Interaction-based Human Activity Comparison, Analysis and Synthesis

BY EDMOND S. L. HO

NORTHUMBRIA UNIVERSITY, NEWCASTLE UPON TYNE, UK

Agenda

- Introduction – About me
- Basics in Character Animation
- Problems in Analysing and Synthesizing Close Interactions
- Our recent work
- Conclusion
- Q&A Session

Introduction – About Me

About me

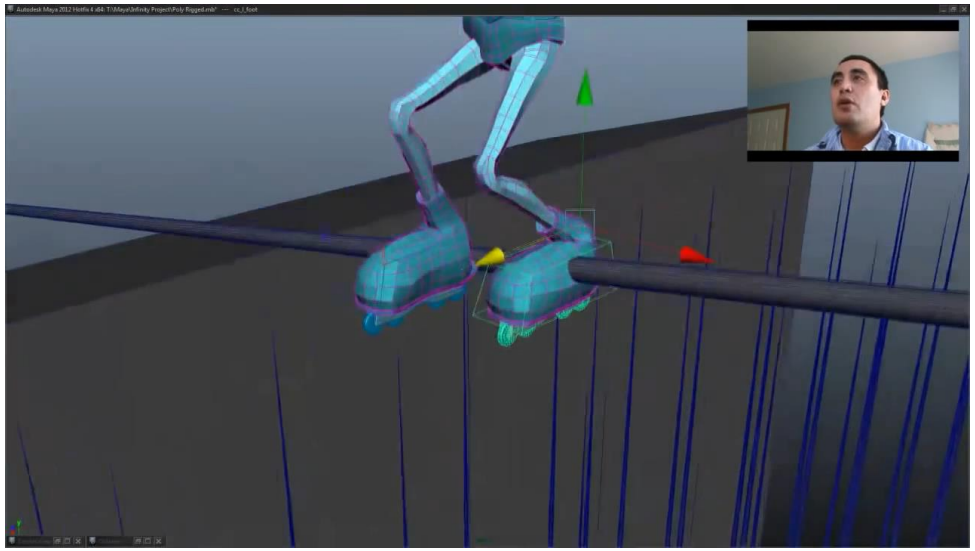
- Edmond S. L. HO
  - Senior Lecturer, Dept. of Computer and Information Sciences, Northumbria Uni
    - Since Aug 2016
  - Research Assistant Professor, Dept. of Computer Science, Hong Kong Baptist Uni
    - Oct 2011 – Aug 2016
  - PhD in Informatics, University of Edinburgh, 2011

- Research interests
  - Computer Graphics, Computer Vision, Robotics, Motion Analysis, and Machine Learning

Basics in Character Animation – a **brief** overview

Editing Character Manually

- Manually editing (i.e. posing character) is still widely used in animation production nowadays
  - Directly editing the orientation of a body segment by specifying the 'joint rotations' (eg x, y, and z rotations)
  - Specifying the location of a body segment and the new pose will be computed using Inverse Kinematics (IK)
  - Character animation = sequence of poses
    - Extremely time consuming and labour-intensive
    - A recent study Kyto et al. (2015) found that 61 % of the time in animation production was spent on character posing by professional animators.



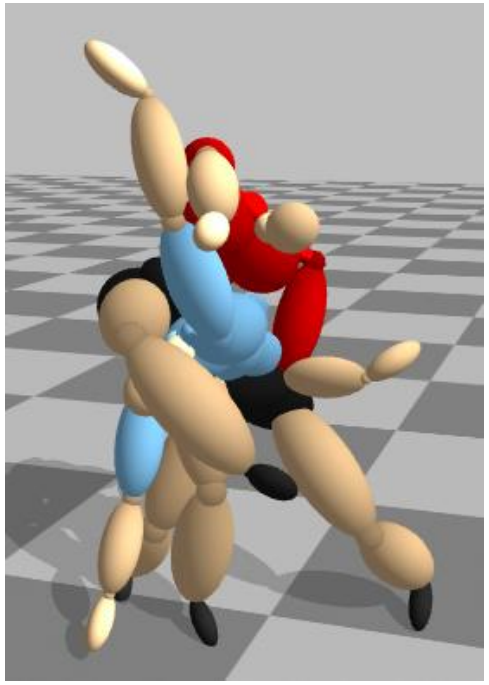
Character Animation Tutorial Video (randomly) downloaded from YouTube

Basics in Computer Animation

- It is important to (semi-)automate these tasks
- An active research area in the last 2-3 decades
- A wide range of methods are available
- A (very) brief review is included in the following slides

Inverse kinematics (IK)

- Not an easy task to animate characters by specifying the joint angles one-by-one
  - Trial and error, time-consuming to create complicated postures
- More intuitive to edit an existing pose by specifying the target position(s) and orientation(s) of a(some) joint(s) as constraints
  - Then, the full body configuration (e.g. all joint angles) will be calculated as final pose representation



Inverse kinematics (IK)

- This can be solved numerically
  - Decide how to change each joint angles in order to satisfy the constraints
    - i.e. to move the end-effector from the current position to the goal
  - Need to find out the relationship between ***how the changes in the joint angles (e.g. X) affect the configuration of the end-effector (e.g. Y)***

$$dY = \frac{\partial F}{\partial X} dX$$

    - Can be done by computing the matrix of partial derivatives called Jacobian *J*

$$J_x = \frac{\partial F}{\partial X}$$

Inverse kinematics (IK)

- E.g. If we want to control a robot arm with 4 joint parameters (e.g. joint angles) to control the 3D position of the end-effector, we can represent this IK problem by

$$\dot{Y} = J_x \dot{X}$$

where

$\dot{Y} = \left[\dot{p}_x, \dot{p}_y, \dot{p}_z \right]^T$ contains the changes of the 3D position of the end-effector (Note: we can specify joint orientation as constraints as well)

$\dot{X} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4 \right]^T$ contains the changes of the joint angles

- Then, the Jacobian will be:

$$J_x = \begin{bmatrix} \frac{\partial p_x}{\partial \theta_1} & \frac{\partial p_x}{\partial \theta_2} & \frac{\partial p_x}{\partial \theta_3} & \frac{\partial p_x}{\partial \theta_4} \\ \frac{\partial p_y}{\partial \theta_1} & \frac{\partial p_y}{\partial \theta_2} & \frac{\partial p_y}{\partial \theta_3} & \frac{\partial p_y}{\partial \theta_4} \\ \frac{\partial p_z}{\partial \theta_1} & \frac{\partial p_z}{\partial \theta_2} & \frac{\partial p_z}{\partial \theta_3} & \frac{\partial p_z}{\partial \theta_4} \end{bmatrix}$$

Inverse kinematics (IK)

- Numeric solutions to IK
  - Using the inverse Jacobian

$$V = J\dot{\theta}$$

$$J^{-1}V = \dot{\theta}$$

  - However, sometimes the Jacobian is not a square-matrix (cannot find the inverse!)
    - A pseudo inverse (J^{+}) can be used

$$V = J\dot{\theta}$$

$$J^{T}V = J^{T}J\dot{\theta}$$

$$\left(J^{T}J\right)^{-1}J^{T}V = \left(J^{T}J\right)^{-1}J^{T}J\dot{\theta}$$

$$\left(J^{T}J\right)^{-1}J^{T}V = I\dot{\theta}$$

$$\left(J^{T}J\right)^{-1}J^{T}V = \dot{\theta}$$

$$J^{+}V = \dot{\theta} \qquad \text{where} \quad J^{+} = \left(J^{T}J\right)^{-1}J^{T}$$

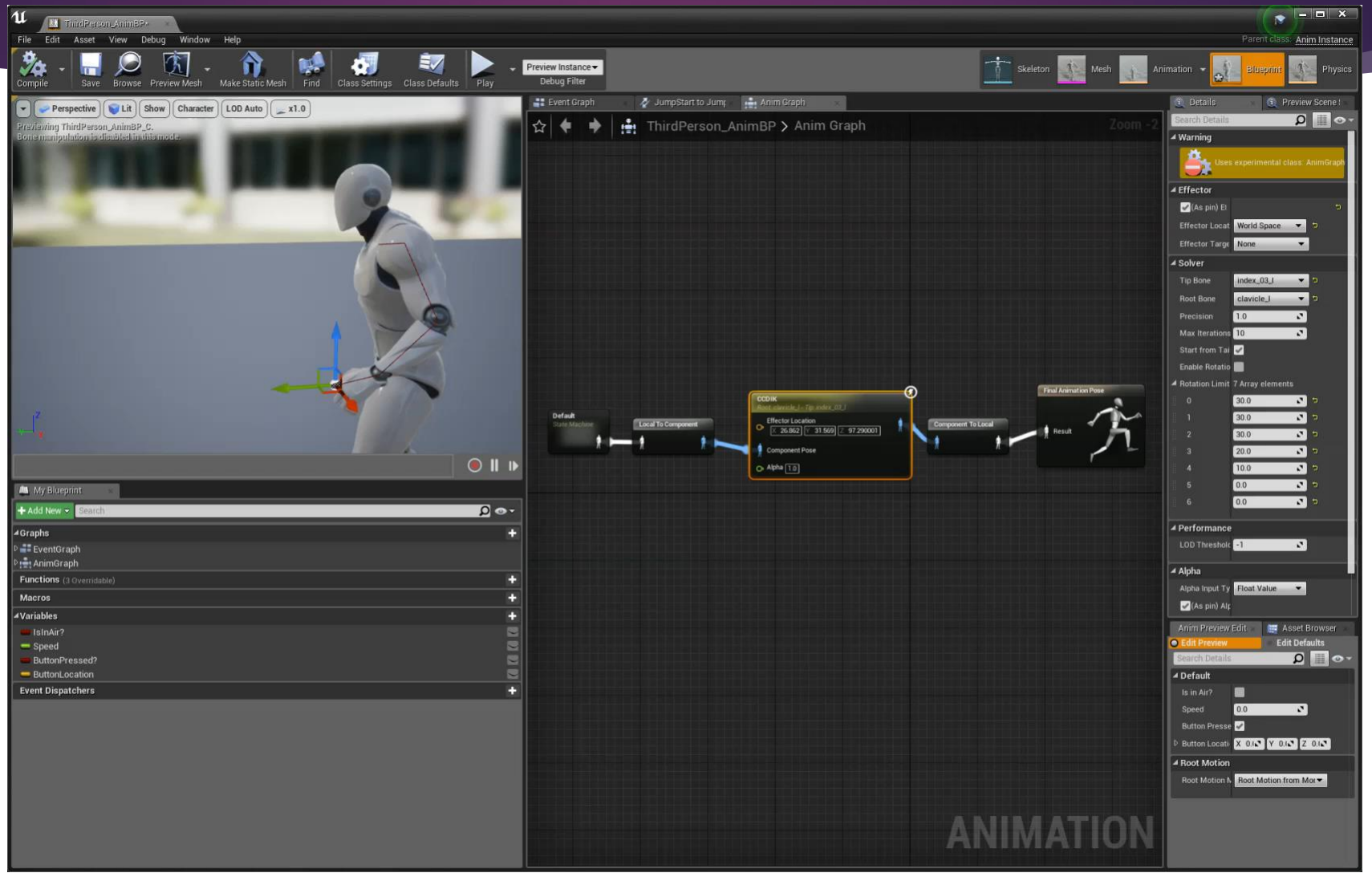
Inverse kinematics (IK)

- Solving IK problem procedurally
- We can use another approach called Cyclic Coordinate Decent (CCD)
  - The idea is simple, rotate the joint one by one to satisfy the constraints
  - Start from the outermost joint (i.e. closest to the end-effector)
  - repeat the whole process until the constraint is being satisfied

  - For each joint, given the current positions of goal G, end-effector E, and the joint Ji to be rotated, we can find the required rotation θ to move E to G by dot product

$$(E - J_i) \bullet (T - J_i) = \|E - J_i\| \|T - J_i\| \cos\theta$$

CCDIK in Unreal Game Engine

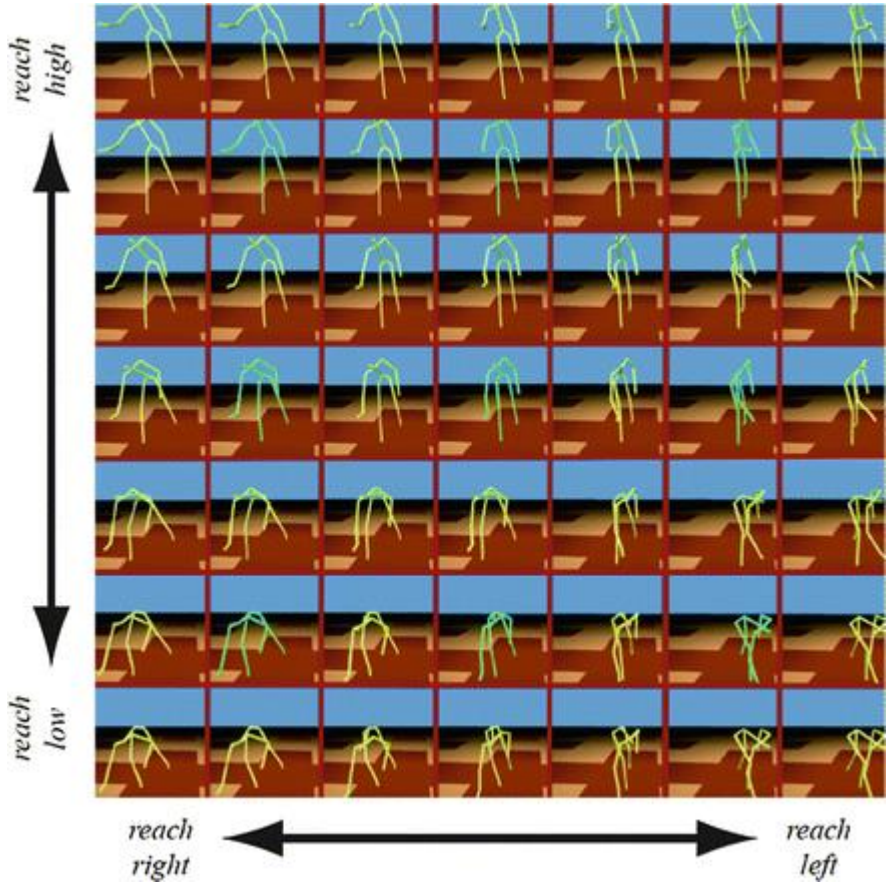


Data-Driven Pose Synthesis

- The idea of data-driven motion synthesis is to make use of captured motion data to create the required postures
  - such that **natural** and **humanlike** movement can be created by specifying a relatively small number of constraints
- An early work by Rose et al. (1998) edits poses by interpolating collected poses to satisfy the constraints
  - based on an old technique called motion blending in computer animation, i.e. combining different motions into a single one
  - The concept of verbs and adverbs is proposed to generate new poses from examples

verbs and adverbs - Rose et al. (1998)

- *verbs* refer to parameterized motions constructed from sets of similar motions, and *adverbs* are parameters that control the verbs
- For each verb, the sample motions (green) are time aligned by manually specifying the key-time for every motion
- Then, the motions clips are placed on a parameter space based on the characteristics of the motion clips
- Motion blending (yellow) is done by computing the weights of the sample motions in the corresponding verb using radial basis functions (RBF)
- By specifying the adverbs, new motion will be created. In addition, users can create a *verbgraph* so that transition motions between verbs can be generated



Synthesizing a natural-looking pose

- Can be viewed as finding a solution (i.e., joint parameters) from a natural movement space created using captured motions
- Grochow et al. (SIGGRAPH2004) propose to use scaled Gaussian process latent variable model (SGPLVM) (Lawrence 2004) to create such a natural pose space
- By specifying constraints such as the positions of the hands and feet, natural-looking full-body pose can be synthesized
- However, due to the complexity of the learning process, the model cannot be trained with a large number of poses

Style-based IK

Synthesizing a natural-looking pose

- As opposed to offline training approaches, online modeling has shown to be effective for real-time application with large motion dataset
- For example, Chai and Hodgins (SIGGRAPH2005) use a lazy learning approach
  - to learn low-dimensional local linear models (principal component analysis (PCA)) to approximate the high-dimensional manifold which contains the natural and valid poses during runtime
- Given the current pose of the character and the target positions of the selected joint(s) as constraints, a set of postures that are similar to the current one is used to learn the local linear model

Synthesizing a natural-looking pose

- By interpolating the poses in the low-dimension space while minimizing the energy terms to ensure that
  - the synthesized pose is smooth (i.e., joint velocities), and
  - satisfy the constraints given by the user and the probability distribution of the captures motion in the training data
- Natural-looking full-body motion can be synthesized
- The general problem of these methods is that it is difficult to ensure the set of extracted postures to be logically similar as kinematics metric is used

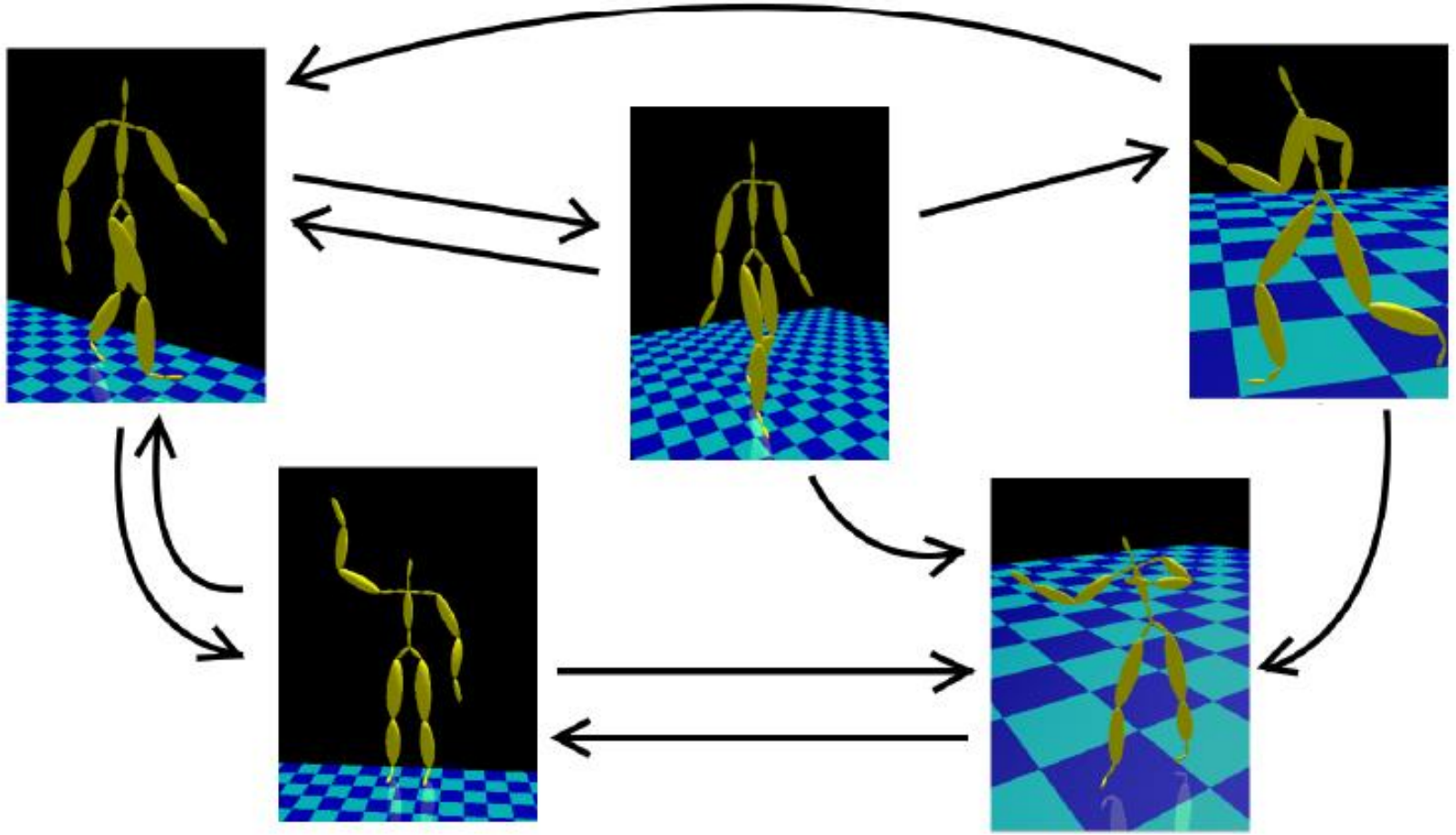
Motion Editing approaches

- We want to…
  - Synthesize new motions by editing existing motions
- We can …
  - Pose editing by Inverse Kinematics (IK)
- Various motion editing techniques have been proposed in the area in Computer Animation
  - Motion Graph
  - Interpolation and Parameterization

Motion Graph

[Arikan and Forsyth, 2002, Lee et al., 2002, Kovar et al., 2002]

- Syn
  ca
- No
- Co
  be
- Cre



Building Motion Graph

- Segment long captured motion sequence into short clips
  - E.g. dividing a walking motion into walking cycles
- ‘Concatenate’ short motion segments to create a long sequence
  - Here, we can re-arrange the short clips to create new motion!
- However, we cannot directly concatenate the short clips
  - Let’s skip the details for now…

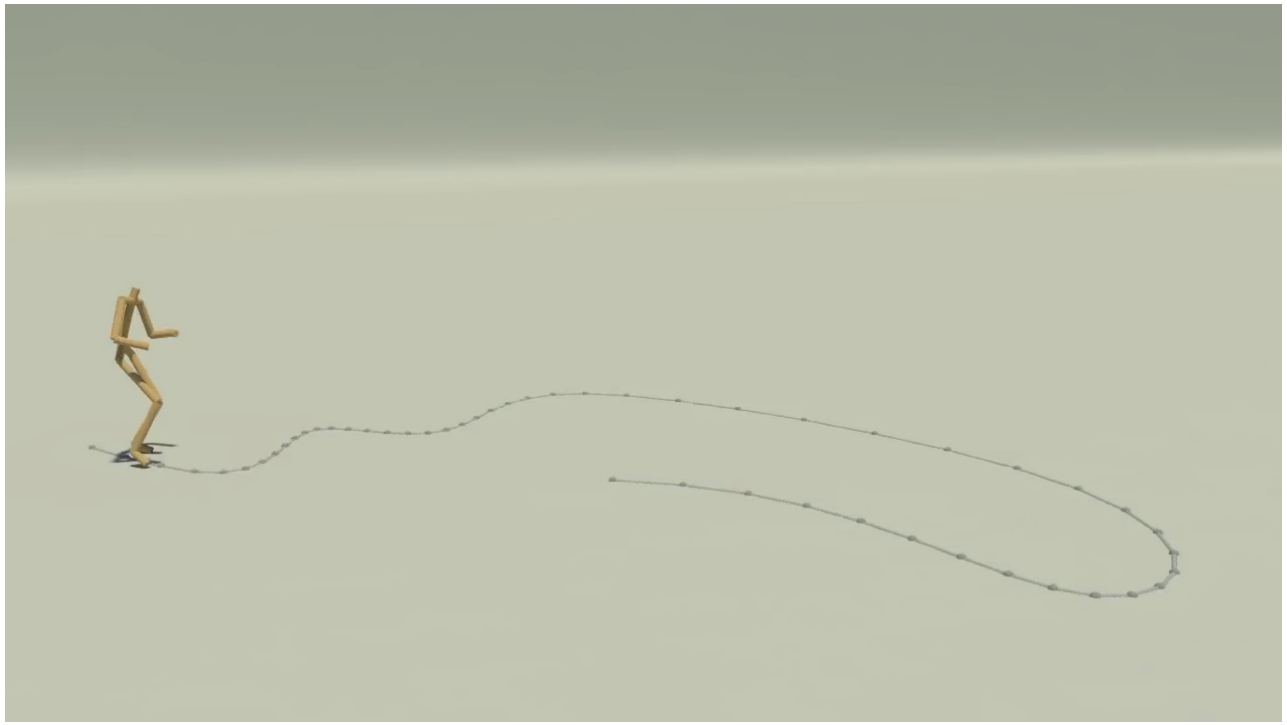
Animations created using Motion Graph

- Follow a walking path (left)
- Interactively control the walking direction (right)

Deep Learning based approaches

- A huge jump from the previous slides!
- A deep learning framework for character motion synthesis and editing is proposed in SIGGRAPH 2016 (by Holden et al.)
- The core idea is to learn a natural motion manifold (represented by the hidden units of a convolutional autoencoder) can be used for a wide range of tasks in character animation
  - By specifying additional constraints, we can do
    - Noise removal, motion correction, style transfer, path following, pose editing…., etc.

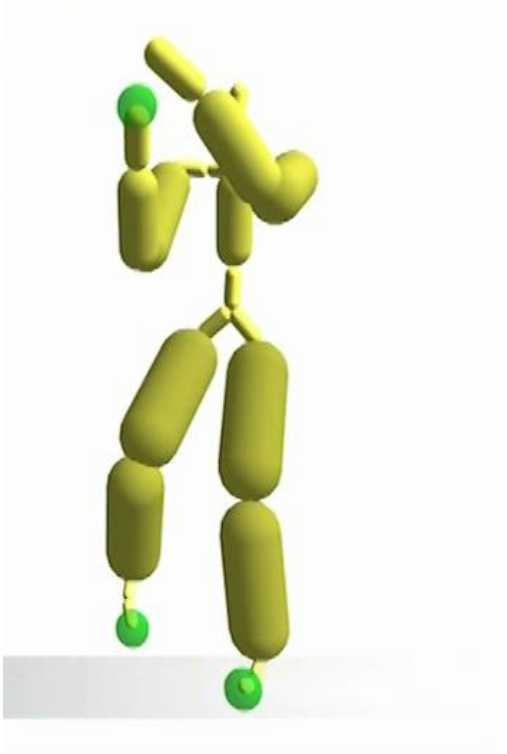
Holden et al. SIGGRAPH2016



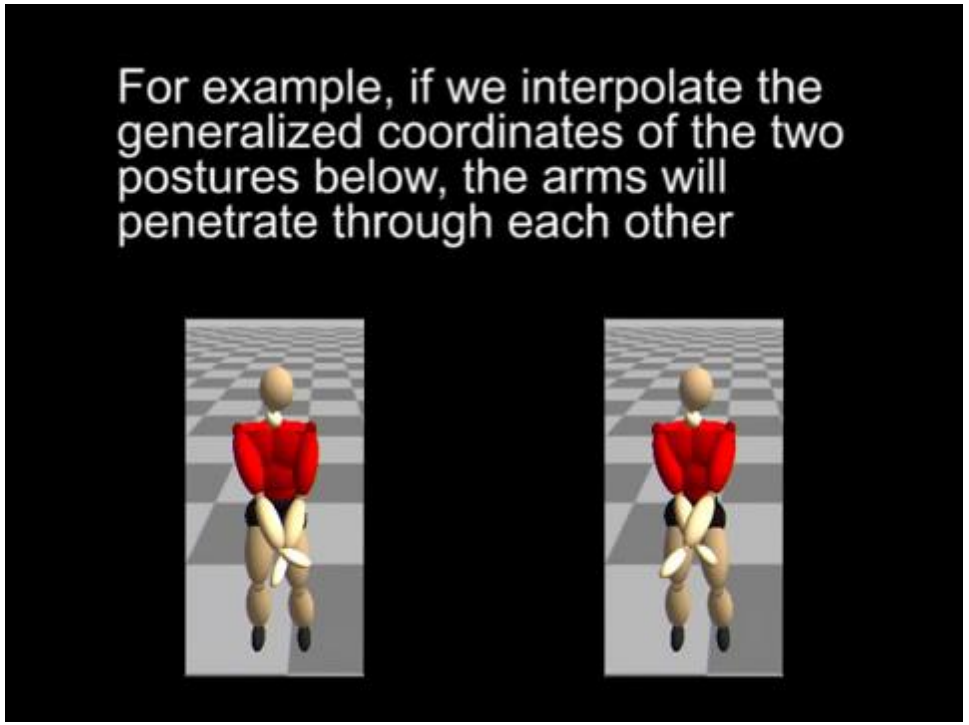
Problems in Analysing and Synthesizing Close Interactions

Close interactions

- Handling close interactions (human-human and human-object interactions) is a challenging topic
  - Conventional representations (eg 3D positions, joint angles) do not contain the 'interaction' information
  - Directly applying existing techniques to analyse and synthesize interaction will not work well
    - Motion analysis: mixing up different motion/interaction classes
    - Motion synthesis: interpenetration of body parts (yellow character)
      - Still relying on manual editing in the animation industry
      - Extremely time consuming and labour-intensive



The underlying problems



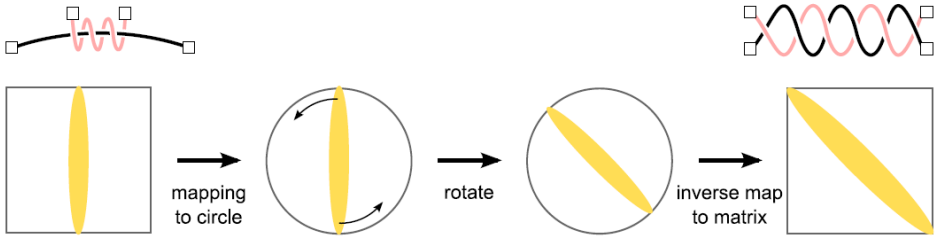
Topology Coordinates (Ho and Komura, EG2009)

- Finding the topological relationship between 2 strings
  - Here, the chained body segments, such as limbs, are presented as strings
- There are three attributes in topology coordinates
- The first attribute is the **writhe**, which counts how much the two curves are twisting around each other#
- Writhe can be calculated by using Gauss Linking Integral (GLI) [Poh68] by integrating along the two curves g1 and g2 as:

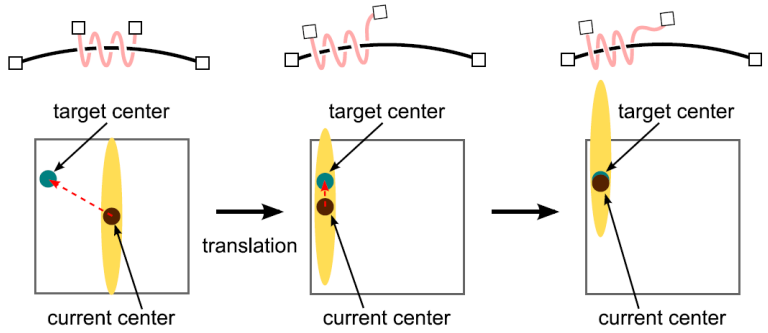
$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{\|\gamma_1 - \gamma_2\|^3}$$

Topology Coordinates (Ho and Komura, EG2009)

- The computed GLI values can then be represented in a matrix form (called writhe matrix in the paper)
- The distribution of the values in the matrix describes how the 2 strings are twisted around each other
- We further proposed the 2nd and 3rd attributes in topology coordinates to control the strings, namely **density** and **center**



Editing the density value

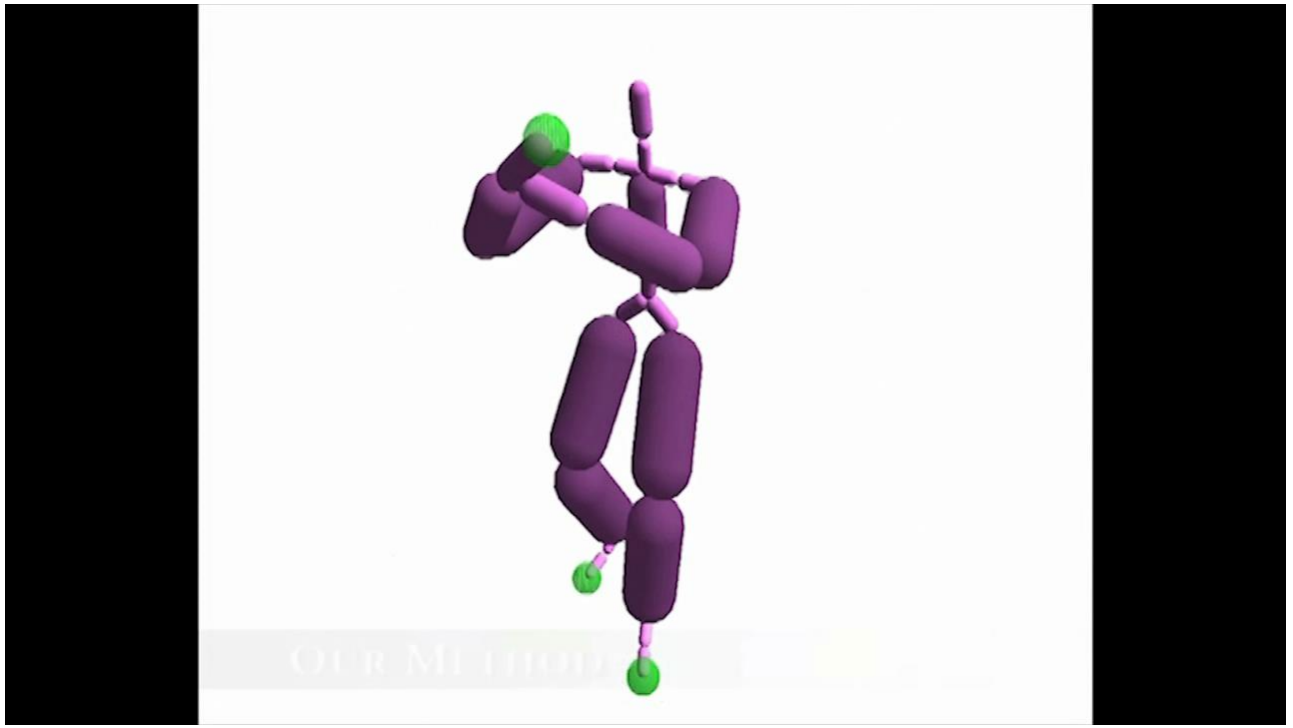


Editing the center value

Topology Coordinates (Ho and Komura, EG2009)

Topology-aware Data-driven IK (Ho et al. CGF2013)

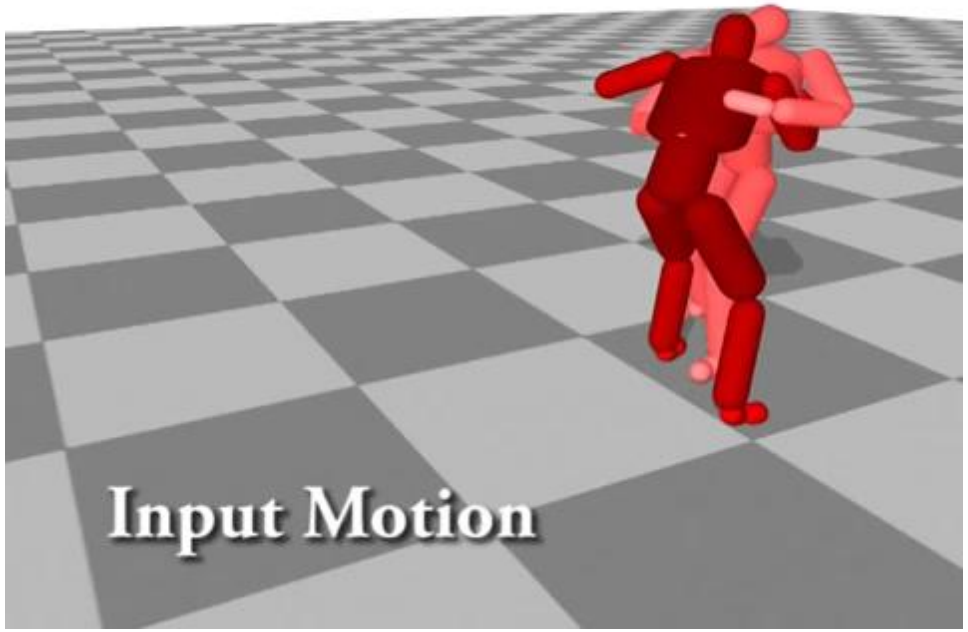
- To further improve the motion quality (i.e. naturalness) by using a data-driven approach
- Using the lazy-learning approach to learn a natural motion space
- Select topologically similar poses when creating the space
  - To avoid a significant change in topology and results in interpenetration of the body parts



Interaction Mesh (Ho et al. SIGGRAPH2010)

- The aforementioned tangle-based approaches are less effective when there is not much 'tangling'
- We proposed a more general approach by encoding the spatial relationships using a mesh structure
- A simple concept
  - Model the 'in-between' space between the characters using a 3D mesh
  - When editing the character(s), minimize the distortion of the mesh to preserve the spatial relationships
- Can be used for retargeting and editing close interactions

Interaction Mesh (Ho et al. SIGGRAPH2010)



Yin et al. TVCG2019

- Capturing closely interacting people using 3D sensors is a challenging task
  - Due to the occlusion problems for vision-based systems
  - Inertial and magnetic systems usually suffer from drifting and distortion problems
- In this work, we proposed a sampling framework **to generate a lot of poses** from a single seed pose, with
  - a newly proposed Interaction Coordinates (IC) to represent close interactions
  - a physical prior to bias the system to sample physically valid poses
  - using a small amount of annotated 2D poses

Interaction comparison

"Interaction-based Human Activity Comparison"

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

IEEE Transactions on Visualization and Computer Graphics, accepted, Jan 2019.

Introduction

- Comparing human activities is a core problem in areas such as sports sciences, rehabilitation and monitoring
  - Usually require the user to perform a set of pre-defined activities
    - Then evaluate the correctness/quality by comparing the performed activities with given exemplars
  - Traditional motion analysis methods typically require
    - the type of the activities to be known in advance
      - To apply the right criteria for evaluations, and
      - Can only evaluate the similarity of activities (same class)
    - 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

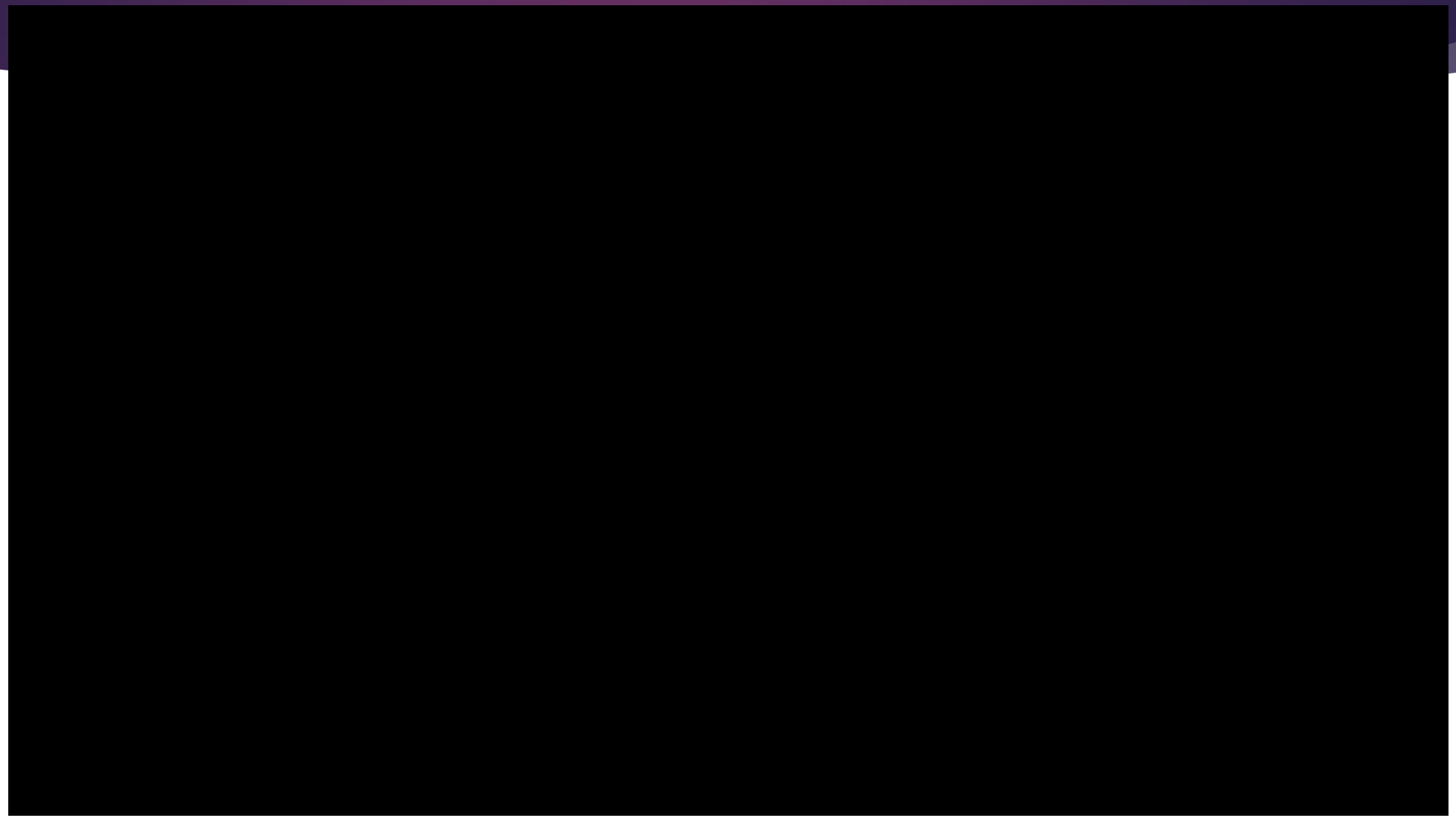
Introduction

- Existing research 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

Introduction

- 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

Introduction

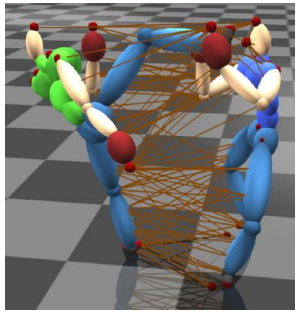


Our observations

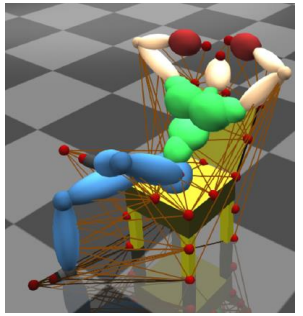
- We observed that high-level activities are usually defined based on character-character or character-environment *interactions*, such as **punching an opponent**, **sitting on a chair** and **jumping over a fence**
- The contextual meaning of the activity depends heavily on the interaction instead of individual movement. 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

Our method – an overview

- We propose a new metric for evaluating the degree of similarity between interactions by adapting Earth Mover's Distance onto a customized interaction mesh structure that represents spatial-temporal interactions



- We establish correspondences between topologically different structures from different interactions, such that we can **evaluate the similarity of interactions between different classes**

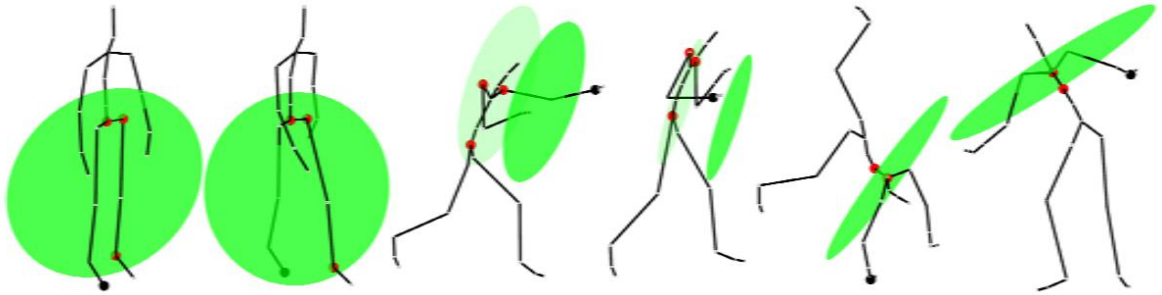


Related Work

- There is a large body of research about analysing 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** (Kovar and M. Gleicher, SIGGRAPH2004) and **joint angles**-based distance (Lee et al., SIGGRAPH2002) to evaluate different types of motion
  - E.g. finding suitable motions to create smooth transitions in Motion Graphs

Related Work - Logical rules based features

- **Relational motion features** - Logical rules based on combined kinematic features can be used as the motion features in motion retrieval (Müller et al. SIGGRAPH2005, Müller et al. SCA2006, Müller et al. SCA2009)
  - Each logical rule is associated with 4 joints
    - Using 3 of them to create a plane
    - The feature is defined by a Boolean value to indicate whether the 4th joint is in front of the plane
    - 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



Related Work - Relative kinematic features

- such as Relative Joint Distance
  - the feature dimension increases exponentially when considering multiple characters
    - Eg computing all pairs of joint distance among all characters
  - feature selection (Yun et al., CVPRW2012), (Tang et al., CAVW2008) can be used to maintain a reasonable feature dimension
    - the optimal set of the feature depends on the type of interactions
    - difficult to find a globally optimal set of low dimensional feature to represent all interaction types

Related Work – Spatial relation-based features

- Knot Theory –inspired representation
  - Based on the Tangle concept in Knot Theory
  - Treating the tree structure of the human skeleton as a set of strings, and the interactions between body parts as the 'tangles' on the strings
  - Using Gauss Linking Integral to represent how these strings wrap around each other
  - Such a representation can be used as motion indexing and retrieval (Ho and Komura, TVCG2009)
  - However, the representation cannot effectively represent non-contacting interactions such as one character avoiding an attack from another

Methodology

Interaction Databases

- Multi-character (Close) interaction motion data is very limited
  - Mostly of the public datasets are having social interactions (such as shaking hands, waving, etc) only
- We captured/collected 5 databases in this work
  - Character-Character (2C)
  - Character-Retargeted Character (CRC)
  - Human-Object Interaction (HOI)
  - 2 People Boxing (2PB)
  - 2 People Daily Interaction (2PD)

1. Character-Character (2C) Dataset

- Kick-boxing motions
  - involves a large variety of movements (punch, kicking, and combos)
- We adapt the interaction synthesis framework proposed in (Shum et al., TVCG2012) to synthesize high-quality interactions
  - To 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 (Shum et al., VRST2007)
  - Second, we define a set of semantic interaction classes, each defines the interaction pattern (Shum et al., SIGGRAPHAsia2008) 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 (Shum et al., TVCG2012), and extract the interactions that fit into our pre-defined list of interaction classes

Character-Character (2C)

- 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 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 describ level details of the attack (left or right)

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

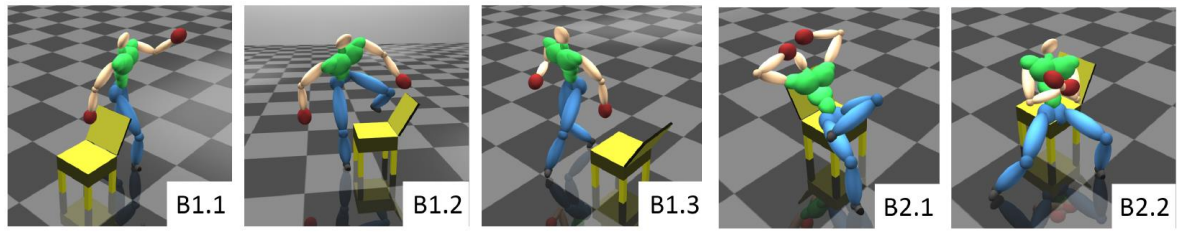
Character-Retargeted Character (CRC)

- 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 in the 2C daatbase
- We then resize one character into 80% to 130% of the original size in every 10% step
  - It suggests that such a range (Ho et al., SIGGRAPH2010) is effective for interaction retargeting without changing contact information
- We finally retarget the movement using Autodesk MotionBuilder
- An example frame of retargeted interaction is shown in 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
- 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

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



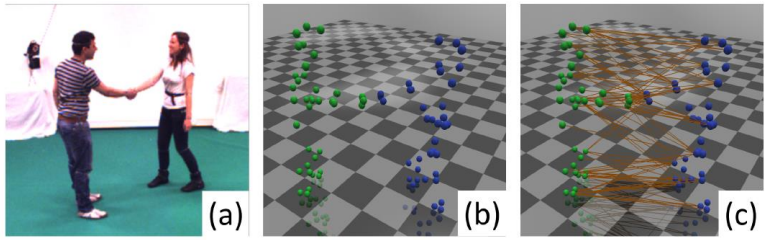
2 People Boxing (2PB)

- To evaluate the performance of our system for real-people interactions, we created a database of boxing motion performed by 2 people (2PB)
- This is a challenging database with complex interactions and body movement
- We collect around 6 minutes of boxing from 4 pairs of professional boxers
- 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).

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

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 (Aa et al., 2011).
- 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
- 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



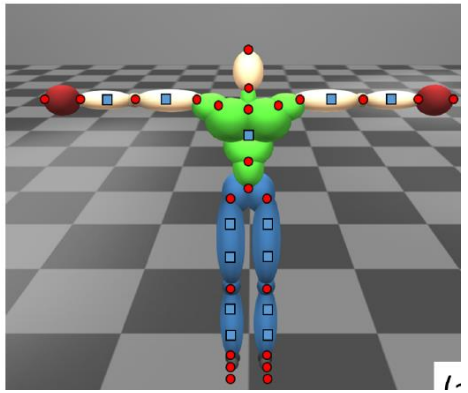
Interaction Type	Class
A and B Walk Around in a Circular Manner	D1
A and B Dance Together	D2
A and B Shake Hands	D3
A and B Chat with Each Other	D4

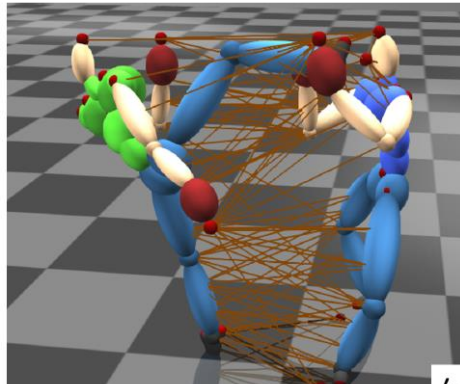
UNIFIED INTERACTION COMPARISON

- Customized Interaction Mesh Structure
  - 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, 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

Creating the mesh representation

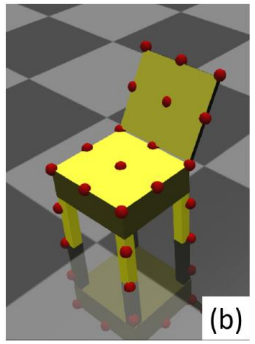
- We customize the process to generate the interaction mesh (Ho et al., SIGGRAPH2010) such that the resultant mesh is more suitable for interaction comparison
  - to have a uniform distribution of vertices to ensure that the comparison is not biased to body parts with more joints
- We include a set of vertices by uniformly sampling the skeletal structure
- In our implementation, a character consists of 25 joints (red)
  - We uniformly sample body segments using a sampling length of 15cm
  - This process creates another 13 vertices (blue)
- To create the interaction mesh, we apply Delaunay Tetrahedralization on the point cloud (those 38 vertices) in each frame
  - we further remove all edges connecting to the same character, as those edges do not contribute to the interaction

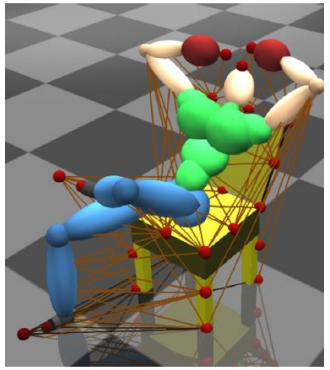




Creating the mesh representation

- For human-object interactions, the mesh is computed using the same approach but we consider the object as the second character
  - approximate the object using boxes
  - we uniformly sample the surface of the boxes using a predefined sampling distance, which is set as 20cm in our experiments
  - combining the vertices from the character and the object to generate the interaction mesh
- Existing work (Müller et al. 2009, Yun et al. 2012) and fully-connected meshes (i.e. connecting all vertices with edges) (Wang et al. 2012) suffer from the exponentially growing size of the feature. E.g.
  - If we extract one feature based on one vertex pairs for two characters of 38 vertices each, there will be (38*2)*(38*2 - 1) = 5700 features
- On average, each interaction mesh for two characters consists of only 170 edges





Sample motion data with interaction mesh

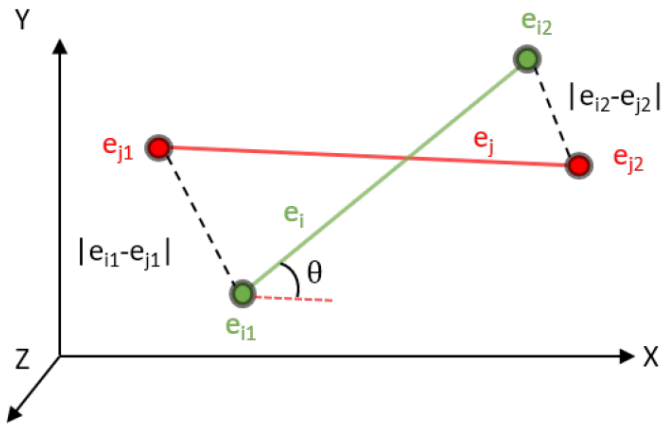
Distance between Interaction Meshes

- Using the interaction mesh representation, we can effectively represent interactions of different semantic meaning (e.g. punching vs. kicking)
  - Therefore, unlike previous research, our algorithm allows the comparison of two interactions with different semantic meaning
- We propose a distance function that adapts the Earth Mover's Distance (EMD) to find the best correspondence between the input interaction meshes
  - Such a distance function can effectively compare interaction mesh of different topologies and dimensions
- We will 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

Distance between Interaction Meshes

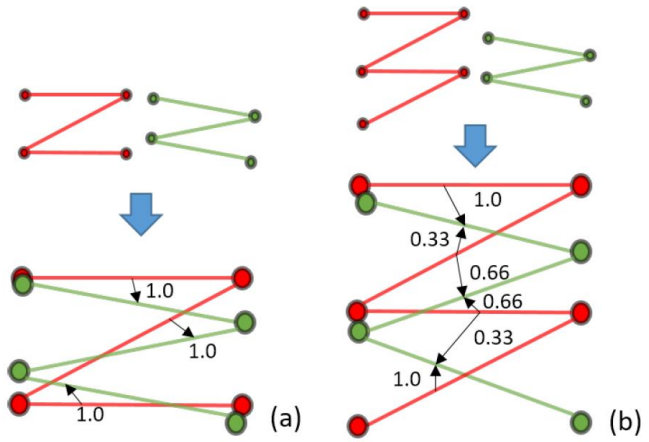
- Edge-Level Distance Function
  - Given edge e_i from interaction i and edge e_j from interaction j, we represent the difference between the two edges using a customized cosine distance function, which effectively combines the Euclidean distance and orientation distance between the two edges:

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



Earth Mover's Distance

- We then adapt a **mass transport solver** 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
- Illustrating the concept in 2D



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

subjected to:

$$\sum_{j=1} f_{i,j} = 1.0,$$

$$\sum_{i=1} f_{i,j} = \frac{n}{m},$$

Earth Mover's Distance

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

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

- 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

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

- 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:

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

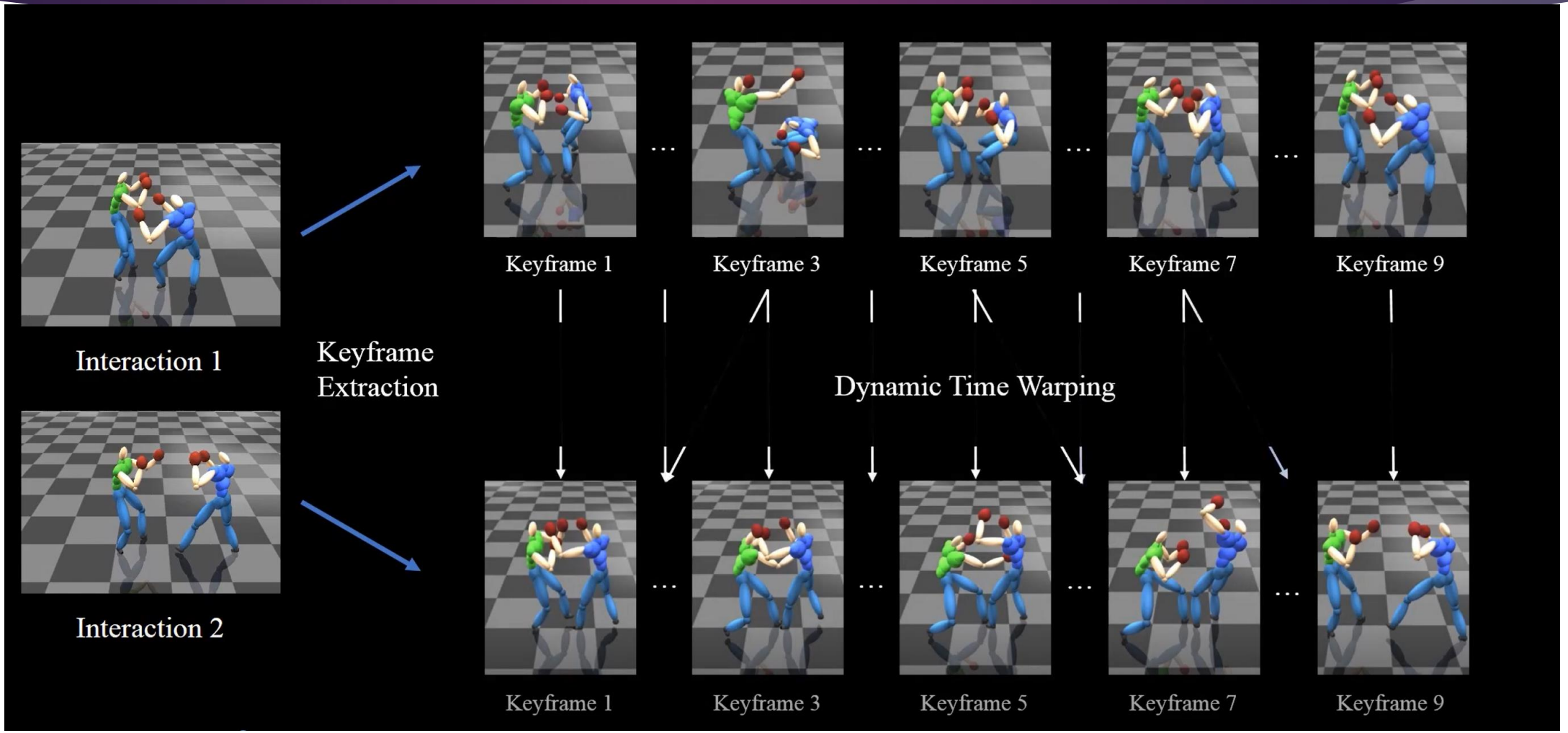
Distance between Interaction Sequences

- Spatial Normalization
  - We normalize interactions spatially to compare them with local coordinates, thereby eliminating the influence from different world coordinates
  - We consistently use 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

Distance between Interaction Sequences

- Temporal Sampling – converting motion into block-based key-moments
  - 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 the required number of the keyframes
  - The center of each block is then considered as a keyframe

Distance between Interaction Sequences



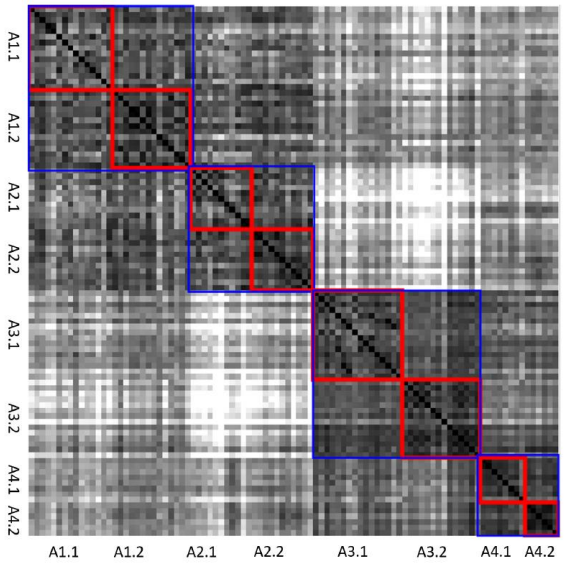
Experimental Results

Experimental Results

- We compare our method with
  - an interaction-based feature - **space-time proximity graphs** (Tang et al. EG2012)
  - traditional human-centered features including **joint positions** and **joint angles**
- 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 application with user-defined constrains.
- For the object in the HOI database, we represent the object as
  - a set of position for joint positions evaluation
  - a static simplified skeleton for joint angles evaluation

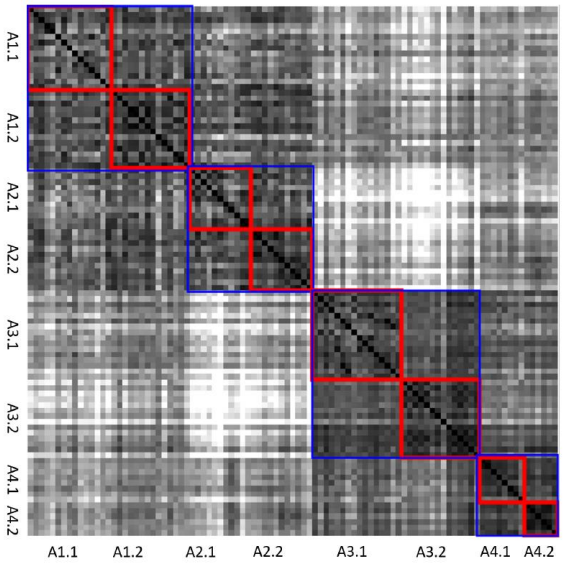
Interaction Similarity Analysis

- 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 tell *how different* two interactions are
- The last criterion is usually overlooked in existing approaches
- 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



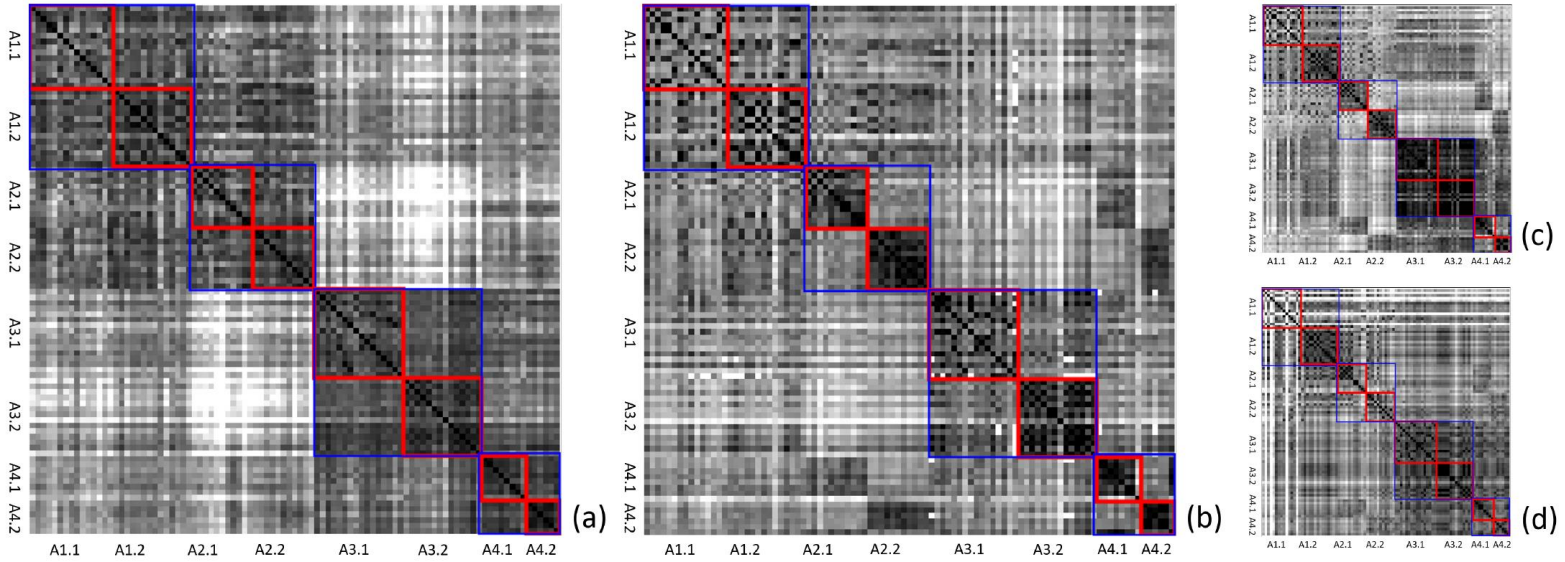
Interpreting similarity matrices

- The pixel color represents the normalized distance between two interactions – **Darker** pixels represent **higher similarity**
- Each individual class (highlighted by **red** squares) ***should*** show the highest intraclass similarity
- Classes belonging to the same attacking type (highlighted by **blue** squares) ***should*** show the second highest similarity
- Classes belonging to the same interaction type (i.e. A1.1-A2.2 and A3.1-A4.2) ***should*** show 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 ***should be*** smaller



2Character: Similarity matrices

- Comparing to STProximityGraphs, our method performs better in intra-class similarities, such as A1 and A3, in which one character punches the other
- Our method also performs better in identifying motion classes with the same attacking type (highlighted by blue squares)



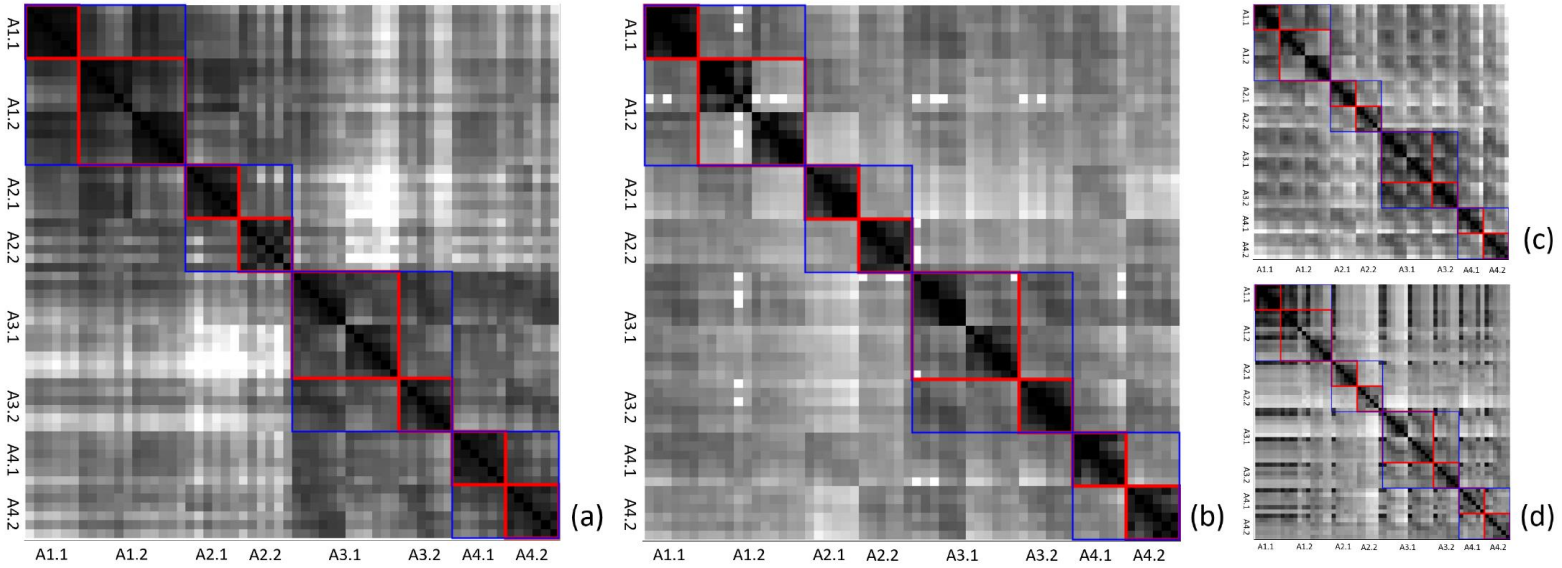
blue squares
red squares

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

Methods: a) EMD (ours), b) STProximityGraphs (Tang et al. 2012), c) Joint positions, d) Joint angles

CRC: Similarity matrices

- This is a challenging database as the interactions are edited during the retargeting process, which results in different 3D postures
- Both a) and b) show high intra-class similarity, but b) is less effective in evaluating the difference in semantically similar classes



blue squares

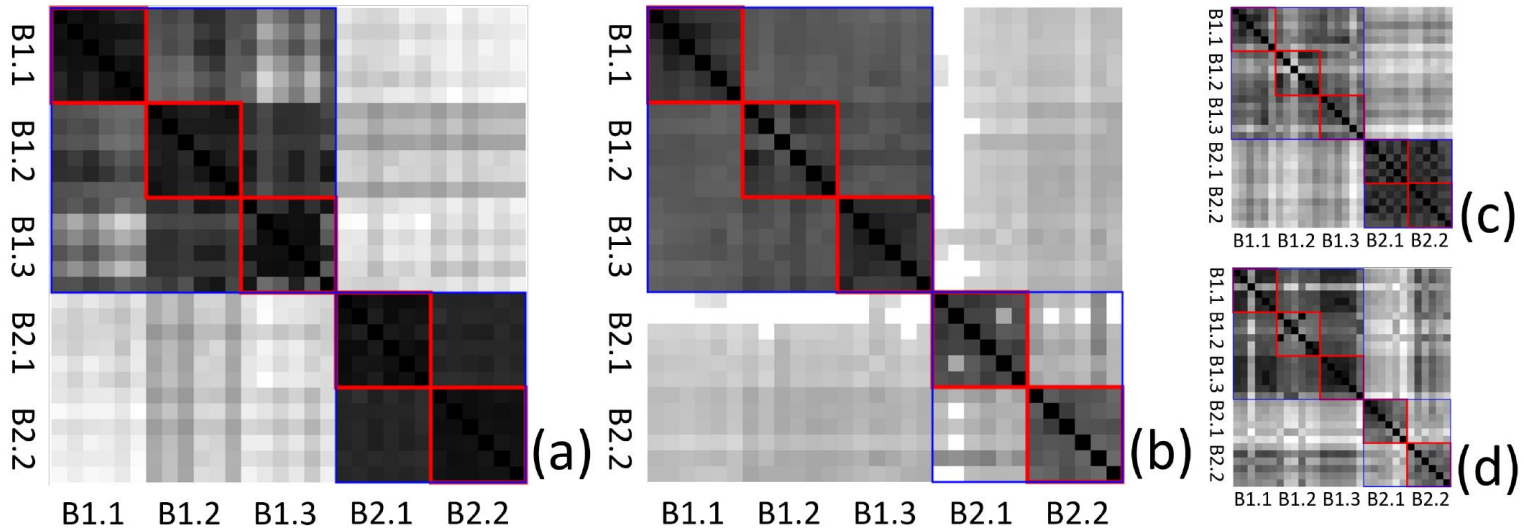
red squares

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

Methods: a) EMD (ours), b) STProximityGraphs (Tang et al. 2012), c) Joint positions, d) Joint angles

HOI: Similarity matrices

- 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



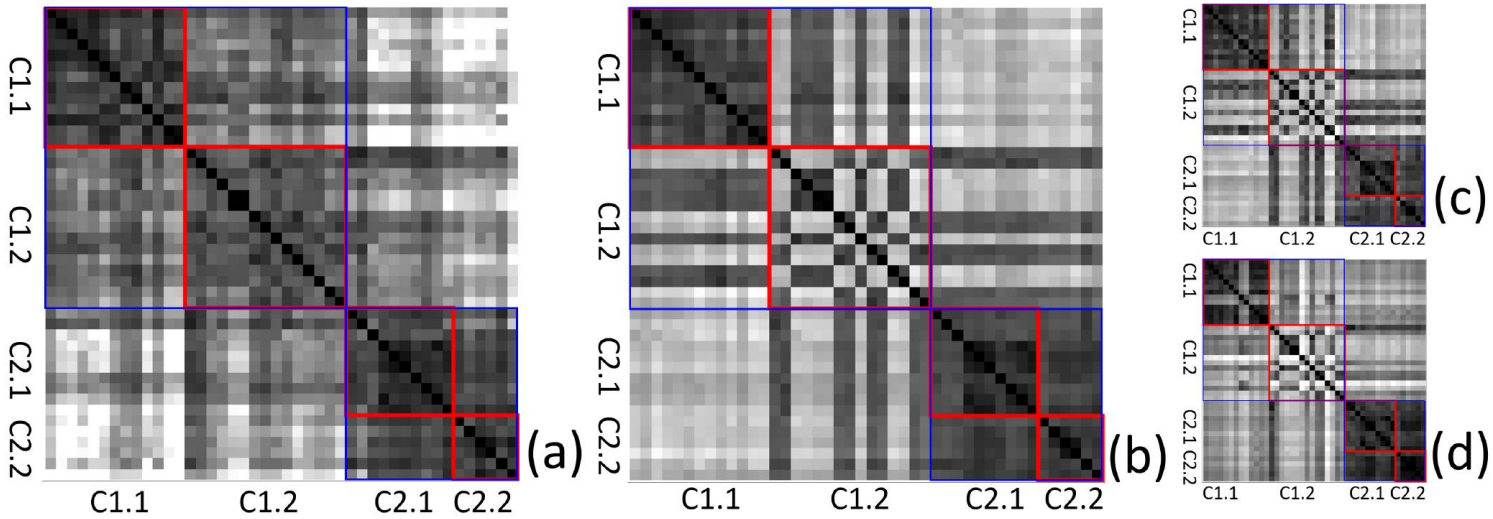
blue squares red squares

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

Methods: a) EMD (ours), b) STProximityGraphs (Tang et al. 2012), c) Joint positions, d) Joint angles

2PB: Similarity matrices

- Due to the complex, ambiguous real-people motion, the other methods fail to identify the intra-class similarity accurately, especially for C1.2 in which two boxers perform multiple punches simultaneously



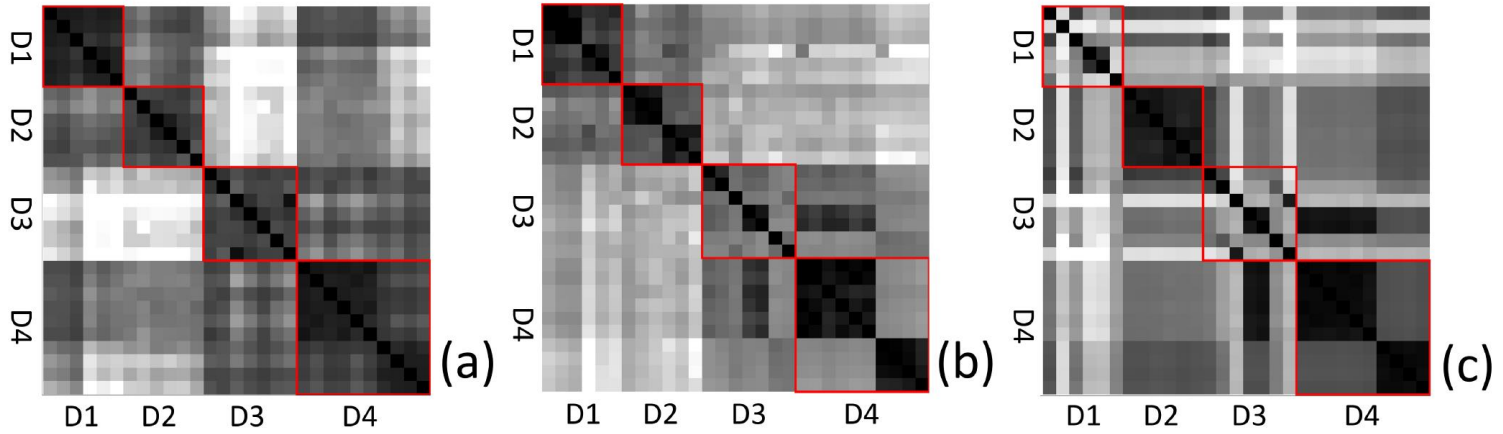
blue squares red squares

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

Methods: a) EMD (ours), b) STProximityGraphs (Tang et al. 2012), c) Joint positions, d) Joint angles

2PD: Similarity matrices

- Notice that this database is extracted from a public database where real-people motions are represented as point clouds
  - joint angle [10] evaluation is not available
- Since STProximityGraphs employs a binary function to determine the topology difference, it is very sensitive to small gesture changes

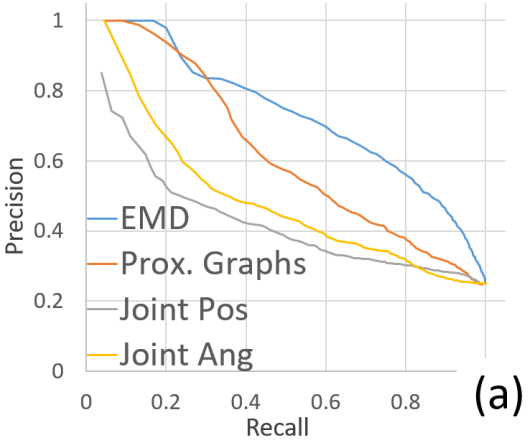


red squares

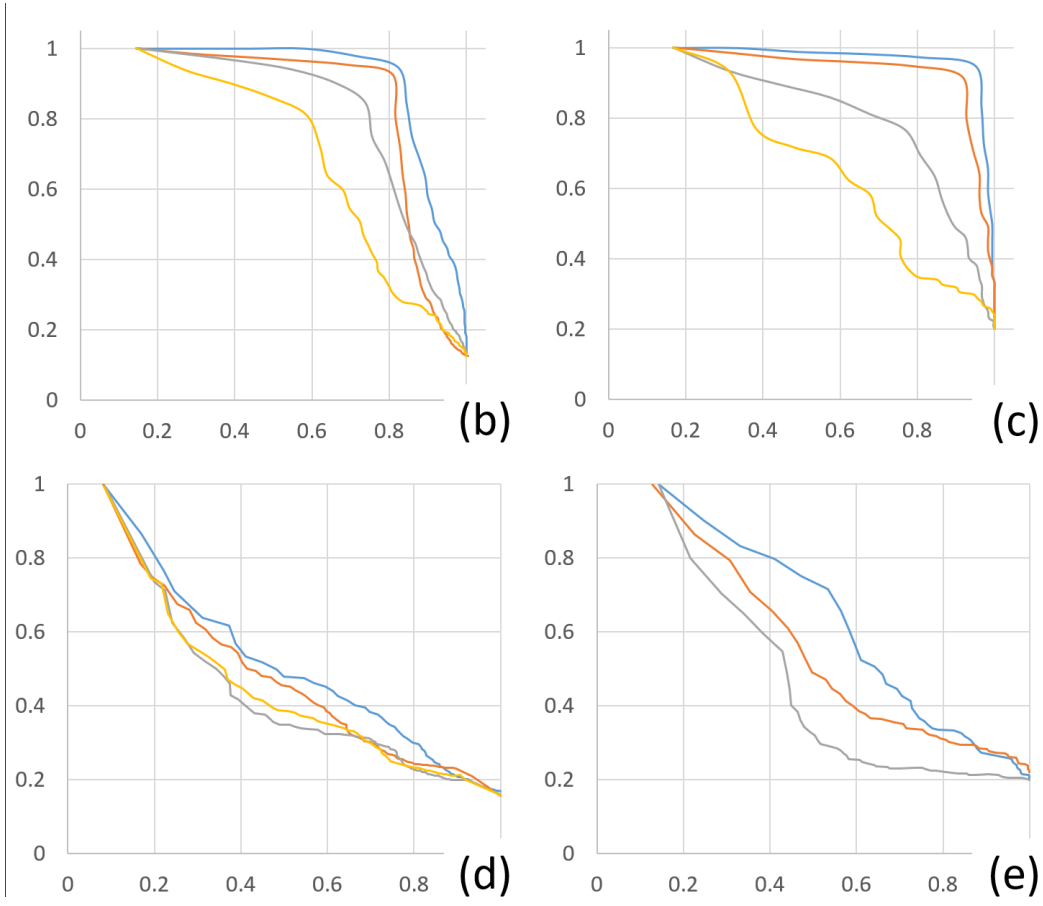
Interaction Type	Class
A and B Walk Around in a Circular Manner	D1
A and B Dance Together	D2
A and B Shake Hands	D3
A and B Chat with Each Other	D4

Methods: a) EMD (ours), b) STProximityGraphs (Tang et al. 2012), c) Joint positions

Interaction Retrieval Analysis



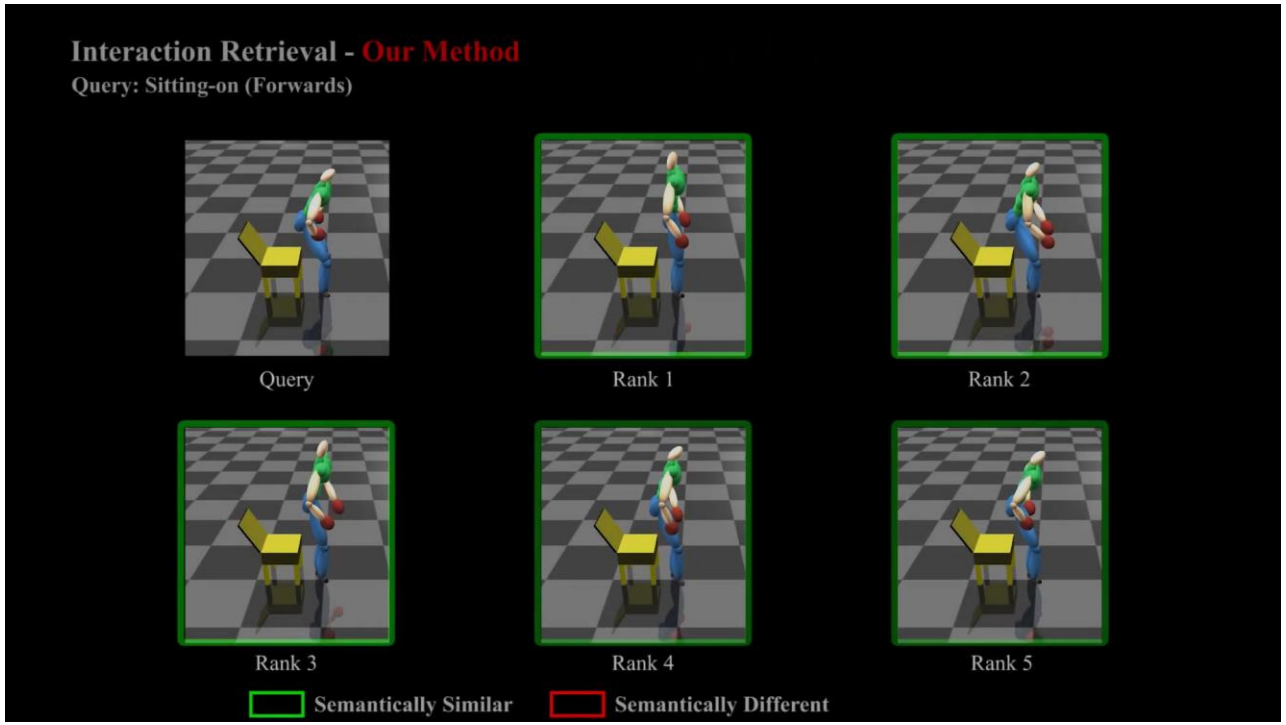
- We compare the methods using precision and recall
- We treat each interaction in the database as a query, and average the results from all queries to form the plot
- Given a query interaction, only the retrieved results within the same semantic lowest level sub-class are considered as relevant results
- Our method outperforms the others in all five databases



Databases: a) 2C, b) CRC, c) HOI, d) 2PB, e) 2PD

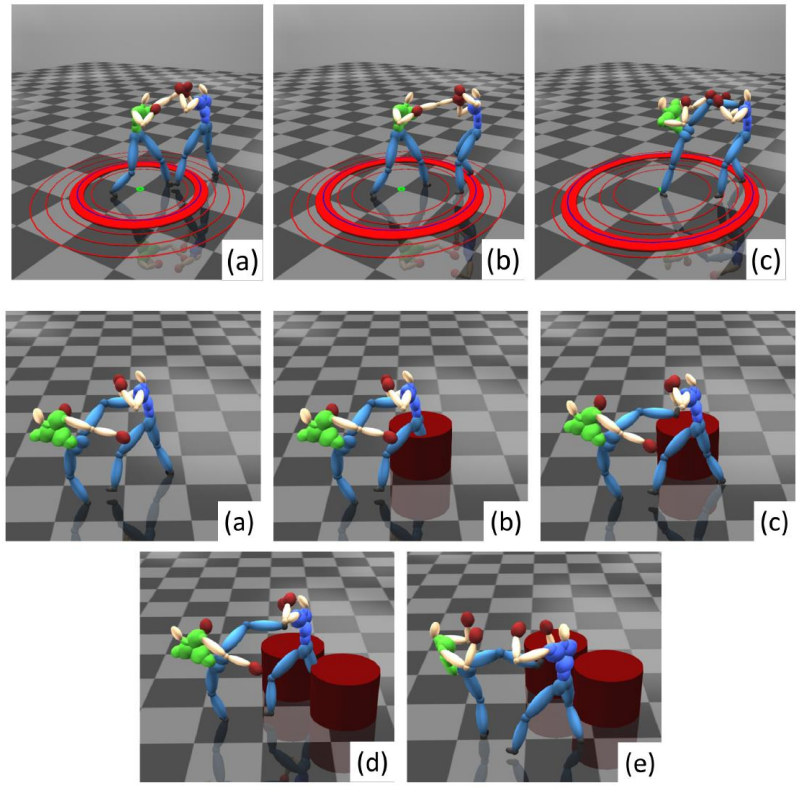
Retrieval results – some samples in 2C and CRC

Retrieval results – some samples in HOI



Interactive Retrieval Application

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



Interactive Retrieval Application

Conclusion

- We propose a new method for activity comparison from the interaction point of view
  - 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

Thank you!

- Any Questions?
- Email: e.ho@northumbria.ac.uk, Website: http://www.edho.net
- *By the way… we are organizing the ACM SIGGRAPH conference on Motion, Interaction and Games 2019 (**MIG2019**) http://mig2019.website*
  - *Full paper submission deadline: 22nd July 2019 (extended!)*