinsic similarity between interactions, and produce superior results compared with existing work.

3 Interaction Databases

To demonstrate the robustness of our proposed system, we construct five different databases involving different types of interactions. The number of interactions and respective durations of the database are shown in Table 1.

3.1 Character-Character (2C)

We created a character-character (2C) database using kick-boxing motions. Kick-boxing was chosen as it involves a large variety of movements and is considered to be one of the most challenging domains of human motion research [32].

We adapt the interaction synthesis framework proposed in [32] to synthesize high-quality interactions. The major advantages of such an approach are that we can guarantee the availability of data for a wide variety of interaction classes, and categorize the data with synthesizing parameters. To synthesize interactions, first, we capture the shadow boxing of a single boxer and construct an *action level motion graph* [33]. Second, we define a set of semantic interaction classes, each defines the interaction pattern [34] to be performed the characters. Third, we perform the *temporal tree expansion* to synthesize the interactions between two characters using a set of reward functions [32], and extract the interactions that fit into our pre-defined list of interaction classes.

The complete list of semantic interaction classes is shown in Table 2. The labels of basic kick-boxing moves are borrowed from [35], in which high-intensity moves are classified into punches, kicks and defense (i.e. avoid in our case). Such basic moves are then combined to form the list of semantic interaction classes. Designing such a list requires domain

Interaction Type	Attacking Type	Attacking Body Part	Class
A Attacks, B Avoids	Punch	Left Punch	A1.1
		Right Punch	A1.2
	Kick	Left Kick	A2.1
		Right Kick	A2.2
A Attacks, B Being Hit	Punch	Left Punch	A3.1
		Right Punch	A3.2
	Kick	Left Kick	A4.1
		Right Kick	A4.2

TABLE 2: 2C & CRC: Hierarchical semantic classes.

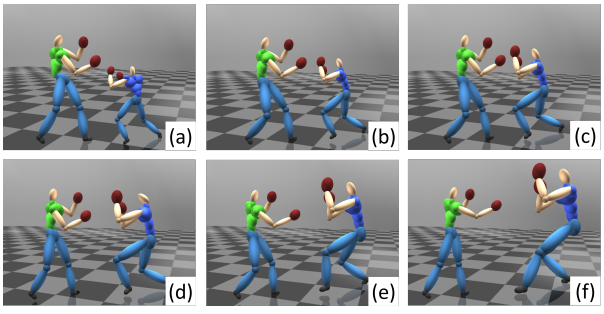

Fig. 3: CRC database: For the same interaction, the size of the blue character is scaled by (a) 80%, (b) 90%, (c) 100%, (d) 110%, (e) 120%, and (f) 130%.

knowledge, and is more of an art than a science. Our strategy is to enumerate different combinations of common boxing interaction by first deciding the outcome of the interaction (i.e. attack avoided or attack hit). This is because whether an attack is hit or avoided forms the most significant context in sports such as boxing. We then list the attacking type of the attacking character (i.e. punch or kick), and then further describe the lower-level details of the attack (left or right).

3.2 Character-Retargeted Character (CRC)

In order to evaluate the robustness of our method to different interactions with the same context, as well as its robustness against geometry changes, we also create a character-retargeted character (CRC) database. In such a database, we adjust the size of a character but maintaining the nature of the interaction.

The database is created by first synthesizing interactions with the method mentioned above. We then resize one character into 80% to 130% of the original size in every 10% step. The scale range is designed according to [5]. It suggests that such a range is effective for interaction retargeting without changing contact information, which is important for the interaction context. We finally retarget the movement using Autodesk MotionBuilder, a third-party software that provides a user interface for maintaining contacts during retargeting with inverse kinematics [36]. An example frame of retargeted interaction is shown in 3. Table 2 shows the semantic classes defined.

Interaction Type	Spatial Variations	Class
Walking-around	From the Back	B1.1
	Stepping Over	B1.2
	At the Front	B1.3
Sitting-on	Forwards	B2.1
	Sideway	B2.2

TABLE 3: HOI: Hierarchical semantic classes.

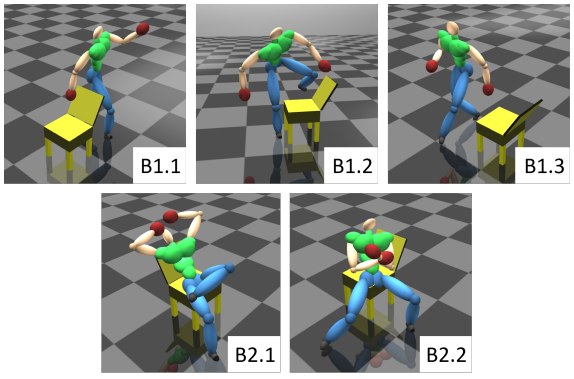

Fig. 4: HOI: The five class of human-object interactions corresponding to Table 3.

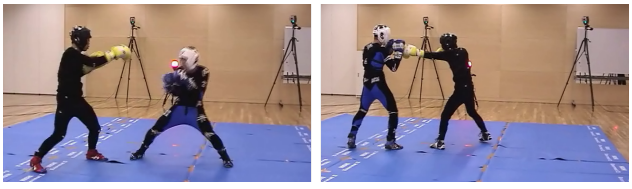

Fig. 5: 2PB: Motion capture for real-people boxing.

3.3 Human-Object Interaction (HOI)

We further created a human-object interaction (HOI) database to demonstrate how our method can be used in a more general context. We use a chair as the object since it has a complex structure and multiple ways to interact with, such as sitting on and walking around.

This database is constructed by capturing human motion in an environment with a chair of known dimensions and positions. We model the chair with boxes manually based on the real-world dimensions obtained. Table 3 specifies the classes of interaction we included. We first define 2 types of interactions (i.e. walking-around and sitting-on). We then define a number of spatial variations (e.g. from the back, stepping over, at the front) for each of the types. Example interactions of each class are shown in Fig. 4.

3.4 2 People Boxing (2PB)

To evaluate the performance of our system for real-people interactions, we created a database of boxing motion performed by 2 people (2PB), which was captured at the University of Tokyo. This is a challenging database with complex interactions and body movement.

Interaction Type	Movement Variations	Class
A and B Attack at the Same Time	With a Single Punch	C1.1
	With Combo Punches	C1.2
A Attacks, B Avoids	B Avoids Only	C2.1
	B Avoids and Counter-attacks	C2.2

TABLE 4: 2PB: Hierarchical semantic classes.

Interaction Type	Class
A and B Walk Around in a Circular Manner	D1
A and B Dance Together	D2
A and B Shake Hands	D3
A and B Chat with Each Other	D4

TABLE 5: 2PD: Semantic classes.

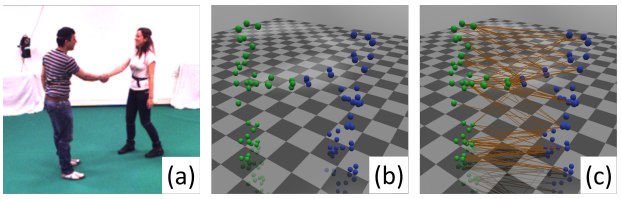

Fig. 6: 2PD: (a) Motion capture, (b) the corresponding point clouds, and (c) the interaction mesh.

We collect around 6 minutes of boxing from 4 pairs of professional boxers, as shown in Fig. 5. We define the semantic classes that covered the majority of the boxing movements as shown in Table 4. The interaction classes defined here are generally more complicated than that of the 2C database, in the sense that the actions from the real boxers are less synchronized (e.g. both attacking in similar timings) and more continuous (e.g. longer combo punches).

3.5 2 People Daily Interaction (2PD)

We created a real-people database of 2-people daily interactions (2PD). This is based on the Utrecht Multi-Person Motion (UMPM) benchmark [37].

The original dataset contains multi-person daily interactions such as walking around each other and shaking hands. We consider only 2 people interactions in the scene and define 4 semantic classes of commonly occurred characteristic interactions, as shown in Table 4. Unlike the previous databases we mentioned above, this database is presented in a C3D surface point cloud format instead of a skeletal representation. We consider each C3D point as a joint when generating the interaction mesh structure. Fig. 6 shows a frame in the video footage, the corresponding point clouds and the interaction mesh respectively.

4 Unified Interaction Comparison

In this section, we explain our unified framework for interaction comparison, which involves three major components. First, we elaborate the approach to represent an interaction sequence using a series of customized interaction meshes. Then, with the help of the Earth Mover's Distance, we propose a distance function to evaluate the difference between two interaction meshes. Finally, we deal with the spatial and temporal variations using normalization and sampling respectively, and align two sequences with Dynamic Time Warping (DTW).

4.1 Customized Interaction Mesh Structure

Here, we explain how we adapt the interaction mesh structure [5] to represent the interaction between characters. Without loss of generality, we explain our system using two characters interactions. We then explain how the system can be applied to human-object interactions.

Given two characters interacting with each other, we utilize the interaction mesh structure as a feature representation, as it can gather the implicit spatial relationship of the character effectively. Considering one frame of an interaction, an interaction mesh is created by generating a volumetric mesh using Delaunay Tetrahedralization [6], considering the 3D Cartesian joint positions of the interacting characters as vertices. An interaction is therefore represented by a series of interaction meshes. The topology and the dimension of the meshes vary over time according to the changing poses of the characters, which allows representing the varying spatial relationship over time.

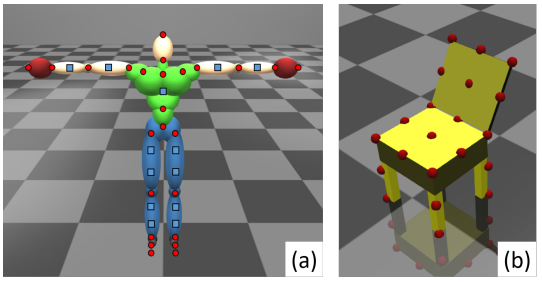

Fig. 7: Sampled vertices on (a) a character and (b) an environment object.

We customize the process to generate the interaction mesh [38] such that the resultant mesh is more suitable for interaction comparison. In particular, we would like to have a uniform distribution of vertices to ensure that the comparison is not biased to body parts with more joints. Therefore, on top of the set of vertices generated by the joint positions of the characters in [38], we include a set of vertices by uniformly sampling the skeleton structure of the characters using a predefined length. This allows us to maintain a more uniform density for the mesh, such that the interaction comparison based on the mesh is not biased to specific joints. In our implementation, a character consists of 25 joints, which are shown as the red circles in Fig. 7a. We uniformly sample body segments using a

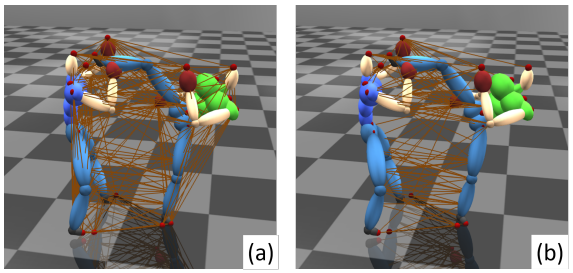

Fig. 8: Interaction mesh creation: (a) edges from Delaunay Tetrahedralization (b) edges after filtering.

sampling length of $15cm$. This process creates another 13 vertices, which are shown as the blue squares in Fig. 7a. We found that further reducing the sampling length leads to similar results but came with higher computational overhead due to the more complicated interaction mesh generated, and therefore chose the mentioned sampling length.

To create the interaction mesh, we consider frame t of an interaction between two characters, and denote $\mathbf{V}^t$ as the set of vertices of the characters:

$$\mathbf{E}^t_{DT} = DT(\mathbf{V}^t), \tag{1}$$

where DT is the Delaunay Tetrahedralization process, and $\mathbf{E}^t_{DT}$ is the set of edges created. Different from [38] that considers all edges, we filter $\mathbf{E}^t_{DT}$ by removing all edges connecting to the same character, as those edges do not contribute to the interaction. The resultant set of edges, $\mathbf{E}^t$, is regarded as the interaction mesh of frame t. The brown lines in Fig. 8 show the edges before and after filtering.

Finally, the temporal sequence of an interaction is represented as a series of interaction meshes, $\mathbf{E} \in \left\{\mathbf{E}^0, \mathbf{E}^1, \cdots, \mathbf{E}^{t_{total}}\right\}$, where t_{total} is the total number of frames in the interaction.

We utilize the same algorithm to create the interaction meshes for human-object interactions, but we consider the object as the second character. In particular, we approximate the object using boxes. Similar to the skeleton sampling algorithm as explained above, we uniformly sample the surface of the boxes using a predefined sampling distance, which is set as $20cm$ in our experiments. Fig. 7b shows an example of vertices sampled from a chair. Finally, we combine the vertices from the character and the object to generate the interaction mesh, as shown in Fig. 1c and d.

For the 2PD database, the motions are represented with C3D surface marker points and the body hierarchy information is not given explicitly. Therefore, we do not conduct the sampling process, which is done according to the body hierarchy. Also, for some body parts, there are not enough surface markers to effectively calculate the joint angles using inverse kinematics. Therefore, we consider each point as a joint when generating the interaction mesh structure, and use the point cloud as a query in the activity comparison experiments.

Existing work [1], [2] and fully-connected meshes (i.e. connecting all vertices with edges) [39] suffer from the exponentially growing size of the feature. For example, if we extract one feature based on one vertex pairs for two characters of 38 vertices each, there will be $(38 \times 2) \times (38 \times 2 - 1) = 5700$ features. To tackle this issue, we utilize Delaunay Tetrahedralization to sample spatial proximity, and prune edges connecting to the same character. On average, each interaction mesh for two characters consists of only 170 edges.

4.2 Distance between Interaction Meshes

One of the major features of our interaction representation structure is that it can represent semantically dissimilar interactions using the topologically and dimensionally varying interaction meshes, thanks to the use of Delaunay Tetrahedralization in evaluating geometry proximity. This allows us to effectively represent interactions of different semantic meaning (e.g. punching vs. kicking) using a consistent format. Therefore, unlike previous research, our algorithm allows the comparison of two interactions with different semantic meaning, and thereby find out if they have any intrinsic similarity. To achieve this, we propose a distance function that adapts the Earth Mover's Distance (EMD) [4] to find the best correspondence between the input interaction meshes. Such a distance function can effectively compare interaction mesh of different topologies and dimensions.

Here, we explain how to compute the distance between two interaction meshes of two-character interactions. The same distance function is used for human-object interaction, by considering the environment object as the second character.

4.2.1 *Edge-Level Distance Function*

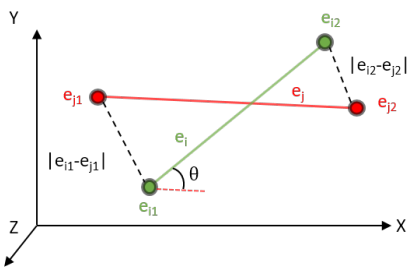

Fig. 9: The discance between two edges.

Given edge e_i from interaction i and edge e_j from interaction j, we represent the difference between the two edges using a customized cosine distance function, which effectively combines the Euclidean distance and orientation distance between the two

edges. It is defined as:

$$d(e_i, e_j) = (|e_{i1} - e_{j1}| + |e_{i2} - e_{j2}|) \times \frac{1}{2}(1 - \cos\theta), \quad (2)$$

where $|*|$ denotes Euclidean distance, e_{i1} and e_{i2} are the two endpoints of e_i connecting characters 1 and 2, e_{j1} and e_{j2} are that of e_j, θ is the angle between the two edges, and $\cos\theta$ is calculated by vector dot product. The idea of the equation is visualized in Fig. 9. The cosine term is multiplied by $\frac{1}{2}$ such that it has a range of $[0.0, 1.0]$. Compared with other designs, such as the weighted sum of distances and cosine angles, ours does not require any parameter tuning.

4.2.2 Earth Mover's Distance

We then adapt a mass transport solver [4] to find the optimal edge-level correspondence between two interaction meshes. The idea is to match the edges by minimizing the overall sum of the distance of all the edges. Given two sequences of interaction meshes $\mathbf{E_I}$ and $\mathbf{E_J}$, let us consider one interaction mesh $\mathbf{E}_I^{t_I} \in \mathbf{E}_I$ at frame t_I and one interaction mesh $\mathbf{E}_J^{t_J} \in \mathbf{E}_J$ at frame t_J. The mass transport solver optimizes a set of unidirectional flow to map the edges $e_i \in \mathbf{E}_I^{t_I}$ to $e_j \in \mathbf{E}_J^{t_J}$ with a minimized overall distance:

$$f_{i,j}^* = \arg\min_{f_{i,j}} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} d(e_i, e_j) f_{i,j} \right), \quad (3)$$

subjected to:

$$\sum_{j=1} f_{i,j} = 1.0, \quad (4)$$

$$\sum_{i=1} f_{i,j} = \frac{n}{m}, \quad (5)$$

where m and n are the total number of edges in the mesh $\mathbf{E}_I^{t_I}$ and $\mathbf{E}_J^{t_J}$ respectively, $d(e_i, e_j)$ is the distance between two edges calculated with Eq. 2, $f_{i,j}$ is the set of flow values to be optimized. The constraint in Eq. 4 ensures that in case an edge is mapped into multiple ones, the sum of all outgoing flows is always 1.0. The constraint in Eq. 5 ensures that the sum of all incoming flows to an edge is a constant. These equations jointly guarantee that all edges in $\mathbf{E}_I^{t_I}$ map to all edges in $\mathbf{E}_J^{t_J}$ evenly.

With the optimal set of flow values $f_{i,j}^*$, the minimum distance between two interaction meshes is calculated as:

$$D(\mathbf{E}_I^{t_I}, \mathbf{E}_J^{t_J}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d(e_i, e_j) f_{i,j}^*. \quad (6)$$

Fig. 10 visualizes the concept of the mass transport solver in two simplified 2D scenarios, in which the red mesh is matched onto the green one. The flow to match the two meshes is represented by the black arrows, while the corresponding number is the magnitude of the flow. Fig. 10a is a simpler case in which both meshes have the same number of edges, and a solution of one-to-one mapping can be achieved. On the other hand, in Fig. 10b, since the red mesh has more edges than the green one, some of the edges in the red mesh match partially to those in the green one.

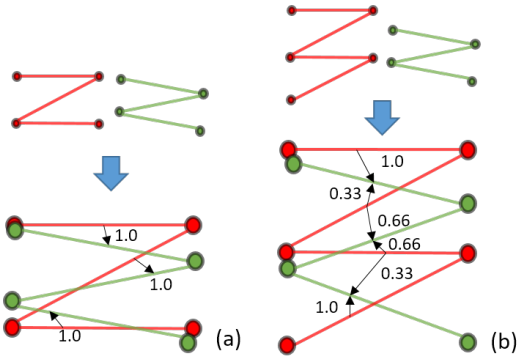

Fig. 10: The concept of mass transport solver in 2D.

Finally, the EMD is calculated as the normalized minimal distance. With EMD, the distance between two meshes, which are usually topologically and dimensionally different, can be calculated:

$$EMD(\mathbf{E}_I^{t_I}, \mathbf{E}_J^{t_J}) = \frac{D(\mathbf{E}_I^{t_I}, \mathbf{E}_J^{t_J})}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{i,j}^*}. \quad (7)$$

4.3 Distance between Interaction Sequences

Here, we explain how to evaluate the distance between two sequences of interaction.

4.3.1 Spatial Normalization

We observe that real human compares interactions with little consideration on their absolute position and orientation. For example, two kicking interactions happening in different positions and facing directions are typically considered to be similar. Therefore, we normalize interactions spatially to compare them with local coordinates, thereby eliminating the influence from different world coordinates.

In general, there are two strategies to do the spatial alignment. The first strategy assumes that the interacting characters have unique identities. One therefore consistently uses the same character in different interactions as a reference to normalize the whole time series of interactions, by removing its pelvis translation and its horizontal facing angle in each frame. The second strategy assumes that the two characters are anonymous. One can then obtain two normalized results by considering either of them as the reference. In this case, when comparing interactions, both normalized results are evaluated and the one that generates the smaller difference is selected.

In our research, we opt for the first strategy since the logical meaning of interactions in movies and games usually depends on the unique identities of the

characters. For example, "a hero kicking a monster" is different from "a monster kicking a hero".

4.3.2 Temporal Sampling

We also observe that real human is sensitive to characteristic features of the interaction instead of its duration. For example, the duration of a sitting down motion is not very important in defining its context. Motivated by [40], we design a non-linear sampling algorithm to obtain a set of keyframes that better represents the context of the motion. We aim at extracting the significant postures from the data point of view rather than the human perception point of view [41], although there are similarities in both applications.

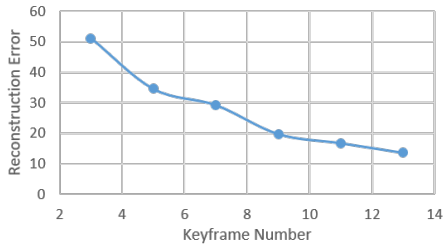

Fig. 11: Reconstruction error against keyframe number.

Here, given one sequence of interaction mesh, we first consider each frame as a block. Then, we go through all the neighbouring block pairs. Starting from the pair with the least distance calculated by Eq. 7, we merge the pair into a single block. If there is more than one frame in a block, the distance is represented by the maximum distance of all frame combinations. We repeat the process to further merge the block pairs until the number of blocks equals to the required number of the keyframes. The center of each block is then considered as a keyframe. We set the required number of keyframe as 9, which is determined by analyzing the average reconstruction error using different keyframes, as shown in Fig. 11.

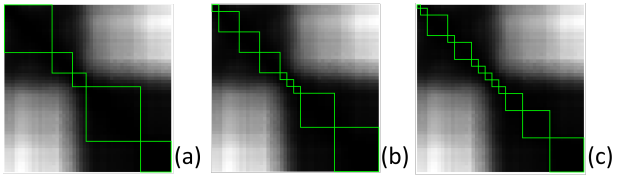

Fig. 12: The results of keyframes sampled overlaid on a self-similarity matrix with (a) 5, (b) 9, and (c) 13 keyframes. Green squares represent keyframe regions, and darker pixels indicates smaller distance.

Fig. 12 shows the sampling results using different keyframe numbers. Here, the self-similarity matrix of an interaction calculated with Eq. 7 is shown, with a darker color representing smaller distance. The green squares represent the sampled blocks. The sampling algorithm samples more keyframes in the regions where the distance changes rapidly across frames, which is typically the frame range with complex interaction. It samples fewer in bigger regions with high similarity, which contribute less to the context of the interaction.

4.3.3 Temporal Alignment

Finally, similar to [40], we align the keyframes of two interactions using Dynamic Time Warping (DTW) and calculate the distance between them.

Given the keyframe sequences of interaction meshes, $\mathbf{E}_I$ and $\mathbf{E}_J$, and each of them is represented by W keyframes, we obtain an optimal warping path $\mathbf{p} = [(p_{I1}, p_{J1}), (p_{I2}, p_{J2}), \cdots, (p_{IW}, p_{JW})]$ to align the two keyframe sequences. Using such a path, for each $w \in [0, W]$, the interaction mesh $\mathbf{E}_I^{p_{Iw}} \in \mathbf{E}_I$ at keyframe p_{Iw} is aligned with $\mathbf{E}_J^{p_{Jw}} \in \mathbf{E}_J$ at keyframe p_{Jw}. Therefore, the DTW distance is defined as:

$$DTW(\mathbf{E}_I, \mathbf{E}_J) = \frac{1}{W} \sum_{w=1}^{W} EMD(\mathbf{E}_I^{p_{Iw}}, \mathbf{E}_J^{p_{Jw}}). \quad (8)$$

5 Experimental Results

In this section, we evaluate the performance of our system. We compare our method with an interaction-based feature known as *space-time proximity graphs* [29], as well as traditional human-centered features including joint positions [9] and joint angles [10]. We first evaluate the performance in comparing and evaluating distances between interactions using similarity matrices. We then analyse the quality and the accuracy on interaction retrieval using precision and recall analysis, as well as an interactive retrieval applications with user-defined constrains.

For [9] and [10], we normalize the interaction in the same way as our method according to Section 4.3.1. We utilize all joint information from the characters to form the feature vector. For the object in the HOI database, we represent the object as a set of position for [9], and as a static simplified skeleton for [10].

The experiments were performed on a computer with dual Intel Xeon E5-2687W CPUs, an NVIDIA Quadro K4000 display card and 64GB RAM. Extracting the interaction meshes and sampling the keyframe are performed as a pre-process. The computational time depended mainly on the number of samples in the database and the duration of the interaction. The pre-process took 1.5 hours, 0.5 hour and 4 hours for the 2C, CRC and HOI databases respectively. Given the meshes, computing the distance between two interactions took 0.2 seconds on average.

Our interaction database is open for public usage at our website. Also, please refer to the attached video for more results such as the quantitative retrieval analysis.

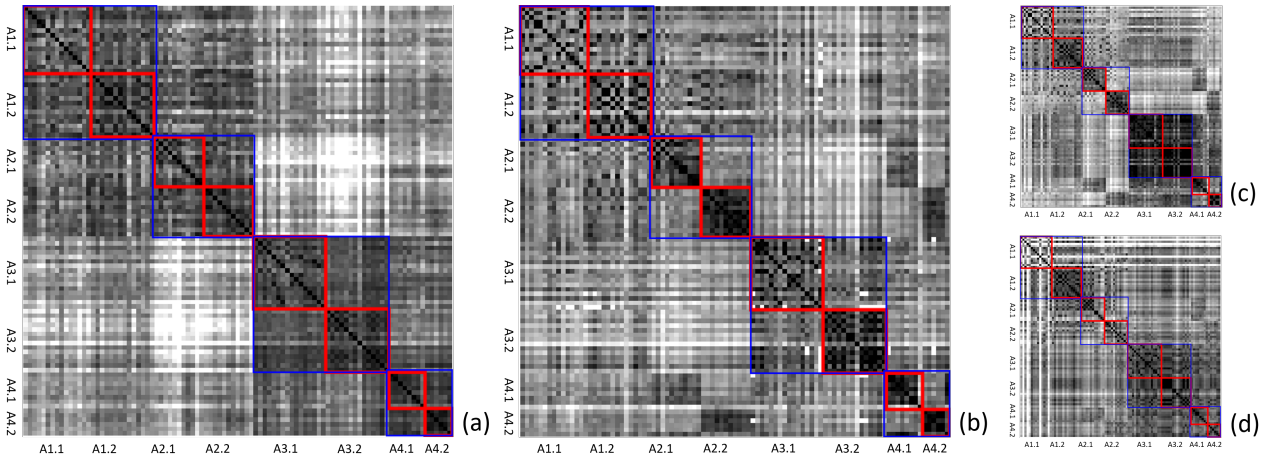

Fig. 13: 2C: Similarity matrices evaluated by (a) our method, (b) [29], (c) [9], (d) [10].

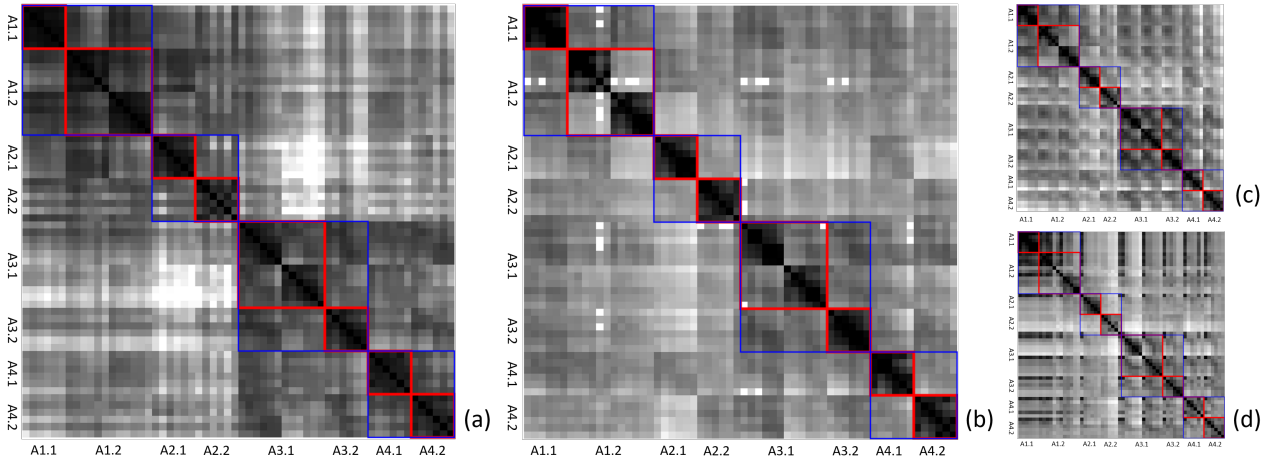

Fig. 14: CRC: Similarity matrices evaluated by (a) our method, (b) [29], (c) [9], (d) [10].

5.1 Interaction Similarity Analysis

Here, by analyzing the similarity matrices, we evaluate the quality of the method with three key criteria:

- high intra-class similarity, to find out interactions of similar context
- high inter-class difference, to distinguish interactions of different context
- different levels of inter-class similarity according to the semantic similarity, to effectively evaluate *how different* two interactions are

The last criterion is usually overlooked in existing works. Typical supervised machine learning methods for classification can create very high intra-class similarity and inter-class difference, but there is little continuous evaluation of difference for pairs that are different to a certain extent.

Fig. 13 to 17 show the similarity matrices of all comparing methods in different databases, in which each pixel shows the similarity between two interactions. The pixel color represents the normalized distance between two interactions, and the value 3σ (i.e. standard derivation) of each method is used as the normalizer of the respective matrix. Darker pixels represent higher similarity. We arrange the interaction according to interaction classes defined in Table 2 and Table 3, which are marked in the X and Y axes. We also highlight in the matrix the square areas that belong to the different levels of class similarity using blue and red lines for a better observation.

The similarity matrix of our method in the 2C database is shown in Fig. 13a. Each individual class (highlighted by red squares) shows the highest intra-class similarity. Classes belonging to the same attacking type (highlighted by blue squares) shows the second highest similarity. Classes belonging to the same interaction type (i.e. A1.1-A2.2 and A3.1-A4.2) shows moderate similarity. Interactions of different interaction types are generally different, but if they have the same attacking type or the same attacking body part, the difference is smaller. This demonstrates how our method fulfils the three criteria mentioned above.

Comparing to [29] in Fig. 13b, our method performs better in intra-class similarities, such as A1 and A3, in which one character punches the other. Our method

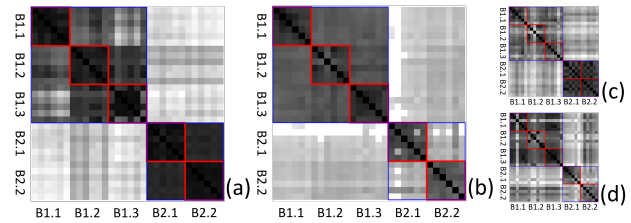

Fig. 15: HOI: Similarity matrices evaluated by (a) our method, (b) [29], (c) [9], (d) [10].

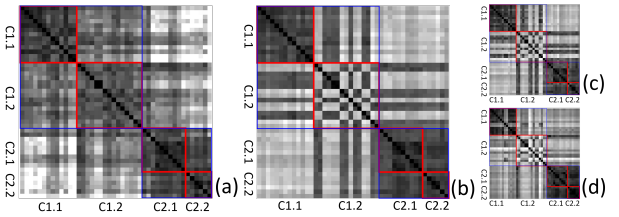

Fig. 16: 2PB: Similarity matrices evaluated by (a) our method, (b) [29], (c) [9], (d) [10].

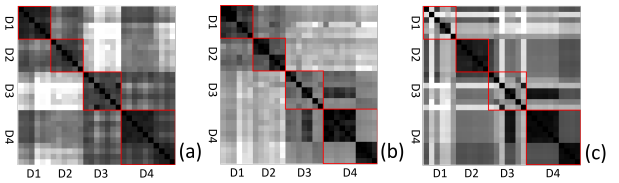

Fig. 17: 2PD: Similarity matrices evaluated by (a) our method, (b) [29], (c) [9].

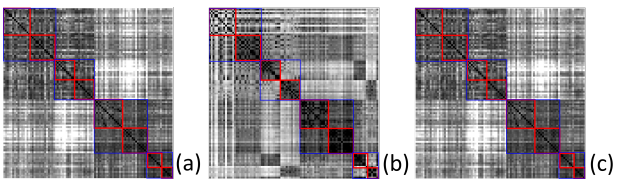

Fig. 18: 2C: Similarity matrices evaluated by (a) our method, (b) our method without pruning self-connecting edges, (c) our method with an alternative keyframe selection strategy.

also outperforms [29] in identifying motion classes with the same attacking type (highlighted by blue squares) - there is a large distance between A2.1 and A2.2, as well as between A4.1 and A4.2. This is mainly because the distance function in [29] involves a topology distance term that counts the edges with different vertices. Such a term is sensitive to small changes of interaction and does not align well with human understanding. Human-centered features including [9] in Fig. 13c and [10] in Fig. 13d do not work well. Due to the lack of interaction information, both of them show a high inter-class distance between the semantically similar classes within each blue square, as well as a small inter-class distance. They fail to distinguish if a character is hit or not.

The results of the CRC database are shown in Fig. 14. This is a challenging database as the interactions are edited during the retargeting process, which results in different 3D postures. Still, our proposed method generates a high intra-class similarity, and there are different levels of similarity aligning with the semantical meaning. This supports the robustness of our method in evaluating the interaction of different character sizes. [29] also results in a high intra-class similarity, but it fails to highlight the semantic closeness in this database indicated by the red and blue squares. Both [9] and [10] struggle in identifying the similarity within each class, suffering from the difference in postures after motion retargeting.

The results of the HOI database are shown in Fig. 15. Our method has a high intra-class similarity indicated by the red squares, and a reasonable similarity for the classes of the same interaction type indicated by the blue squares. [29] shows a less significant intra-class similarity indicated by the red squares. Also, the semantic closeness between B2.1 and B2.2 cannot be identified. [9] and [10] cannot clearly differentiates B1.1, B1.2 and B1.3. [10] further cannot identify the similarity between B2.1 and B2.2.

The results of the 2PB database are shown in Fig. 16. Due to the complex, ambiguous real-people motion, [9], [10], [29] fail to identify the intra-class similarity accurately, especially for C1.2 in which two boxers perform multiple punches simultaneously. These methods also fail to identify the intrinsic similarity among the two sub-classes in C1.x, for which our method performs consistently.

The results of the 2PD database are shown in Fig. 17. Notice that this database is extracted from a public database where real-people motions are represented as point clouds. Therefore, joint angle [10] evaluation is not available. Since [29] employ a binary function to determine the topology difference, it is very sensitive to small gesture changes, which results in the poor intra-class similarity in all four classes.

Finally, we include experiments in the 2C database to evaluate the effect some design strategies in our system, as shown in Fig. 18. First, our system removes edges connecting to the same character such that the evaluation focus on the interaction between the characters. Fig. 18a (ours) and b (ours without pruning) show that keeping self-connected edges results in poor performance in identifying the similarity in semantically similar classes, such as A1.x and A2.x. Second, for keyframe selection in Fig. 12, while we select the middle frame of each block, other selection algorithms also work as well. Fig. 18c shows the result of an alternative selection strategy, which selects the last frame in a block instead of the middle frame. There is no significant difference comparing to our method in Fig. 18a.

5.2 Interaction Retrieval Analysis

Here, we implement an interaction-based retrieval system. Given one interaction, we apply Eq. 8 to evaluate its difference with respect to all motion in the database, and retrieve the most similar ones across all interaction classes. Fig. 2 shows the retrieved results of using a right punch and hit interaction as the query (i.e. A3.2 in Table 2), annotated with the corresponding ranks and differences. The advantage of our system is that it can compare different types of interactions and discover their intrinsic similarity.

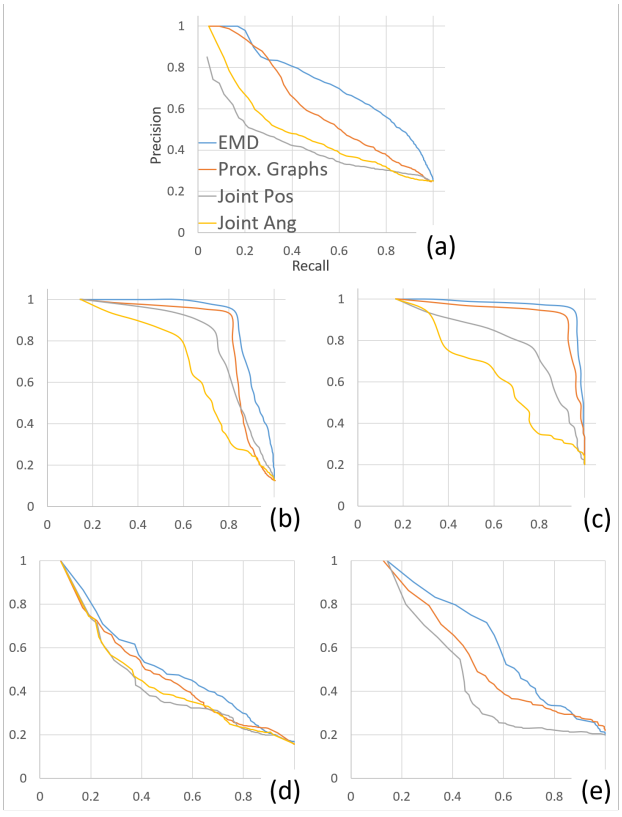

Fig. 19: Precision and recall of all methods for (a) 2C, (b) CRC, (C) HOI, (d) 2PB and (e) 2PD.

We compare the methods using precision and recall [42] as shown in Fig. 19. We treat each interaction in the database as a query, and average the results from all queries to form the plot. Given a query interaction, only the retrieved results within the same semantic lowest level sub-class (e.g. A1.1) as defined in Table 2 (for 2C and CRC) and Table 3 (for HOI) are considered as relevant results. Our method outperforms the others in all five databases.

We further evaluate how our system performs for different types of interactions. We use the 2C database here as it has a large interaction variety. We group the interaction classes according to the attacking types in Table 2. Fig. 20 shows the precision and recall results. It can be seen that in general, avoiding interactions (i.e. A1.x and A2.x) are more challenging, potentially due to the large variety of avoiding actions. Our method outperforms [29] in general in A1.x, but it does not have a clear advantage over [29] in A2.x. A4.x is a kick and being hit interaction. Due to the farther attacking position comparing to punches, such a class has a larger intra-class spatial variety. As a result, [9] and [10] perform particularly poorly. We also plot the precision and recall results of the real-human database 2PB in Fig. 21. It shows that our method performs consistently better than the others in both classes C1.x and C2.x.

In order to evaluate the accuracy and consistency of interaction retrievals using our system, Table 6 shows the numerical values of the average matching retrieval results in different ranges of ranks. The second column shows the accuracy of the exact matching class. From the second column to the right-most one,

Retrieved Rank	Matching Class	Matching Attacking Type & Interaction Type	Matching Interaction Types	All Others
1-5	92.5%	7.5%	0.0%	0.0%
6-10	71.7%	18.8%	9.5%	0.0%
11-15	30.4%	43.1%	20.9%	5.6%
16-20	16.3%	52.4%	23.6%	7.7%
21-25	0.0%	38.7%	53.2%	8.1%
26-30	0.0%	36.3%	52.9%	10.8%
31-35	0.0%	19.2%	66.1%	14.7%
36-40	0.0%	8.9%	74.5%	16.6%
41-45	0.0%	0.0%	78.1%	21.9%
46-50	0.0%	0.0%	76.7%	23.3%

Retrieved Rank	Matching Class	Matching Attacking Type & Interaction Type	Matching Interaction Types	All Others
1-5	95.0%	5.0%	0.0%	0.0%
6-10	42.5%	55%	2.5%	0.0%
11-15	12.6%	69.7%	10.3%	7.4%
16-20	0.0%	41.1%	47.2%	11.7%
21-25	0.0%	30.7%	48.1%	21.2%
26-30	0.0%	9.6%	51.3%	39.1%
31-35	0.0%	0%	46.3%	53.7%
36-40	0.0%	0%	32.7%	67.3%

Retrieved Rank	Matching Class	Matching Interaction Type	All Others
1-5	94.8%	5.2%	0.0%
6-10	18.3%	71.4%	10.3%
11-15	0.0%	57.6%	42.4%
16-20	0.0%	32.3%	67.7%
21-25	0.0%	17.1%	82.9%

Retrieved Rank	Matching Class	Matching Interaction Type	All Others
1-5	89.4%	7.5%	3.1%
6-10	72.6%	19.8%	7.6%
11-15	39.2%	47.4%	13.5%
16-20	14.9%	56.2%	28.9%
21-25	18.8%	45.9%	35.3%
26-30	0%	37.2%	62.8%
31-35	0%	14.6%	85.4%

Retrieved Rank	Matching Class	All Others
1-5	91.3%	8.7%
6-10	40.6%	59.4%
11-15	13.1%	86.9%
16-20	0.0%	100.0%

TABLE 6: Numerical retrieval results for (top) 2C, (second) CRC, and (third) HOI, (fourth) 2PB, (bottom) 2PD. The terms *Attacking Type* and *Interaction Type* are referred to Table 2, 3, 4, 5.

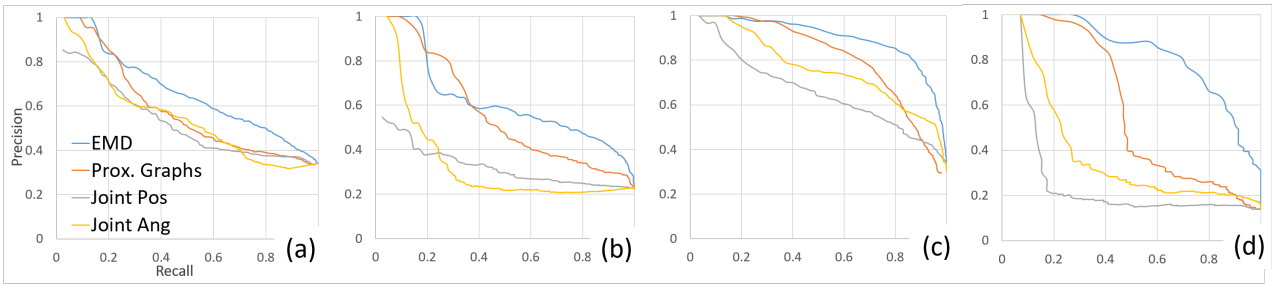

Fig. 20: 2C: Precision and recall of all methods for (a) A1.x, (b) A2.x, (c) A3.x, (d) A4.x.

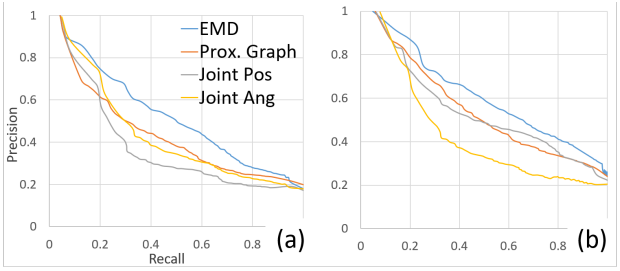

Fig. 21: 2PB: Precision and recall of all methods for (a) C1.x, (b) C2.x.

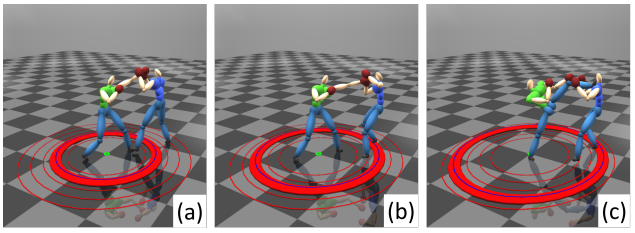

Fig. 22: Interactive retrieval by adjusting the distance between two characters.

the relevancy of the matching decreases. It can be observed that higher-rank results are highly-relevant, with the relevancy dropping in lower-rank results as expected.

For a retrieval application, one possible solution to limit the number of retrieved results is to introduce a manually-tuned threshold on the similarity value. Retrieval results with similarity smaller than such a value are considered irrelevant. Since the aim of the retrieval experiments here is to demonstrate the overall picture of retrieval consistency, we do not include such a threshold value.

5.3 Interactive Retrieval Application

We also implement an interactive retrieval system based on user-provided constraints with the 2C database. These constraints demonstrate the potential of applying our system in interactive animation production, in which the required interaction that fits with the environment and storyboard can be found interactively.

For the best run-time speed, we first precompute the distance between all pairs of interactions in the database using Eq. 8. During run-time, the user provides a query interaction with constraints. Our system then retrieves the most similar interaction that satisfies the constraints in real-time. Such an operation takes 0.02 second on average.

We design a distance constraint as:

$$d_{min} < \left|\mathbf{V}^0_{hips\_A} - \mathbf{V}^0_{hips\_B}\right| < d_{max}, \tag{9}$$

where d_{min} and d_{max} are the lower bound and the upper bound distances given by the user, $\mathbf{V}^0_{hips\_A}$ and $\mathbf{V}^0_{hips\_B}$ are the 3D hips positions of the two interacting characters at frame 0 respectively. This constraint therefore enforces the distance between the characters during the first frame of an interaction. Fig. 22 shows the result of applying the distance constraint, in which the inner and outer radii of the red torus represent d_{min} and d_{max} respectively, the blue circle represents the initial distance between the two characters, and the red circles are markers for visualizing different distance values. Fig. 22a shows the initial interaction. When the preferred distance between the characters increases in Figs. 22b-c, similar interaction that fits the constraints are retrieved.

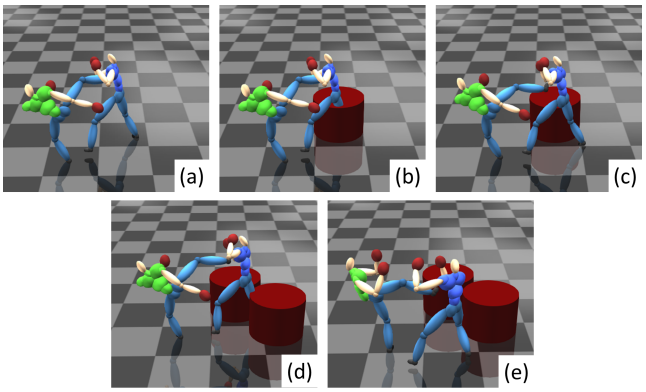

Fig. 23: Interactive retrieval by introducing objects.

We also design an object collision constraint as:

$$\left|\mathbf{V}^t_j - \mathbf{V}_{ob}\right| > d_{ob} \quad \forall t, j, \tag{10}$$

where $\mathbf{V}_{ob}$ is the 3D position of an obstacle, d_{ob} is the distance to avoid colliding with it, $\mathbf{V}^t_j$ represents the position of joint j at frame t. We consider all the

joints of both characters in all frames to ensure that the characters do not collide with the obstacle during the interaction. Fig. 23 shows an example of applying the collision constraint. Fig. 23a is the initial interaction. In Figs. 23b-e, the user introduces objects that lead to collisions. The system then retrieves the most similar interaction that satisfies the object constraint.

6 Conclusion and Discussions

In this paper, we propose a new method for activity comparison from the interaction point of view. This allows us to evaluate movement in a way aligning with the high-level semantic meaning of the interaction. Our method can compare interactions of different topology and discover their intrinsic semantic similarity. Experiments show that our system outperforms existing ones in better evaluating interaction similarity and providing a continuous scale of similarity results. The algorithm can also be used for interaction retrieval to obtain semantically similar interactions, and to suggest suitable interactions based on a set of user-defined constraints.

Our system adapts Earth Mover's Distance to compare interaction meshes of different topologies. Theoretically speaking, such a design can be applied to other features such as joint relative distance as well. However, we prefer the interaction graph structure as it can be used robustly for different kinds of interactions. It can also discover spatial proximity, which is one important aspect in defining interactions.

We use boxing/kickboxing in this project as it has clear logic and rules, which help us to define the hierarchical semantic classes. However, the semantic meaning of some interactions, especially general daily activities, are less well defined. Understanding how real-people comprehend the semantic meaning of interactions, as well as how they weight different factors that affect the semantic similarity, is a challenging but important topic. We are interested in exploring theories in cognitive science for future research.

One future direction is to perform a formal perceptual analysis to obtain real-human perception on the semantic meaning of interactions. Such a study would involve designing a proper experiment to gather perception data. We can then cross-reference that to our results and evaluate how well our algorithm models human understanding. We may also consider using supervised machine learning to learn a metric with the ground-truth perception information.

Another direction is to apply our distance metric for visualization purposes. In areas such as sports science and rehabilitation, there are a lot of training exercises involving human-human or human-object interactions. The proposed interaction comparison method can better visualize how two interactions, potentially one from a novice and another from an expert, are similar or different.

We propose a simple vertex sampling process in Section 4.1 such that the interaction comparison is less affected by the joint hierarchy of the character. It is an interesting future direction to explore more advanced methods in vertex sampling, such as using samples to replace joints, considering the topology of the joint hierarchy, or even considering the surface information of the character instead of the skeleton. A good sampling scheme would facilitate the comparison of interaction with characters of different joint hierarchies (e.g. having long legs) or even different structure (e.g. having multiple arms).

We use a chair as the object in the HOI database as it has a complex structure and different ways to interact with, thereby covering a wide range of human-object interactions. One future direction is to evaluate the method with more diverse objects, or even to incorporate multiple objects into the scene. More research can be done to model the sub-part of the object the character interacts, especially for larger objects.

While we demonstrate the method using 2-person and character-object interactions, it can be applied to single human activities by representing a posture with an interaction mesh connecting different joints, thereby modelling the spatial relationship among all body parts. However, unlike interactions, many single character motions involve minor spatial differences, such as waving versus pointing. The lack of an object or another character to interact with results in less information for the interaction mesh to represent. Therefore, our method may not have an absolute advantage over existing ones. One of our future direction is to explore combining interaction mesh with traditional features such as joint angles to strengthen single character motion comparisons.

Our proposed method will be less effective for activities in which there is no close interaction or the characters are far away from each other, which exist in some general daily activities. This is because if the body parts of the characters are not in proximity, the interaction mesh created tend to be having a similar uniform structure (e.g. similar edge lengths, similar topology). This increases inter-class similarity and degrades the retrieval performance. One possible solution to be explored is to introduce non-linear functions for normalizing the length and angle of the interaction mesh.

It is challenging to identify semantically similar interactions with a large variety of style or movement strategy. For example, in a punching-avoiding interaction, the avoider can duck or back-step. The former involves squatting and then standing up, while the latter involves only one step backwards. This explains the challenge in A1.x and A2.x. Our system performs better than existing ones by focusing on interaction features. Still, obtaining human-level accuracy requires more research.

ACKNOWLEDGEMENT

This project was supported by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1) and the Royal Society (Ref: IES\R2\181024).

REFERENCES

[1] M. Müller, A. Baak, and H.-P. Seidel, "Efficient and robust annotation of motion capture data," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '09. New York, NY, USA: ACM, 2009, pp. 17–26.

[2] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012.

[3] E. S. L. Ho and T. Komura, "Character motion synthesis by topology coordinates," *Computer Graphics Forum*, vol. 28, no. 2, pp. 299–308, 2009.

[4] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of the Sixth International Conference on Computer Vision*, Washington, DC, USA, 1998, pp. 59–66.

[5] E. S. L. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 33:1–33:8, Jul. 2010.

[6] H. Si and K. Gärtner, "Meshing piecewise linear complexes by constrained delaunay tetrahedralizations," in *In Proceedings of the 14th International Meshing Roundtable*. Springer, 2005, pp. 147–163.

[7] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, "Convolutional wasserstein distances: Efficient optimal transportation on geometric domains," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 66:1–66:11, Jul. 2015.

[8] J. C. Chan, J. K. Tang, and H. Leung, "Synthesizing two-character interactions by merging captured interaction samples with their spacetime relationships," *Computer Graphics Forum*, vol. 32, no. 7, pp. 41–50, 2013.

[9] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 559–568, 2004.

[10] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 491–500, 2002.

[11] H. P. H. Shum, T. Komura, and S. Takagi, "Fast accelerometer-based motion recognition with a dual buffer framework," *The International Journal of Virtual Reality*, vol. 10, no. 3, pp. 17–24, Sep 2011.

[12] E. S. L. Ho and H. P. H. Shum, "Motion adaptation for humanoid robots in constrained environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3813–3818.

[13] L. Ren, A. Patrick, A. A. Efros, J. K. Hodgins, and J. M. Rehg, "A data-driven approach to quantifying natural human motion," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1090–1097, Jul. 2005.

[14] M. Kapadia, I.-k. Chiang, T. Thomas, N. I. Badler, and J. T. Kider, Jr., "Efficient motion retrieval in large motion databases," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '13. New York, NY, USA: ACM, 2013, pp. 19–28.

[15] J. K. T. Tang, H. Leung, T. Komura, and H. P. H. Shum, "Emulating human perception of motion similarity," *Comput. Animat. Virtual Worlds*, vol. 19, no. 3–4, pp. 211–221, 2008.

[16] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 588–595.

[17] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao, "Learning a 3d human pose distance metric from geometric pose descriptor," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1676–1689, Nov 2011.

[18] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 42:1–42:13, Jul. 2017.

[19] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics*, vol. 35, no. 4, 2016.

[20] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics (Proc. SIGGRAPH 2018 - to appear)*, vol. 37, no. 4, 2018.

[21] K. H. Lee, M. G. Choi, and J. Lee, "Motion patches: Building blocks for virtual environments annotated with motion data," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 898–906, Jul. 2006.

[22] J. Won, K. Lee, C. O'Sullivan, J. K. Hodgins, and J. Lee, "Generating and ranking diverse multi-character interactions," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 219:1–219:12, Nov. 2014.

[23] K. Hyun, K. Lee, and J. Lee, "Motion grammars for character animation," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 103–113, May 2016.

[24] M. Kim, K. Hyun, J. Kim, and J. Lee, "Synchronized multi-character motion editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 79:1–79:9, Jul. 2009.

[25] E. S. L. Ho, H. P. H. Shum, Y.-m. Cheung, and P. C. Yuen, "Topology aware data-driven inverse kinematics," *Comp. Graph. Forum*, vol. 32, no. 7, pp. 61–70, Oct 2013.

[26] E. S. L. Ho and T. Komura, "Indexing and retrieving motions of characters in close contact," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 481–492, 2009.

[27] R. A. Al-Asqhar, T. Komura, and M. G. Choi, "Relationship descriptors for interactive motion adaptation," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '13. New York, NY, USA: ACM, 2013, pp. 45–53.

[28] V. Ivan, D. Zarubin, M. Toussaint, T. Komura, and S. Vijayakumar, "Topology-based representations for motion planning and generalization in dynamic environments with interactions," *Int. J. Rob. Res.*, vol. 32, no. 9-10, pp. 1151–1163, Aug. 2013.

[29] J. K. Tang, J. C. Chan, H. Leung, and T. Komura, "Interaction retrieval by spacetime proximity graphs," *Comp. Graph. Forum*, vol. 31, no. 2pt2, pp. 745–754, May 2012.

[30] E. S. L. Ho, J. C. P. Chan, Y.-m. Cheung, and P. C. Yuen, "Modeling spatial relations of human body parts for indexing and retrieving close character interactions," in *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '15. New York, NY, USA: ACM, 2015, pp. 187–190.

[31] M. Li and H. Leung, "Multiview skeletal interaction recognition using active joint interaction graph," *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2293–2302, Nov 2016.

[32] H. P. H. Shum, T. Komura, and S. Yamazaki, "Simulating multiple character interactions with collaborative and adversarial goals," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 741–752, May 2012.

[33] ——, "Simulating competitive interactions using singly captured motions," in *Proceedings of the 2007 ACM symposium on Virtual Reality Software and Technology*, ser. VRST '07. New York, NY, USA: ACM, Nov 2007, pp. 65–72.

[34] H. P. H. Shum, T. Komura, M. Shiraishi, and S. Yamazaki, "Interaction patches for multi-character animation," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–8, 2008.

[35] I. Ouergui, N. Hssin, M. Haddad, E. Franchini, D. G. Behm, D. P. Wong, N. Gmada, and E. Bouhlel, *The Journal of Strength & Conditioning Research*, vol. 28, no. 12, pp. 3537–3543, 2014.

[36] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," in *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*. New York, NY, USA: ACM, 2004, pp. 522–531.

[37] N. v. d. Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp, "Utrecht multi-person motion (umpm) benchmark: a multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction," in *Proceedings of the Workshop on Human Interaction in Computer Vision (HICV), in conjunction with ICCV 2011*, 2011.

[38] E. S. L. Ho, J. C. P. Chan, T. Komura, and H. Leung, "Interactive partner control in close interactions for real-time

applications," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 3, pp. 21:1–21:19, Jul. 2013.
[39] Y. Wu, "Mining actionlet ensemble for action recognition with depth cameras," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1290–1297.
[40] X. Zhao, M. Choi, and T. Komura, "Character-object interaction retrieval using the interaction bisector surface," *Comput. Graph. Forum*, vol. 36, no. 2, pp. 119–129, May 2017.
[41] J. Assa, Y. Caspi, and D. Cohen-Or, "Action synopsis: Pose selection and illustration," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 667–676, Jul. 2005.
[42] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

Yijun Shen received his B.Eng. degree from The University of Sheffield, U.K., in 2012, and his M.Sc. degree from The University of Edinburgh, U.K., in 2013. He has received a full PhD studentship from Northumbria University, U.K., and is concluding his Ph.D. degree. He has also received a research internship from INRIA, France, in 2016. His research interests include character animation and human motion analysis.

Longzhi Yang is the Director of Learning and Teaching and an Associate Professor at the Department of Computer and Information Sciences with Northumbria University, U.K. His research interests include computational intelligence, machine learning, big data, computer vision, intelligent control systems, and the applications of such techniques under real-world uncertain environment. He is the Founding Chair of the IEEE Special Interest Group on Big Data for Cyber Security and Privacy. He received the Best Student Paper Award at the 2010 IEEE International Conference on Fuzzy Systems.

Edmond S. L. Ho received the BSc degree in Computer Science from the Hong Kong Baptist University, the MPhil degree from the City University of Hong Kong, and the Ph.D. degree from the University of Edinburgh. He is currently a Senior Lecturer with the Department of Computer and Information Sciences at Northumbria University, UK. Prior to that, he was a Research Assistant Professor with Hong Kong Baptist University. His research interests include character animation, human motions analysis, robotics, and human activity understanding.

Hubert P. H. Shum is an Associate Professor (Reader) at Northumbria University, U.K., as well as the Director of Research and Innovation of the Computer and Information Sciences Department. Before this, he worked as a Senior Lecturer at Northumbria University, U.K., a Lecturer in the University of Worcester, U.K., a post-doctoral researcher in RIKEN, Japan, as well as a research assistant in the City University of Hong Kong. He received his Ph.D. degree from the School of Informatics in the University of Edinburgh, U.K. His research interests include computer graphics, computer vision, motion analysis and machine learning.