SkillVis: A Visualization Tool for Boxing Skill Assessment

Hubert P. H. Shum* Northumbria University
He Wang† University of Leeds
Edmond S. L. Ho‡ Hong Kong Baptist University
Taku Komura§ Edinburgh University

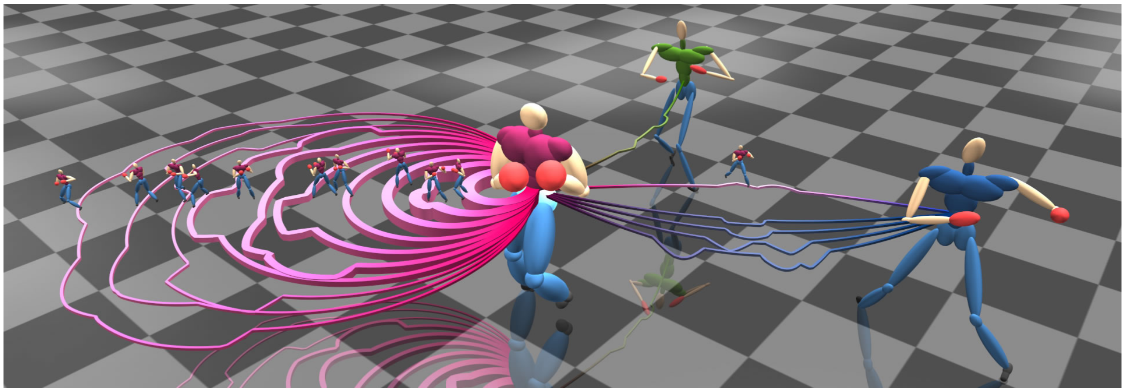

Figure 1: *Our system visualizes high-level boxing skills such as the richness of actions and the transition of them with a graph structure. A graph representation is employed with the characters of different sizes representing nodes with different sizes and edges representing groups of actions.*

Abstract

Motion analysis and visualization are crucial in sports science for sports training and performance evaluation. While primitive computational methods have been proposed for simple analysis such as postures and movements, few can evaluate the high-level quality of sports players such as their skill levels and strategies. We propose a visualization tool to help visualizing boxers' motions and assess their skill levels. Our system automatically builds a graph-based representation from motion capture data and reduces the dimension of the graph onto a 3D space so that it can be easily visualized and understood. In particular, our system allows easy understanding of the boxer's boxing behaviours, preferred actions, potential strength and weakness. We demonstrate the effectiveness of our system on different boxers' motions. Our system not only serves as a tool for visualization, it also provides intuitive motion analysis that can be further used beyond sports science.

Keywords: Motion Graph, Information Visualization, Dimensionality Reduction

Concepts: •**Computing methodologies** → **Animation;**

*e-mail: hubert.shum@northumbria.ac.uk, the corresponding author
†e-mail: h.wang4@leeds.ac.uk
‡e-mail: e.ho@northumbria.ac.uk
§e-mail: tkomura@inf.ed.ac.uk

MiG '16, October 10-12, 2016, Burlingame, CA, USA
ISBN: 978-1-4503-4592-7/16/10
DOI: http://dx.doi.org/10.1145/2994258.2994266

1 Introduction

Computers have been used in the fields of sports and health science to record and improve the performance of both amateur and professional athletes. There are computer-managed weight lifting machines and treadmills recording energy consumption or repetition achieved in every sports club. In the attempt to assist more professional sports activities, some researchers have used the virtual reality technology to create training systems in baseball [Komura et al. 2002], handball [Bideau et al. 2003] and tennis [Molet et al. 1999]. Nevertheless, the analysis of motions done in these technologies are usually on the low level: recording the timing of basic motions or comparing the trajectories with those by better players. On the other hand, human experts are good at evaluating performances by comprehensively capturing different features in the motions. For instance, they consider not only the instantaneous movements of the athlete but also the variety of motions and the ability of transition from one motion to another.

Consider boxing for example. Professional boxers are trained first on some basic postures such as defense, stepping and attack, threading through which are the transitions carried out by the boxer based on the strategy and the opponent in a match. A good boxer can carry out a variety of transitions at will to achieve the best outcome. Such information serves as indicators for assessing the skill level of a player and the same principle applies to many other sports such as tennis, fencing, etc. Unfortunately, there is a research gap in evaluating the motions of the players from a higher level point of view.

In this paper, we propose a robust method to visualize the skill level of a boxer's performance in terms of flexibility and richness of his/her motions. To begin with, we capture the training of a player, in which he/she moves alone in an open space and imagines to interact with an opponent. In boxing, this kind of training is known as shadow boxing. An experienced coach can easily assess the player's skills by watching the shadowing boxing motions. It is considered very important not only for training the skills, but also to warm up the body and get ready for further training with other players. This method greatly reduces the complexity for motion capture due to occlusion and collision, and has shown to be effective in our system for detailed player's skill evaluation.

We use the techniques and data structures in computer animation to process the motion data. The captured motion data are first automatically segmented into shorter clips with meaningful contexts and categorized into groups. Next, our system automatically generates a hierarchical motion graph structure known as *Fat Graph*, which uses nodes representing the postures of the body and edges representing the motion groups. With dimensionality reduction techniques, this Fat Graph can be visualized on a 3D space to evaluate the performance of the player. The transition capability of the player are visualized by the connectivity of the nodes, where the richness and preference of the motions is visualized by the edges in the graph. With the proposed algorithm, it is easy to identify the performance quality and potential problems of a player.

We demonstrate how we can easily evaluate the skills of different boxers with our visualization system, as shown in Figure 1. While we use boxing as our target sport in this paper due to its complex moves and strategic nature, our system can be applied to most activities that require swiftness, flexibility and creativity, such as tennis, fencing and basketball.

The rest of this paper is organized as follows. Section 2 discusses other related works about motion analysis and dimensionality reduction. Section 3 provides an overview of our system. In Section 4 and 5, we explain the algorithms to organize captured motion with a graph structure, and visualize generated graph. Related experiments can be found in Section 6. We conclude and provide some discussion for our proposed method in Section 7.

2 Related Work

Visualization of the skills of athletes, and hence helping them to improve, is a field that has not been fullly explored in the field of sports science. Existing research [Yeadon 1990; Yeadon] mainly focuses on how a motion will appear when parameters of the motion or the body are changed. For example, Yeadon [Yeadon 1990; Yeadon] has done research on how the diving and somersaults motions change when the motions are launched at different timings by using physical simulation. Although such tools are useful for the players to interactively visualize possible results under different parameters, they can only evaluate the performance of the sports that do not require complex maneuvers and strategies, such as jumping, high jumping, sky jumping, or somersaults.

In many sports games, the performance depends not only on physical factors such as velocity, power and strength, but also on flexibility to switch from one motion to another and richness of the player's motions. This high-level information has not been used to visualize the skills of the player in previous research. In this research, we combine the approaches of motion graph [Arikan and Forsyth 2002; Lee et al. 2002; Kovar et al. 2002; Lau and Kuffner 2005; Kwon and Shin 2005] and dimensionality reduction [Grochow et al. 2004; Shin and Lee 2006] to visualize high-level skills information of the athletes for the skill assessments.

Statistical approaches for analyzing the connectivity of different movements have been developed in the area of computer animation and pattern recognition. The Motion Graph approach [Arikan and Forsyth 2002; Lee et al. 2002; Kovar et al. 2002; Min and Chai 2012; Beaudoin et al. 2008; Arikan et al. 2003; Li et al. 2002] is a method to interactively reproduce continuous motions based on a graph generated from captured motion data. Reitsma and Pollard [2007] compared different motion graph techniques comprehensively. Heck et al. [2007] further parametrized the motion space to control how the motions are being generated by blending samples in the motion graph. Such an approach can be used for interactive character control such as those in computer games. When it comes to graph construction, [Min and Chai 2012; Beaudoin et al. 2008] are the most similar ones to our method. Min et al. [2012] grouped similar postures and transitions into nodes and edges. Their focus is the motion variety of synthesized motions so they used generative models to fit the posture and motion data. Our focus is about skill visualization through the analysis of postures and motions so we can afford simpler and faster methods for analysis. Beaudoin et al. [2008] cluster postures first then find motion motifs by converting the motion matching task into a string matching problem. Their priority is to find motifs that are representative while our focus is to visualize motion details and statistics to help people to assess the skills. Xia et al.[2015] constructed a series of local mixtures of autoregressive models (MAR) for modeling the style variations among different motions for real-time style transfer. They demonstrated style-rich motions can be generated by combining their method and motion graph.

Since the Motion Graph produces a lot of edges and nodes without any context, it becomes difficult to control generated motion as the user wishes. Safonova and Hodgins [2007] optimized the graph structure by combining motion graph and interpolation techniques to improve the performance. On the other hand, works to resolve this problem by introducing a hierarchical structure have been proposed in [Lau and Kuffner 2005; Kwon and Shin 2005; Shin and Oh 2006]. These approaches add topological structures into the continuous unstructured data so that the motion synthesis can be done at a higher level. In a sport like boxing, it is possible to create a motion graph of semantic actions such as attack and defence, which is known as the action-level motion graph [Shum et al. 2008; Shum et al. 2012]. A recent work by Hyun et al. [2016] proposed *Motion Grammars* to specify how character animations are generated by high-level symbolic description. Such an approach can be used with existing animation systems which are built based on motion graphs. Ho and Komura [2011] built a finite state machine (FSM) based on Topology Coordinates [Ho and Komura 2009] for synthesizing two-character close interactions. The sparse graph structure can be used for controlling the movement of virtual wrestlers in computer games. The purpose of these approaches, however, is motion generation rather than the visualization of the player's skill.

In our research, we adapted a hierarchical motion graph structure called the Fat Graph [Shin and Oh 2006] on the action level to analyze the connectivity and the variety of a captured motion set. In a fat graph, similar nodes are grouped together as fat nodes, and similar edges are grouped as fat edges, allowing better organization of motion data. The filtered motion graph is a variation of the Fat Graph, in which temporal relationship between poses are considered [Plantard et al. 2016b]. Such a structure, however, is targeted for motion reconstruction and analysis instead of visualization [Plantard et al. 2016a].

Dimensionality reduction methods have been proposed to visualize the overall structure of captured motions. For example, Grochow et al [Grochow et al. 2004] proposed a method to project the 3D motions of the human onto a 2D plane, and further reconstruct 3D motions by mapping arbitrary points from the 2D plane back onto 3D joint space. PCA [Shin and Lee 2006] and ISOMAP [Tenenbaum et al. ; Shum et al. 2010] are proposed to map the motions onto 2D planes. Due to the high variation of human motion, local PCA that considers only a relevant subset of the whole motion database in order to generate a locally linear space is proposed [Shum et al. 2013; Ho et al. 2013]. One can generate motions from arbitrary points on the plane by interpolating the postures of the original motion. Meanwhile, non-linear methods [Lawrence 2004; Wang et al. 2006] and Deep Learning [Holden et al. 2016] have also been used to reduce the dimensionality of motions. The Gaussian Process [Liu et al. 2016] and the mixture of Gaussian Processes [Liu et al. 2016] can be used to represent a set of human postures with a small number of Gaussian parameters. However, such methodologies did not take into the account the connectivity structure of the motions. We apply dimensionality reduction to our graph structure to visualize the connectivity structure of captured motions on a 2D plane.

3 System Overview

The system can be divided into two parts as shown in Figure 2: the motion organization system and the visualization system. The motion organization system captures, analyze and organizes the motion of a player using motion segmentation and motion graph techniques. The visualization system prepares graph layout by projecting entities to appropriate 2D position using dimensionality reduction techniques, and renders the resultant graph with interactive functionalities.

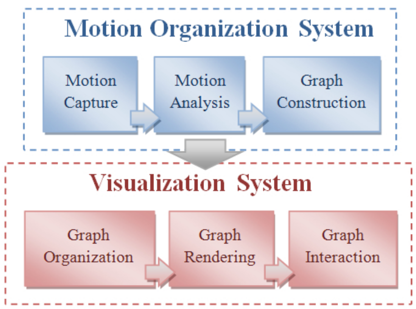

Figure 2: *The system is divided into the motion organization system and the visualization system.*

4 Motion Organization System

The motion organization system first captures the motion required for analysis using motion capture systems. Then, it segments the long sequence of motion into meaningful parts, which are used as building blocks of a motion graph. The system analyzes the similarity of these motion segments and constructs a Fat Graph structure that can be used to evaluate the skill level of the subject.

4.1 Motion Capture

The proposed algorithm can be applied to most activities that require swiftness, flexibility and creativity. For interactive games with multiple players, it would be the best to capture the motion of all players and evaluate them individually. However, even with the state-of-the-art technology, capture motions of multiple players remains difficult due to occlusion and collision among players. Therefore, we proposed to capture the training motion of the players, in which the player moves alone in an open area, imagining to interact with an opponent.

In boxing or any other martial arts, there is a practice called “shadow boxing”. The boxer imagines another boxer standing in front of him/her and repeats the techniques that he/she has been practicing. The boxer launches not only motions such as punching, but also defence, stepping, and the consecutive combination of all such motions. There are similar practice methods in basketball and soccer as well. The players can use the ball to conduct various techniques in the court imagining that their opponents are trying to take away the ball from him/her. The players thus perform various motions to keep the ball and trick the imaginary opponent.

During experiments, we used optical motion capture system to acquire the performed motion (Figure 3). This is because when compared to magnetic and mechanic motion capture system, optical systems produce the least disturbance to the player. Also, we preferred to capture long and continuous clips for the same reason.

Figure 3: *The shadow boxing motions of several boxers were captured using an optical motion capture system.*

4.2 Motion Analysis

We have developed an automatic motion analyzer to segment and classify raw motions into shorter, meaningful motion clips. This is done by analyzing the supporting foot condition, the acceleration profile of joints, and the trajectories of the effective joints.

Here we define the term ”motion” as the raw captured data, and the term ”action” as a semantic segment of the motion we captured. In the field of boxing, an action can be an attack (such as a ”left straight”, ”jab” or a ”right kick”), a defense (such as ”parries”, ”blocking” or ”ducking”) , a transition (such as ”stepping to the left”, ”stepping forward” or ”back step”), or any combination of them.

We observed that most actions start and end at a double supporting state (i.e. both feet touching the floor), which can be detected by monitoring the feet height and velocity. The raw captured motion is segmented into a set of *movement segments*, which are the periods between every two successive double supporting states (Figure 4 Upper).

We also observed that a relatively large amount of force is exerted during any actions such as a punch or a step. The periods with high-level of force exertion is called the *activity segments*. Since the force is proportional to acceleration, these segments can be found when the sum of squares of acceleration of all joints is above a threshold. The threshold is statistically obtained from the acceleration plot of the body (Figure 4 Middle).

The actions are composed by using the movement segments as the building blocks. The timing and the duration of the activity segments are used to determine if the movement segments should be merged together to form longer segments. Regarding the relationship of the movement segments and the activity segments, there could be three possible cases: (1) There is no activity segment inside a movement segment. In this case, the movement segment becomes a single transition action. (2) There is one activity segment inside a movement segment. In this case, this movement segment becomes an action with a special activity. (3) There are more than one activity segments lying across successive movement segments. In this case, the movement segments containing activity segments at the border are merged to form an action (Figure 4 Lower). Note that due to this merging process, the resulting action could contain multiple activity segments. We also filter very short actions that are likely to be generated due to the noise of the supporting feet.

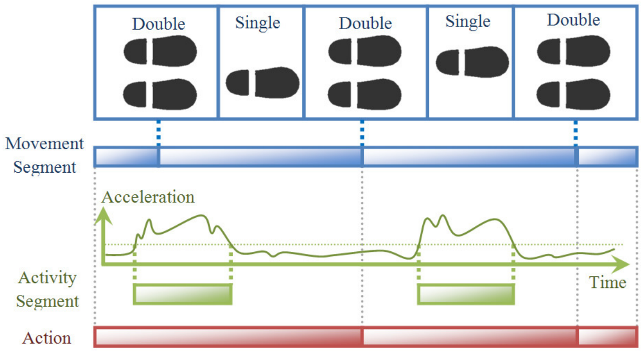

Figure 4: *Upper: The movement segment is defined as the period between two double support supporting phases. Middle: The activity segment is defined as the period with high acceleration. Lower: The action is the combination of movement segment and activity segment.*

Here, we define the *effective joints* to be the set of joints to represent an activity segment. In case (1), since the actions contain no special activities, the pelvis is considered to be the effective joint. However, in case (2) and (3), the effective joint is the joint that contributes the most to the sum of squares of the acceleration in the activity segment. In more complicated actions such as left-right combo punches, there may be multiple effective joints. Such joints are used in the later process to evaluate the similarity of actions.

4.3 Graph Construction

We apply the Fat Graph structure to organize the captured motion. Since Fat Graph is originally proposed for motion synthesis, it is not optimized for skill visualization. We redesign the algorithms to generate nodes and edges in the Fat Graph for our purpose.

The nodes of Fat Graph, known as Fat Nodes, are common starting or ending poses of actions. An unsupervised clustering scheme is used for grouping them into a finite set of pose groups, each represented by the mean pose of the group. In this way, we do not need additional labor such as labelling. Specifically, we used k-mean to cluster postures. The distance between two postures is the Euclidean distance between their respective joint angles. Regarding the cluster number k, a large k would result in many clusters (Fat Nodes) which unnecessarily increases the complexity of the graph. A small k will cluster very different postures into the same node, defeating the purpose of the graph construction. Therefore, we set up a posture difference threshold empirically based on experts' suggestions. Then, we iteratively search for a proper k by initially setting $k = 1$ and incrementing k by 1 until we find the first k that does not violate the distance threshold.

After clustering, the poses representing the Fat Nodes are the standard poses that the player can start various motions from. In the case of boxing, they are usually the fighting poses that the boxer guarding his/her face against the opponent, with both feet landing on the ground and keeping apart in shoulder distance. By evaluating the Fat Nodes, it is possible to tell if a boxer has multiple unnecessary standard poses, or if any standard poses contain potential weakness.

The edges of a Fat Graph, known as Fat Edges, are directional edges that represent groups of similar actions. Each edge points from the Fat Node representing the starting pose to that representing the ending pose. In our implementation, the Fat Edges are represented by the action group classified by a similar scheme introduced for Fat Nodes construction. We apply the same algorithm as in the Fat Node, in which we use k-mean to cluster the actions and search for the smallest acceptable k for a given distance threshold. The only difference is that instead of using posture distance, the actions distance is defined according to the trajectory of the effective joints as explained in Section 4.2. This allows accurate clustering of actions and ensures that the effects of the effective joints are not smoothed out by other joints.

Formally, the distance between two action A_0 and A_1 is defined as:

$$D(A_0, A_1) = \begin{cases} \infty \\ \text{if sequences of effective joints are different} \\ \displaystyle\sum_{j=0}^{j_{total}} \sum_{f=f_{start}}^{f_{end}} [A_0(j)(f) - A_1(j)(f)] \\ \text{otherwise} \end{cases} \tag{1}$$

where A_0 and A_1 are actions defined by 3D position in terms of joints and frames, j and j_{total} are the joint index and the total number of effective joints in the sequence of effective joints inside an action, f, f_{start} and f_{end} are the frame index, starting frame and ending frame of the considering effective joint. The terms $A_0(j)(f)$ and $A_1(j)(f)$ represent the 3D positions of joint j in frame f for the corresponding action. In case two effective joints with different duration are to be compared, the shorter one is linearly scaled to the duration of the longer one.

Actions with small distances calculated from Equation 1 are grouped together to form an action group. In the field of boxing, an action group could contain a set of actions with basic attacks or defences such as "straight punch", "hook punch", "parry", or a set of complex actions combining several attacks and defences. Since member actions in a Fat Edges share the same starting and ending Fat Nodes, if an action group contains multiple starting or ending poses, it is sub-divided. In general, novice players normally has fewer Fat Edges since they have less experience and hence does not acquire enough techniques to perform in a match. The relationship of Fat Nodes and Fat Graphs is demonstrated in Figure 5.

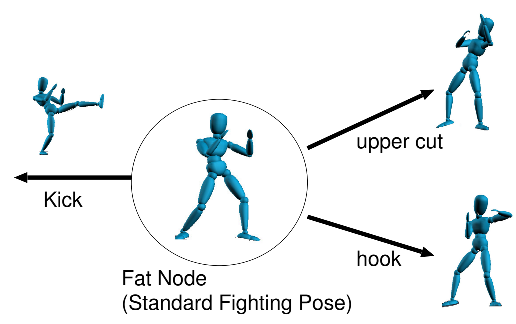

Figure 5: *The Fat Node represents the standard fighting pose. The three outgoing edges represent different action groups.*

4.4 The Skill Index

It requires deep knowledge and years' experience to assess one's skills in sports. For the sports in interest, there are two important indicators. The first one is the richness of the actions that indicates the resourcefulness of a player. A top player has more than one way to achieve the same goal where the choice depends on the situation.

The other is the flexibility of transitions between states so that the player can switch between different states at will. Our graph representation captures both of the indicators. The richness can be represented by the number of Fat Edges between any two Fat Nodes indicating how many kinds of maneuvers, each with variations, the player has for transitions from one state to another. The flexibility is indicated by the connectivity of the graph. A fully connected graph shows great flexibility because there are transitions between any two nodes.

However, these two factors are somehow contradicting. In general, the richer the actions are, the greater the number of different starting and ending poses is hence the poorer the connectivity of actions is. Independently considering either of them would not suffice. Therefore we define a *Skill Index*, *S*, that evaluates the skill level of the player.

The richness of action is represented by the number of Fat Edges in the Fat Graph, while the connectivity is inversely proportional to the number of Fat Nodes. The Skill Index is defined by:

$$S = \frac{\text{Number of Fat Edges}}{\text{Number of Fat Nodes}} \tag{2}$$

In our implementation, we do not consider the Fat Nodes that are not intentionally created. For example, one of our boxers tripped over during a session. While it is good that our system can objectively pick up the posture generated by the accident, we do not include the corresponding Fat Nodes when calculating the Skill Index. Also, we do not consider Fat Edges that contain only one action to make sure this edge is not some randomly performed action.

As an example, experienced boxers could perform a large variety or actions while maintaining the connectivity of actions by limiting the number of starting and ending poses of actions. In this case, the Skill Index of the player will be very large.

5 Visualization System

The graph representation explained above consists of high dimensional Fat Nodes (groups of similar postures of many degrees of freedom) and Fat Edges (groups of similar actions), which presents a challenge for visualization. To decrease the dimensionality for better visualization, we propose two different algorithms for nodes and edges because of their different nature in this graph. Specifically, we project the nodes on a 2D space and represent the edges with curves. For Fat Nodes, we apply Principal Component Analysis (PCA) as it creates a more consistent low dimensional space comparing with other methods. For Fat Edges, we apply PCA on high energy postures of the actions, and use a combination of geometric primitives to visualize the action features.

5.1 Visualizing Fat Nodes

Although the degree of freedom (DOF) of human poses are normally in high dimensionality (45 DOF in our system), they are intrinsically dependent on each other. In fact, the Fat Nodes can be represented effectively in a 2D space where nodes of similar poses are located together while that of different poses are located far apart. By this representation, viewers can easily understand the relationship among postures. In this section, we briefly describe the progress of projecting high dimensionality poses in Fat Nodes to low dimensionality viewing plane.

We define a posture as a vector of 3D positions of joints, each of which is computed as its 3D position with respect to the position of the pelvis. Suppose i_{Total} is the total number of joints, a pose space P can be defined as $P = \{J_i\}$ where $i \in [0, i_{Total}]$. Each dimension in the pose space P represents a DOF, and the j^{th} dimension in the pose space is denoted as P_j, where $j \in [0, 3i_{Total}]$ and $3i_{Total}$ is the total number of DOF.

Given a set of zero-meaned poses $p \in P$, it is possible to calculate the covariance matrix C of size $3i_{Total} \times 3i_{Total}$ to evaluate the intrinsic dependency of the dimensions in the pose space. The element at m^{th} column and n^{th} row of matrix C is defined as:

$$C_{m,n} = cov(P_m, P_n) \tag{3}$$

where $cov(P_m, P_n)$ denotes the covariance between the m^{th} and n^{th} dimension in the pose space.

We calculate the eigenvectors from C. The eigenvectors represent orthogonal dimensions that form a projected space, and each of them comes with an eigenvalue. Since we wish to project the human pose onto a 2D space, we select the two eigenvectors with largest eigenvalues, which indicate the variance of data in the corresponding eigenvector dimension, to form a feature vector.

$$F = [eig_1, eig_2] \tag{4}$$

where eig_1 and eig_2 are the two eigenvectors with largest eigenvalues. Once the feature vector is calculated, an arbitrary pose p can be projected onto the 2D plane by:

$$p' = F \times p \tag{5}$$

where p' is the projected 2D coordinate.

We obtain the mean posture of each Fat Node, and render it with a humanoid character at the corresponding 2D position using Equation 5. We use the size of the character to represent the number of poses that are classified into the node. The more muscular/bigger the character is, the bigger the node is (Figure 6). In this way, one can easily observe the poses that the player normally used to start actions, and hence identify the incorrectly performed poses. For example, in boxing, novice boxers sometimes lose tracking of their boxing rhythm, and hence start or end a punch with an inappropriate posture.

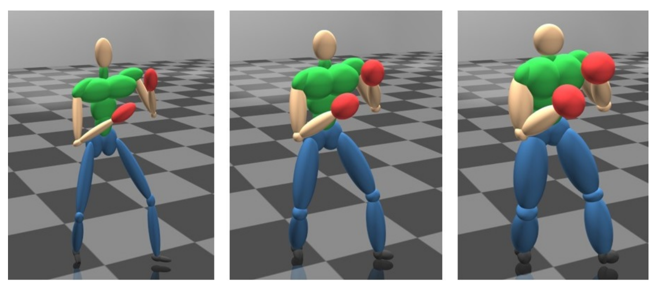

Figure 6: *From left to right, the character becomes bigger and bigger as the size of the nodes goes up.*

5.2 Visualizing Fat Edges

Fat Edges contain information of groups of similar actions, they can represent motion variation, player resourcefulness, etc. We cannot apply dimensionality reduction purely based on the action data itself because the low dimensional projection would be very complex. We propose to visualize each Fat Edge by a 2D curve that connects the starting and the ending Fat Nodes.

We represent the number of actions in the edge by the thickness of the curve. It shows the frequencies of different actions which indicates the player's preference of specific actions. For instance, if a boxer heavily relies on single straight punch, the Fat Edge for such action will be unreasonably thick, while edges for other attacks such as hook punches will be relatively thin, which shows the potential problem of the lack of diversity of attacking strategies.

Finally, to visually distinguish between different actions, we add some geometric patterns on the 3D curve. We collect the high-energy frames of all actions and use the PCA method explained in the last section to project them onto a 1D space. Since the high-energy frames of different actions are typically distinguishing postures, the projection essentially maps all action features onto a normalized 1D space, denoted by $I \in [-1.0, 1.0]$. We specify some geometric patterns to represent values in this 1D space. In particular, we design some patterns for landmark values -1.0, -0.5, 0, 0.5 and 1.0. The patterns to represent values between two landmarks are obtained by linear interpolation. The patterns for landmark values in our system are shown in Figure 7 Upper. Given a Fat Edge, we first obtain a mean action and its corresponding high energy postures. We then obtain the 1D representation of those postures and place a corresponding pattern on the edge. Through the comparison between Figure 7 Lower Left and Lower Right, it shows that adding the geometric patterns gives a better visualization of actions in the edges. This strategy presents an intuitive way to show the players preferences over actions of different complexity.

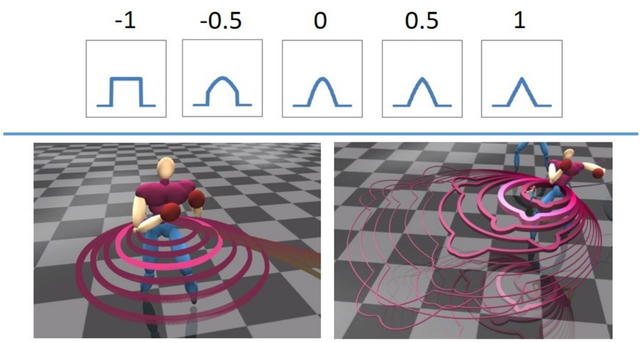

Figure 7: *(Upper) The geometric patterns for landmark values between -1 and 1. (Lower) Comparison of visualization without/with the patterns.*

Although we use Fat Edges to represent actions in groups, there may still be a lot of edges in a graph. It is essential to organize them in a neat way to avoid overlapping of edges. We organize the edges in a way to avoid occlusion. For edges with starting node different from ending node, the edge direction is fixed. The only adjustable variable is the bending side of the curves, which is essentially the sign of the curves. On the other hand, for edges with starting same as ending node, the edge direction is undefined. In other words, the direction of these edges can be any angles in the X-Z plane. In both cases, we select signs and angles such that the edges would blend towards less density region of the graph.

5.3 Interactive Features

We integrate some interactive features in our system to incrementally display relevant information based on the user input. The user could interact with our system using standard input device such as mouse and keyboard. When the user selects any specific entities in the graph, related information will be shown.

For example, when a Fat Node is selected, its corresponding Fat Edges will be highlighted. Information about the number of members in that node, number of outgoing edges, and number of incoming edges are displayed in a sub window. On the other hand, when a Fat Edge is selected, we render the action included in the edge one by one, such that the user can visualize the content of the edge.

6 Experiments

In this section, we present experimental results. We captured motions of four boxers, with different skill levels. We first give detailed motion analysis and visualization of individual motions, then compare them side by side. They demonstrate that our system is an effective tool for motion analysis, skill assessment and comparisons. As it is difficult to show the motions in pictures, we strongly suggest that the readers watch the supplementary video for more details.

The four boxers chosen have different skill levels. As a ground truth, they were evaluated and labelled as Skillful, Medium, Medium and Novice, denoted by S, M2, M1 and N. We show the subjects and their Fat Graphs respectively. To fully explain the features in the visualization system, we first show the Fat Graph of Boxer S in figure 8.

6.1 The Visualization System

In Figure 8, there are three Fat Nodes indicated by red arrows and numbered as 1, 2 and 3, each visualized as a character with a mean posture in the node. The sizes of the nodes are indicated by the body shapes. Node 1 obviously has the most muscular character which means its node size is the biggest. 2 and 3 are far smaller. Fat Edges are rendered as curves between nodes such as the ones shown by 4 and 5. As we explained, the thicknesses of the edges indicate the frequency of the actions taken. 5 is much thicker than 4 suggesting this boxer takes action 5 more often. In addition, an edge can be smooth like a circle or bumpy with geometric patterns shown in 7 and augmented into 3D. A single pattern means one activity segment, e.g. a left punch while multiple patterns mean a series of activities such as a combo attack.

In addition to the basic features, our system supports interactive demonstrations. Figure 8 is the result when the mouse hovers over the Node 1. All the edges starting from this node are highlighted, each with a small character performing the action on it. It gives the user the flexibility to look at the actions from that node. Also, if the screen looks too jammed with all actions performed at the same time, the user could move the mouse onto a specific edge where our system will only render one character performing that action on the edge.

6.2 Boxer Evaluation

Next, we show what assessments we can make by looking at the Fat Graph of Boxer S. First of all, a good boxer always tries to stay in a defensive mode whenever he/she is not attacking or dodging. Even after making an attack or dodge, he/she is supposed to resume the defense mode. From this point of view, Boxer S has a good grasp onto the principle. It can be seen in that Node 1 is the standard defense posture and also the biggest node which means Boxer S stays in the defense mode most of the time. Also, most of the edges leave and go back to it, which means no matter what actions Boxer S takes, he returns immediately to defense mode. Second, there are a good number of actions starting and going back to Node 1, which indicates Boxer S is resourceful and has mastered a variety of actions for different purposes. By looking more closely at the edges, we number the innermost edge as 1 and increment the index outwards. It can be easily seen that edge 1-3 are smooth circles without any geometric patterns, showing they are simple stepping strategies, which means Boxer S moves a lot while waiting for the right timing

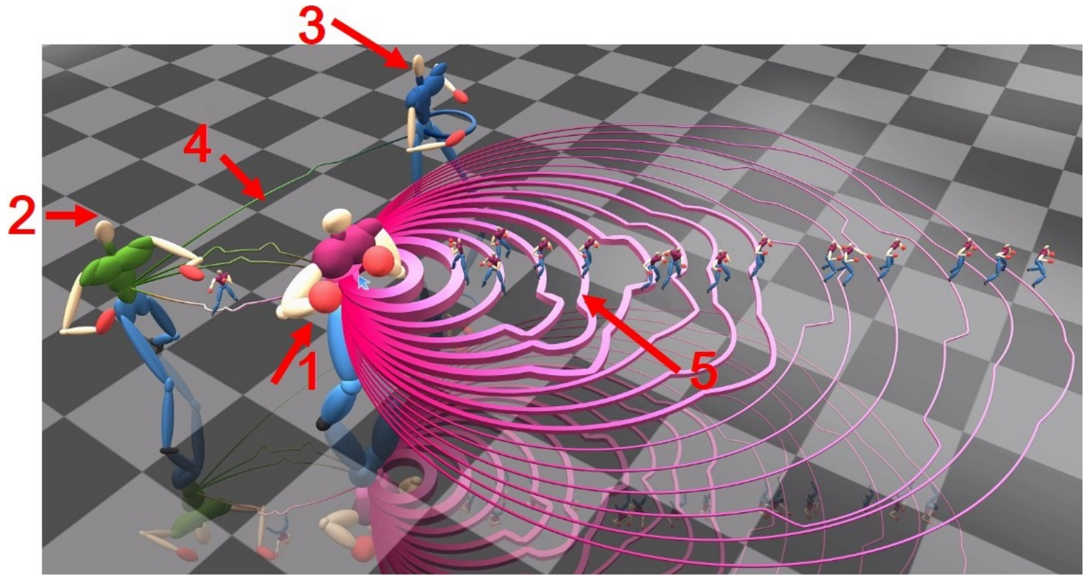

Figure 8: *The Fat Graph of Boxer S. 1, 2 and 3 are Fat Nodes. 4 and 5 are two Fat Edges. 4 connects Node 2 and Node 3. 5 connects Node1 to itself.*

to attack. Starting from the 4th edge (the one pointed by Figure 8 5 is the 5th edge), there are many attacking actions shown. Interestingly, the 4th and 5th edges have only one geometric pattern while the 6th and onward edges have more, showing Boxer S prefers simple attacks such as one punch and use fewer combo attacks because edge 4 and 5 are thicker. For a skillful boxer, it makes sense because complex attacks also expose the attacker himself to attacks (due to that it takes a longer time to resume the defense posture), they are usually less used.

Besides, we can also spot the skill flaws in the Fat Graph. Node 2 is a deformed defense posture because the two hands are lower than they are supposed to. This node is much smaller node and there are only a few edges between Node 1 and Node 2. It means Boxer S stays in Node 2 sometimes which exposes his head to attacks. This is a typical mistake made by even professional boxers especially when they are tired or focusing on looking for opportunities for attacking. In fact, raising the hands for defense is a common advice that boxing coaches give. Node 3 is a special case. There is just one thin edge connecting Node 2 to Node 3 because it was when the Boxer tripped over accidentally during the motion capture. We show it here for completeness.

Through Figure 8, one can see that our system provides intuitive visualization of boxing motions including posture preference, action variety, action preference, skill flaws, etc. Next, we also show the Fat Graphs of the other three boxers in Figure 9.

From Figure 9, one can intuitively understand why their respective skill levels are so. The top one is a novice boxer. The graph has five nodes with four of them being almost equally muscular (the four near the center). It means the boxer transits among them most of the time and all of them except for the green node in the center have ineffective defense postures, e.g. hands are too lower or too apart, making him very vulnerable in boxing. Also, The green one is the relatively main posture, in which the leg is wider apart. The blue one is a secondary posture with narrower leg distance. The red and the orange are less used postures. Leg movement is important in boxing to efficiently launch actions. The postures with wider leg distance are considered inferior, as they limit the ability for a swift evasive move.

The middle is Boxer M1 who has medium skills but slightly worse than Boxer M2. He apparently knows about the principle of holding a defense posture all the time as shown that by the purple character in the middle which is the biggest node. Also, most of the actions leave and go back to this node indicating he is aware that after an action, he is supposed to resume the defense posture. However, overall there are many more nodes. Although as relatively small as they are, it means there are times Boxer M1 forgets about defense posture holding. In addition, the purple main posture has a large variety of actions. However, it is also obvious that some actions are too complex and are only conducted once or twice. This means that the boxer lacks consistency in boxing techniques. The blue posture is a subtle preparation movement for right punch. This should be avoided as the opponent can tell his move when seeing such a posture.

The bottom is the Boxer M2 whose skill level is the closest to Boxer S. Their Fat Graphs also look very similar which shows the consistency of our system. Boxer M2 rarely leaves the defense posture shown by the purple character in the middle of 9 Bottom. However, the biggest difference between him and Boxer S is the actions. The boxer has a large number of locomotion. The first edge (the innermost one) is a circle which indicates a stepping. Note that he relies too heavily on rightleft combo (the second edge), which should be avoid as the opponent could take advantage on such a frequently launched combo. There is stepping in different directions shown by the third and fourth edge. Then, there are some left punches (the fifth edge). The boxer also has a number of other boxing combos to enrich the variety of actions. The green and the blue postures represent should be avoided, as they derive from the core, purple posture. Only limited actions can be started from them.

6.3 Graph Statistics

Finally, we give some statistics about the four Fat Graphs in Table 1. We calculate the Skill Index using Equation 2, and it aligns with the boxers' skill level. Notice that when calculating the Skill Index, we do not consider Fat Nodes that are unintentionally generated such as tripping over accidentally during the capture section. We also do not consider Fat Edges that have only one member action

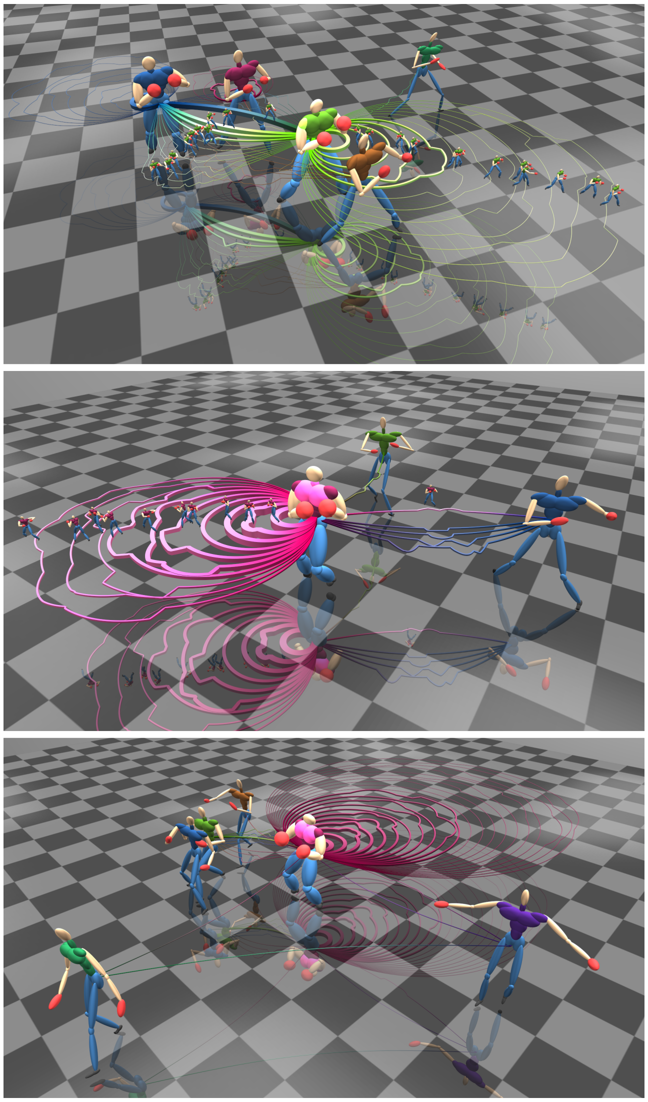

Figure 9: *Top: Boxer N, the novice boxer. Middle: Boxer M1, the boxer with basic skills. Bottom: Boxer M2, the boxer with medium skills.*

to ensure that the evaluated action can be conducted by the boxer consistently.

7 Conclusions

In this paper, we proposed a method to visualize the skills of athletes using Fat Graphs from a higher level point of view. With our algorithm, the flexibility and the richness of the motions can be clarified. In our motion organization system, we introduced a generic motion segmentation and classification method to analyze raw captured sports motions. We also applied a hierarchical motion graph called Fat Graph to arrange segmented motion in a meaningful way. Then, in our visualization system, we utilized dimensionality reduction techniques to project the Fat Graph onto a 2D plane, and rendered it in a 3D space with special 3D features.

In our experiments, we showed the differences between the Fat Graphs of a novice player and that of an experienced player: the former is much sparser and less connected, while the later is dense and well-connected. We made use of the proposed skill index to objectively evaluate the skill level of players in terms of flexibility and connectivity of motions. Moreover, we discussed some of the potential problems of the sports players by analyzing their corresponding Fat Graphs. This information is useful for the players to check their performances and plan their future training.

	Boxer N	Boxer M1	Boxer M2	Boxer S
PN	176	160	112	138
FNN	6	6	3	3
AN	88	80	56	69
FEN	57	36	16	20
SI	1.3	3.0	2.3	5.0
SL	Novice	Medium	Medium	Skillful

Table 1: *Statistics of the boxing motions. PN: Posture Number. FNN: Fat Node Number. AN: Action Number. FEN: Fat Edge Number. SI: Skill Index. SL: Skill Level.*

In this research, we focus on analyzing the skill level of the boxers in terms of high-level motion behaviour such as the richness of the action and the transition of action. We do not evaluate the lower-level parameters such as the speed of the punches, which has been explored in previous works. It is an interesting future direction to combine both high-level and low-level evaluation in order to have a full assessment of the boxers.

There are limitations to our method. First, our method is based on the assumption that the sports skills mainly consist of a finite number of key postures and transitions in between. Admittedly, not all sports follow this pattern. Second, the visualization and skill assessment is based on individual athlete, not considering skills related to collaborations such as those in group sports, in which the assessment might need to employ different criteria.

In the future, we wish to extend the proposed algorithm to the field of computer animation. Currently, when synthesizing animations by motion graphs, experienced animators are required to tell what motions are missed or badly captured. With our system, it is possible to analyze the connectivity and variety of a motion set, which are two critical factors in motion synthesis. However, how to generalize these findings to give high-level suggestion, such as proposing the motions to capture, remains an open problem. In addition, we would like to develop a visualization system to take the adversarial nature of sports. For instance, although two boxers might have roughly the same skill level, in a match, one's skill composition might give him/her advantages over the other. This kind of analysis would be very useful in preparation for a game or predicting the result.

Acknowledgement

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1), Hong Kong Baptist University Science Faculty Research Grants (Ref: FRG2/14-15/105), NSFC Young Scientist Research Grant (Project no.: 61302176) and the Hong Kong Research Grant Council (Project no.: GRF210813).

References

ARIKAN, O., AND FORSYTH, D. 2002. Motion generation from examples. *ACM Transactions on Graphics 21*, 3, 483–490.

ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2003. Motion synthesis from annotations. *ACM Trans. Graph. 22*, 3 (July), 402–408.

BEAUDOIN, P., COROS, S., DE PANNE, M. V., AND POULIN, P. 2008. Motion-motif graphs.

BIDEAU, B., KULPA, R., MENARDAIS, S., FRADET, L., MULTON, F., DELAMARCHE, P., AND ARNALDI, B. 2003. Real handball goalkeeper vs. virtual handball thrower. *Presence: Teleoper. Virtual Environ. 12*, 4, 411–421.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIC', Z. 2004. Style-based inverse kinematics. *ACM Transactions on Graphics (TOG) 22*, 3 (August).

HECK, R., AND GLEICHER, M. 2007. Parametric motion graphs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '07, 129–136.

HO, E. S., AND KOMURA, T. 2009. Character Motion Synthesis by Topology Coordinates. *Computer Graphics Forum*.

HO, E. S. L., AND KOMURA, T. 2011. A finite state machine based on topology coordinates for wrestling games. *Computer Animation and Virtual Worlds 22*, 5, 435–443.

HO, E. S. L., SHUM, H. P. H., CHEUNG, Y.-M., AND YUEN, P. C. 2013. Topology aware data-driven inverse kinematics. *Comp. Graph. Forum 32*, 7 (Oct), 61–70.

HOLDEN, D., SAITO, J., AND KOMURA, T. 2016. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph. 35*, 4 (July), 138:1–138:11.

HYUN, K., LEE, K., AND LEE, J. 2016. Motion grammars for character animation. *Computer Graphics Forum 35*, 2, 103–113.

KOMURA, T., KURODA, A., AND SHINAGAWA, Y. 2002. Nicemeetvr: Facing professional baseball pitchers in the virtual batting cage. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, ACM, New York, NY, USA, SAC '02, 1060–1065.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics 21*, 3, 473–482.

KWON, T., AND SHIN, S. Y. 2005. Motion modeling for on-line locomotion synthesis. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 29–38.

LAU, M., AND KUFFNER, J. J. 2005. Behavior planning for character animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 271–280.

LAWRENCE, N. D. 2004. Gaussian process latent variable models for visualisation of high dimensional data.

LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics 21*, 3, 491–500.

LI, Y., WANG, T., AND SHUM, H.-Y. 2002. Motion texture: A two-level statistical model for character motion synthesis. *ACM Trans. Graph. 21*, 3 (July), 465–472.

LIU, Z., ZHOU, L., LEUNG, H., AND SHUM, H. P. H. 2016. Kinect posture reconstruction based on a local mixture of gaussian process models. *IEEE Transactions on Visualization and Computer Graphics*.

MIN, J., AND CHAI, J. 2012. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics 31*, 6, 153.

MOLET, T., AUBEL, A., NOSER, H., CAPIN, T., LEE, E., PANDZIC, I., THALMANN, D., AND THALMANN, N. 1999. Anyone for tennis.

PLANTARD, P., SHUM, H. P. H., AND MULTON, F. 2016. Ergonomics measurements using kinect with a pose correction framework. In *Proceedings of the 2016 International Digital Human Modeling Symposium*, DHM '16.

PLANTARD, P., SHUM, H. P. H., AND MULTON, F. 2016. Filtered pose graph for efficient kinect pose reconstruction. *Journal of Multimedia Tools and Applications*.

REITSMA, P. S. A., AND POLLARD, N. S. 2007. Evaluating motion graphs for character animation. *ACM Trans. Graph. 26*, 4 (Oct.).

SAFONOVA, A., AND HODGINS, J. K. 2007. Construction and optimal search of interpolated motion graphs. *ACM Trans. Graph. 26*, 3 (July).

SHIN, H. J., AND LEE, J. 2006. Motion synthesis and editing in low-dimensional spaces. *Computer Animation and Virtual Worlds (Special Issue: CASA 2006) 17*, 3-4, 219 – 227.

SHIN, H. J., AND OH, H. S. 2006. Fat graphs: Constructing an interactive character with continuous controls. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 291–298.

SHUM, H. P. H., KOMURA, T., SHIRAISHI, M., AND YAMAZAKI, S. 2008. Interaction patches for multi-character animation. *ACM Trans. Graph. 27*, 5 (Dec), 114:1–114:8.

SHUM, H. P. H., KOMURA, T., SHIRATORI, T., AND TAKAGI, S. 2010. Physically-based character control in low dimensional space. In *Proceedings of the Third International Conference on Motion in Games*, Springer-Verlag, Berlin, Heidelberg, vol. 6459 of *MIG '10*, 23–34.

SHUM, H. P. H., KOMURA, T., AND YAMAZAKI, S. 2012. Simulating multiple character interactions with collaborative and adversarial goals. *IEEE Transactions on Visualization and Computer Graphics 18*, 5 (May), 741–752.

SHUM, H. P. H., HO, E. S. L., JIANG, Y., AND TAKAGI, S. 2013. Real-time posture reconstruction for microsoft kinect. *IEEE Transactions on Cybernetics 43*, 5, 1357–1369.

TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500, 2319–2323.

WANG, J., HERTZMANN, A., AND BLEI, D. M. 2006. Gaussian process dynamical models.

XIA, S., WANG, C., CHAI, J., AND HODGINS, J. 2015. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Trans. Graph. 34*, 4 (July), 119:1–119:10.

YEADON, M. The biomechanics of the human in flight. *The American Journal of Sports Medicine 25*, 4, 575–580.

YEADON, M. 1990. The simulation of aerial movement- iv. a computer simulation model. *Journal of Biomechanics 23*, 1, 85–89.