Computers & Graphics 69 (2017) 104–115

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on Motion in Games 2016

Posture-based and action-based graphs for boxing skill visualization

Yijun Shen[a], He Wang[b], Edmond S.L. Ho[a], Longzhi Yang[a], Hubert P.H. Shum[a,*]

[a] Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, United Kingdom
[b] School of Computing, Faculty of Engineering, University of Leeds, Leeds, United Kingdom

ARTICLE INFO

Article history:
Received 29 July 2017
Revised 25 September 2017
Accepted 26 September 2017
Available online 14 October 2017

Keywords:
Motion graph
Hidden Markov model
Information visualization
Dimensionality reduction
Human motion analysis
Boxing

ABSTRACT

Automatic evaluation of sports skills has been an active research area. However, most of the existing research focuses on low-level features such as movement speed and strength. In this work, we propose a framework for automatic motion analysis and visualization, which allows us to evaluate high-level skills such as the richness of actions, the flexibility of transitions and the unpredictability of action patterns. The core of our framework is the construction and visualization of the posture-based graph that focuses on the standard postures for launching and ending actions, as well as the action-based graph that focuses on the preference of actions and their transition probability. We further propose two numerical indices, the Connectivity Index and the Action Strategy Index, to assess skill level according to the graph. We demonstrate our framework with motions captured from different boxers. Experimental results demonstrate that our system can effectively visualize the strengths and weaknesses of the boxers.

1. Introduction

Computer technologies have taken on a crucial role in modern sports and health sciences, in revolutionizing the way to observe, analyze, and improve the performance of both amateur and professional athletes. Computer-managed weight lifting machines, treadmills and many other training equipment provide energy consumption or repetition and weight management in many sport clubs. Virtual reality technology has been applied in various training systems in baseball [1], handball [2] and tennis [3] to assist more professional sport activities. Nevertheless, these technologies are only able to analyze motions at a low level, i.e. recording the timing or repetitions of basic motions and comparing movement trajectories with those performed by better players. More advanced technologies are needed for personalized and higher-level analysis comparable to that from human experts.

In addition to the instantaneous movement features of the sports players, Experienced sport coaches consider high-level features such as the variety of actions and quality of transitions from one action to another. Taking boxing as an example, professional boxers have in basic actions such as defence, stepping and attack, threading through which the transitions are carried out based on the strategy and the opponent's reactions. The action transitions of a good boxer need to be flexible and contain great variety to achieve the optimal outcome. Such information often serves as an important indicator in assessing the skill level of a player, and the same principle applies to many other sports such as basketball [4] and fencing [5]. Unfortunately, automatic systems for analyzing and evaluating sports motions at such a high level is very limited.

In this paper, we propose a robust visualization system to address the above limitations, by represent motions as an interactive graph of high-level features, including the flexibility and richness of the actions as well as the transitions of actions. Although we use boxing as a demonstration in this paper, our method is generic and can be applied to different sports. Our approach starts with capturing the *shadow boxing* training motion of a boxer, in which the boxer performs boxing with an imaginary opponent. An experienced coach can effectively assess the boxer's skill by watching the shadowing boxing motions. As a positive side effect, this method of motion analysis greatly reduces the complexity of motion capture due to occlusion and collision and has shown to be very effective in our system. The motion data is then processed and visualized in two different graphs: the posture-based graph and the action-based graph, for performance analysis.

In the posture-based graph, the semantic actions segmented from the captured motion are grouped into clusters based on a customized distance function that considers action specific features. Our system then automatically generates a motion graph structure known as *Fat Graph* [6], which uses nodes to represent

* Corresponding author.
E-mail addresses: yi.shen@northumbria.ac.uk (Y. Shen), h.e.wang@leeds.ac.uk (H. Wang), e.ho@northumbria.ac.uk (E.S.L. Ho), longzhi.yang@northumbria.ac.uk (L. Yang), hubert.shum@northumbria.ac.uk (H.P.H. Shum).

https://doi.org/10.1016/j.cag.2017.09.007

groups of similar postures to start and end actions, and edges to represent groups of action. By applying dimensional reduction techniques, this graph can be visualized in a 3D space for performance analysis and evaluation. The transition capability of the boxer are visualized by the connectivity of the nodes, where the richness and preference of the actions are visualized by the edges in the graph. We further propose a skill evaluation metric known as the *Connectivity Index* which evaluates the richness of actions and the flexibility of transitions according to the graph.

Whilst the posture-based graph focuses on the variety of basic postures and the transition flexibility between actions, the action-based graph mainly considers the richness of actions and the transition probability among them. The action-based graph is constructed as a customized Hidden Markov Model (HMM) [7], in which similar actions are grouped into clusters that formulate the nodes. The transition probability among actions is calculated and is expressed as edges between nodes. The graph is visualized in a 3D space, and the positions of the nodes and edges are optimized for better visualization. With such a graph, the pattern of action launching can be easily identified in order to assess the boxing strategy of the boxer. We further propose the *Action Strategy Index* to evaluate the unpredictability of action patterns according to the graph.

We conducted experiments on the motions captured from multiple boxers and evaluate their skills. The corresponding posture-based and action-based graphs were generated. As shown in Fig. 10, we can easily evaluate the skills of different boxers with our visualization system.

There are three main contributions of this work:

- We propose a framework for high-level skill analysis through automatic motion analysis and visualization. Given a captured motion from a sports player, our system automatically segments the motion into semantic action units and constructs two graph structures.
- We propose the posture-based graph, which is a variant of the Fat Graph, to visualize the skills according to different standard postures for launching and ending actions. It allows the user to identify the correctness of standard postures and the diversity of actions. We further propose the Connectivity Index that evaluates the richness of actions and the flexibility of transitions.
- We propose the action-based graph, which is a variant of the Hidden Markov Model (HMM), to visualize the skill according to different groups of action. It allows the user to identify the preference of actions and their transition probability. We further propose the Action Strategy Index to evaluate the unpredictability of action patterns.

The preliminary results of this work were published in a conference paper [8], which proposed only the posture-based graph. In this paper, we extend the work by introducing the new action-based graph. We perform analysis and experimental evaluation of such a graph, and compare its performance with the posture-based graph. We have also updated the paper thoroughly such that the two graphs are presented in an organized and effective manner.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The details of motion capture and organization are given in Section 3. In Sections 4 and 5, we explain the design and implementation of the posture-based graph and the action-based graph respectively. Related experiments can be found in Section 6. The paper is concluded in Section 7 with future research directions discussed.

2. Related work

2.1. Sports visualization

Helping athletes on skill improving via the visualization of sport motions is a field that has not been fully explored in the field of sports science. Existing research [9,10] mainly focuses on the appearance changes of motions when body and motion parameters are changed. For example, Yeadon [9,10] has done research on how diving and somersault motions change when the motions are launched at different timings by using physical simulation. Although such tools are useful for the athletes to interactively visualize possible results under different parameters, they can only evaluate the performance of 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 athlete in previous research and it is the major difference between our work and the afore-mentioned ones. In this research, we combine the approaches of motion graph [11–13] and dimensionality reduction [14,15] to visualize high-level skills information of the athletes for the skill assessments.

2.2. Motion graphs for motion modeling

The Motion Graph approach [11–13,16–19] is a method to interactively reproduce continuous motions based on a graph generated from captured motion data. Reitsma and Pollard [20] compared different motion graph techniques comprehensively. Heck et al. [21] further parametrized the motion space to control how the motions are generated by blending samples in the motion graph. Such an approach can be used for interactive character control such as that in computer games. When it comes to graph construction, [16,17] are the ones most similar to our method. Min et al. [16] grouped similar postures and transitions into nodes and edges. Their focus was the motion variety of synthesized motions so they used generative models to fit the posture and motion data. Our focus is on skill visualization through the analysis of postures and motions so we can afford simpler and faster methods of analysis. Beaudoin et al. [17] cluster postures first then find motion motifs by converting the motion matching task into a string matching problem. Their priority was to find motifs that were representative while our focus is to visualize motion details and statistics to help people assess the skills. Xia et al. [22] 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 [23] optimized the graph structure by combining motion graph and interpolation techniques to improve performance. On the other hand, works to resolve this problem by introducing a hierarchical structure were proposed [6]. 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 [24,25]. A recent work by Hyun et al. [4] 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

[26] built a finite state machine (FSM) based on Topology Coordinates [27] 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 [6] 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 the temporal relationship between poses are considered [28]. Such a structure, however, is targeted for motion reconstruction and analysis rather than visualization [29].

2.3. Statistical motion modeling

Dimensionality reduction methods have been proposed to visualize the overall structure of captured motions. Grochow et al [14] proposed a method to project the 3D motions of a human onto a 2D plane, and further reconstruct 3D motions by mapping arbitrary points from the 2D plane back onto 3D joint space. PCA [15] and ISOMAP [30] 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 [31,32]. One can generate motions from arbitrary points on the plane by interpolating the postures of the original motion. Meanwhile, non-linear methods [33,34] and Deep Learning [35] have also been used to reduce the dimensionality of motions. The Gaussian Process [36] and the mixture of Gaussian Processes [36] can be used to represent a set of human postures with a small number of Gaussian parameters. However, such methodologies do 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.

Other researchers have focused on the connectivities of motion/actions by methods such as Markov models. Hidden Markov Model (HMM) [7] has been widely used in analyzing and synthesizing human motion. Typically, the hidden states of the HMM are the distribution of body poses and the dynamics of the motions are represented by the transitions between the hidden states. The parameters of the HMM can then be learned from training data using the Expectation-Maximization (EM) algorithm. Hara et al. [37] proposed to model daily activities using HMM in intelligent house. Françoise et al. [38] proposed to use HMM models for analyzing Tai Chi motion sequences. An early work proposed by Brand and Hertzmann [39] proposed to learn the dynamics of human motion using HMM in their motion style synthesis model. Tango and Hilton [40] proposed to learn a HMM model from captured human motion for synthesizing in-between frames in keyframe animation. Ren et al. [41] presented a data-driven approach for quantifying naturalness of human motion including those synthesized by HMM. While existing work focuses on finding statistical distributions of motions, our focus is on visualizing the motion richness and the transition dynamics for skill assessments.

3. Motion capture and organization

We first capture the motion required for analysis using motion capture systems. Then, we propose an automatic system to segment long sequences of captured motion into meaningful actions, which are used as building blocks of our posture-based and action-based graphs.

Fig. 1. The shadow boxing motions of several boxers were captured using an optical motion capture system.

Here, we follow the definition from [25], in which a *motion* is considered to be a raw sequence of captured human movement, and an *action* is considered to be a short, meaningful segment of movement within a motion. 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.

Postures and actions are good entities for skill visualization, as sports players typically plan their strategies and evaluate their performances with such terms. For example, a boxer typically thinks about what sort of attack/defense/transition should be launched during a match. A coach typically evaluates the overall strategy in the action level, as well as how well individual postures and actions are performed.

3.1. Motion capture

Although it would be best to capture the motions of all players in multi-player sports because the data would reflect the features of the motions, capturing multiple players remains difficult due to the occlusions and collisions among players. Fortunately, it is possible to only capture individual motions for our purposes without compromising the true motion characteristics. In boxing or any other martial arts, there is a training practice called “shadow boxing”. The boxer imagines a boxing session with another boxer, and launches boxing actions to interact with such an imaginary opponent. The boxer launches not only offensive actions such as punching, but also defence, stepping, and the consecutive combination of all such actions. There are similar practice methods in basketball and soccer as well, in which players use the ball to conduct various techniques in the court, imagining that their opponents are trying to take the ball away from them. The players thus perform various actions to keep the ball and trick an imaginary opponent. This technique has also been used by coaches for skill assessment hence is suitable for our analysis. We employed an optical motion capture system to acquire the performed motion as shown in Fig. 1 as it was less intrusive and highly accurate. Also, we preferred to capture long and continuous clips of motion, such that the player could perform the motion in a natural manner.

3.2. Motion analysis

After data capture, the system automatically segments meaningful actions from the raw captured motion, and identifies the ef-

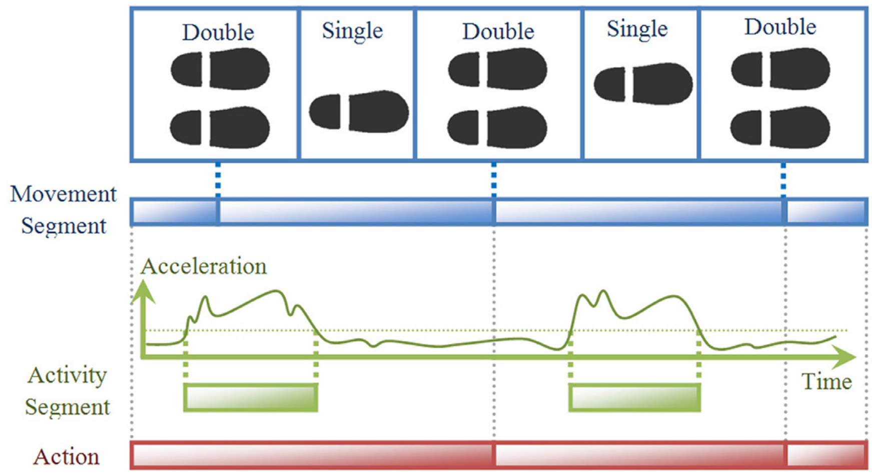

Fig. 2. Upper: The movement segment is defined as the period between two double support supporting phases. Middle: The activity segment is defined as the period with high acceleration. Lower: The action is the combination of movement segment and activity segment.

fective joints that contribute the most to the semantic meaning of the actions.

For boxing motions, we observed that actions normally start and end in a double supporting state (i.e. both feet touching the floor), as the state is usually dynamically stable. We detect such a state by monitoring the feet height and velocity and setting corresponding thresholds. This allows us to segment the raw captured motion into a set of *movement segments*, which are the periods between every two successive double supporting states, as visualized in Fig. 2 Upper.

We also observed that actions normally require a relatively larger force to be performed, such as a punch or a step. We define periods with a high-level of force exertion as the *activity segments*. Since force is proportional to acceleration, these segments can be found when the sum of squares of acceleration of all joints is above a threshold, as visualized in Fig. 2 Middle. The threshold is statistically obtained from the acceleration profile of the motion.

Finally, 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 action of pure body transition. (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 one or more 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 as visualized in Fig. 2 Lower. Note that due to this merging process, the resulting action may contain multiple activity segments. In our system, we implement an optional step to filter very short actions that are likely to be generated due to the noise of the supporting feet.

We define the *effective joints* to be the set of joints to represent an activity segment. In case (1) above, since the actions contain no special activities, the pelvis is considered to be the effective joint. 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 as there are multiple activity segments. Such joints are used in later processes to evaluate the similarity of actions.

4. Posture-based graph

The posture-based graph focuses on evaluating the common postures that are used to start and end actions. In such a graph, the nodes represent similar postures and the edges represent similar actions. It allows us to evaluate the consistency of common postures and the diversity of actions.

4.1. Graph construction

We adopt a Fat Graph structure [6] in the action level [25] to generate the posture-based graph, as it can effectively simplify the graph representation by grouping similar postures and actions together. The Fat Graph was originally proposed for motion synthesis, and thus it is not optimized for skill visualization. We redesign the algorithms to generate nodes and edges in the Fat Graph for our purpose.

4.1.1. Fat Nodes

In our system, the nodes of the Fat Graph, known as Fat Nodes, are the common starting or ending postures of the actions. We design an unsupervised clustering scheme for grouping all starting/ending postures into a finite set of posture groups, which avoids additional labour for posture labelling and grouping. Specifically, we used k-means to cluster postures. The distance between two postures P_0 and P_1 is defined as:

$$D(P_0, P_1) = \sum_{i=0}^{i=i_{total}} |\theta_0(i) - \theta_1(i)| \tag{1}$$

where $\theta_0(i)$ and $\theta_1(i)$ represent the 3D joint angle of the joint i in posture P_0 and P_1 respectively, and i_{total} is the total number of joints. 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. 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 value of k that does not violate the distance threshold. After clustering, we use the mean posture of a group to represent the corresponding Fat Node. The nodes in the graph represent the set of standard postures which the player starts various action from. In the case of boxing, they are usually the fighting postures that the boxer uses to guard his/her face against the opponent, with both feet landing on the ground and keeping shoulder width apart.

By evaluating the Fat Nodes alone, we can already tell if a boxer has multiple unnecessary standard postures, or if any standard postures contain potential weakness. In general, experience players have fewer Fat Nodes, such that they can start actions in a standard posture effectively without the needs of shifting to other ones. Novice players sometimes may have a particular Fat Node for some particular actions. This is discouraged in boxing training as such postures hint the opponent as to what actions are going to be launched.

4.1.2. Fat Edges

We design the edges of a Fat Graph, known as Fat Edges, as directional edges that represent groups of similar actions. Each edge points from the Fat Node representing the starting posture to that representing the ending posture.

Similar to the Fat Nodes, we implement an unsupervised clustering algorithm to group similar actions into Fat Edges. We use k-means to cluster the actions and search for the smallest acceptable k for a given distance threshold. We define the actions distance according to the trajectory of the effective joints as explained in Section 3.2. 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 actions A_0 and A_1 is defined as

$$D(A_0, A_1) = \begin{cases} \infty \\ \text{if } A_0 \text{ and } A_1 \text{have different sequences} \\ \text{of effective joints} \\ \\ \sum_{j=0}^{j_{total}} \sum_{f=f_{start}}^{f_{end}} [A_0(j)(f) - A_1(j)(f)] \\ \text{otherwise} \end{cases} \qquad (2)$$

where $A_0(j)(f)$ and $A_1(j)(f)$ represent the 3D positions of effective joint j in frame f in the action $A-0$ and A_1 respectively, j_{total} is the total number of effective joints in the actions, f_{start} and f_{end} are the starting frame and ending frame of the considering effective joint. 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.

In the field of boxing, a Fat Edge typically contains 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 have to share the same starting and ending Fat Nodes, if an action group contains multiple starting or ending poses, it is sub-divided into multiple Fat Edges.

Again, by only looking at Fat Edges, one can tell the differences between experienced and novice players. Experienced players normally have Fat Edges with similar numbers of actions, as they have mastered a large variety of boxing actions and can switch between them effectively using a small number of stable transition maneuvers. Novice boxers tend to have a larger number of Fat Edges but each with a small number of actions, due to the inability to reproduce boxing actions consistently. Figure 3 shows the relationship of Fat Nodes and Fat Edges.

4.2. The connectivity index

It requires deep knowledge and years of experience to assess one’s skills in sports. Here, we make use of the posture-based graph and define an index representing the skill level, allowing more objective and efficient skill assessment.

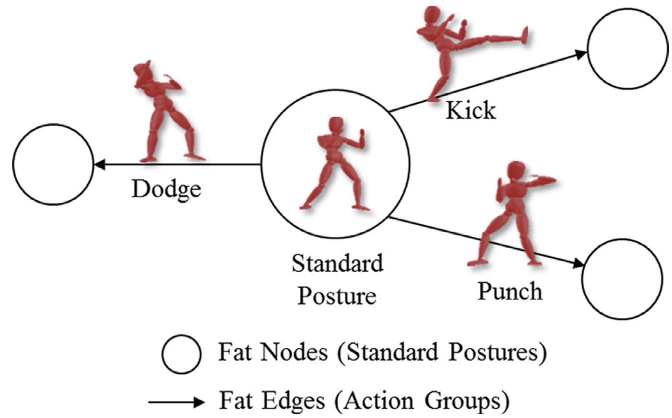

Fig. 3. The Fat Node represents the standard fighting pose. The three outgoing Fat Edges represent different action groups.

In many types of sports, there are two important skill indicators. The first one is the richness of the actions that indicates the resourcefulness of a player. The other is the flexibility of transitions between states so that the player can switch between different states at will. Our posture-based graph captures both of the indicators. The richness can be represented by the number of Fat Edges, indicating how many kinds of maneuvers the player has. The flexibility is indicated by the connectivity of the graph, which is inversely proportional to the number of Fat Nodes. A fully connected graph shows great flexibility because there are transitions between any two nodes.

Notice that 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. We therefore define a *Connectivity Index* that evaluates both the action richness and the action flexibility of a player

$$CI = \frac{\text{Number of Fat Edges}}{\text{Number of Fat Nodes}} \qquad (3)$$

To accurately reflect the skill level of a player, in our implementation, we do not consider 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 only consider Fat Edges that are consistently performed, as those having only a small number of member actions could be randomly performed actions. Empirically, we consider edges having more than 2 member actions.

4.3. Visualization system

Here, we describe the design of our visualization system to visualize the posture-based graph in an effective manner. We also introduce interactive features for the user to view the graph with different levels of details.

The posture-based graph consists of high dimensional Fat Nodes (groups of similar postures of many degrees of freedom) and Fat Edges (groups of similar actions in the spatial-temporal domain), which presents a challenge for visualization. To reduce the dimensionality for better visualization, we propose two different schemes for nodes and edges due to their different nature in this graph. Specifically, we project the Fat Nodes on a 2D space using Principal Component Analysis (PCA) as it creates a more consistent low dimensional space compared with other methods. We represent Fat Edges with 2D curves and augment the curves with a combination of geometric primitives to visualize the action features.

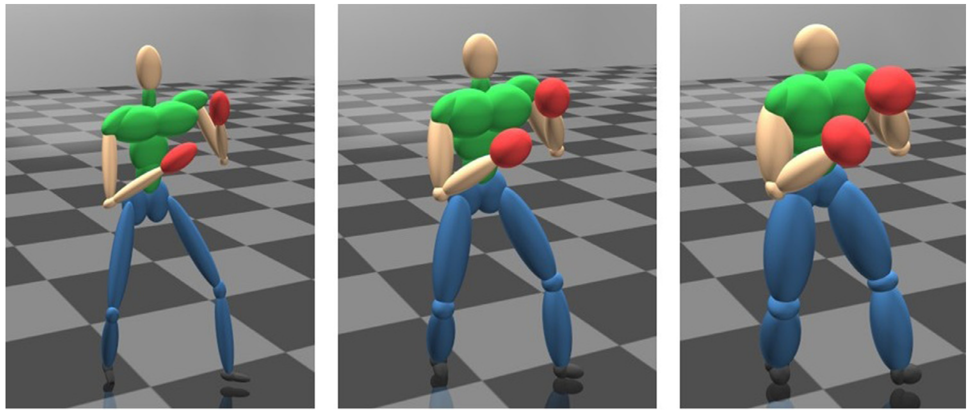

Fig. 4. From left to right, the character becomes larger as the size of the nodes increases.

4.3.1. Visualizing Fat Nodes

Although the degree of freedom (DOF) of human postures are in high dimensionality (45 DOF in our system), they are intrinsically dependent on each other [14]. In fact, the Fat Nodes can be represented effectively in a 2D space where nodes of similar postures are located together while those of different postures are located far apart. This allows viewers to easily understand the relationship between postures.

For each Fat Node, we obtain the mean posture as its representation. Given a set of postures, we apply principal component analysis (PCA) to reduce the dimensionality to 2. Essentially, we calculate the covariance matrix to evaluate the intrinsic dependency of the dimensions. We then calculate the eigenvectors from such a covariance matrix, and use the two eigenvectors with largest eigenvalues to form a feature vector.

PCA is used as it has shown to be effective on human postures [14]. However, since we only have a small number of postures, we believe other dimensionality reduction techniques would also work.

We render the mean posture of each Fat Node onto a 2D X-Z plan. This allows the user to identify inappropriately performed postures. In boxing, novice boxers sometimes lose track of their boxing rhythm, and hence start or end an action with an inappropriate posture. We use the fatness of the character to represent the number of member postures in the node, as shown in Fig. 4. This allows the user to easily observe the postures that the player usually uses to start actions.

4.3.2. Visualizing Fat Edges

Here, we explain how to visualize the Fat Edges, which contain information of groups of similar actions.

We do not apply dimensionality reduction techniques directly on the action data itself because the low dimensional projection would be very complex. Instead, we propose to visualize each Fat Edge by a 2D curve that represents its mean action on the *X–Z* plane. We optimize the angle and sign of these curves to minimize occlusion. For edges with a starting node different from the ending node, the edge angle is fixed. The only adjustable variable is the bending side of the curves, which is essentially the sign of the curves. For those with the same starting and ending node, both edge angle and bending side can be controlled. We optimize the signs and angles of the edges in a greedy manner such that they would blend towards a less dense region of the graph.

To visually distinguish between different Fat Edges, we add some geometric patterns to the 2D curves. We collect the high-energy frames of all actions and project them onto a 1D space using the PCA system explained in Section 4.3.1. Since the high-energy frames of different actions are typically distinguishing postures, the projection essentially maps all action features onto a normalized 1D space in the range of $[-1.0, 1.0]$. To visualize the value in this 1D space, we design some geometric patterns for landmark values -1.0, -0.5, 0.0, 0.5 and 1.0 as shown in Fig. 5 Upper. The patterns to represent values between two landmarks are obtained by linear interpolation between nearby landmarks.

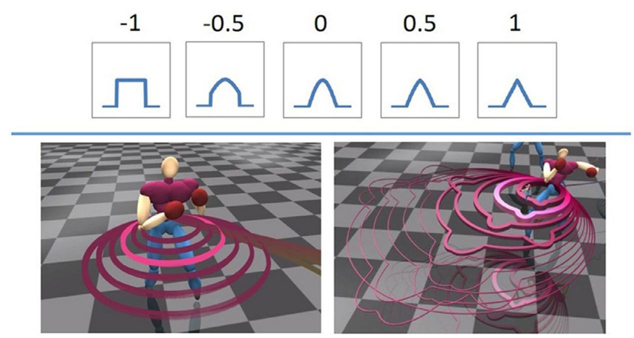

Fig. 5. The geometric patterns for landmark values between -1 and 1. Each pattern represents a landmark posture in an action. (Lower) Comparison of visualization without/with the patterns. Each curve represents a group of action. The right image shows the uses of landmark patterns to identify different types of action.

We further represent the number of member actions in the edge by the thickness of the curve. This allows the user to identify the player's preferred actions. For instance, if a boxer relies heavily on single straight punches, the Fat Edge for such action will be unreasonably thick, while edges for other attacks will be relatively thin, which demonstrates a potential lack of diversity attacking strategies.

Through the comparison between Fig. 5 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.

4.3.3. Interactive features

We integrate some interactive features in our system to display relevant information based on user input. When the user selects any specific entities in the graph, related information will be shown.

When a Fat Node is selected, its corresponding Fat Edges will be highlighted for easier observation. Information about the number of members in that node, number of outgoing edges, and number of incoming edges are displayed in a sub window. When a Fat Edge is selected or highlighted (because of a Fat node selection),

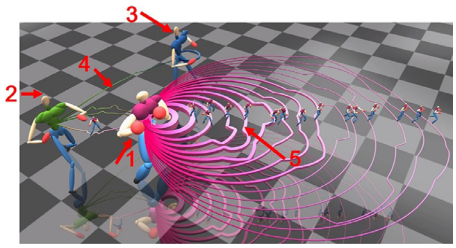

Fig. 6. The posture-based graph of the Boxer S. 1, 2 and 3 are Fat Nodes. 4 and 5 are two Fat Edges. 4 connects Node 2 and Node 3. 5 connects Node1 to itself.

we render the member actions included, such that the user can understand the content of the edge.

As an example, in Fig. 6, 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 fatness. Node 1 is represented by the most muscular character, which indicates the largest node size. Nodes 2 and 3 are far thinner. Fat Edges are rendered as curves between nodes such as the ones shown by 4 and 5. The thicknesses of the edges indicate the frequency of the actions taken. Edge 5 is thicker than edge 4, suggesting that this boxer takes action 5 more often. In addition, an edge can be smooth like a circle or bumpy with geometric patterns. A single pattern means one activity segment such as a single punch, while multiple patterns indicate a series of activities such as a combo attack. Our system also supports interactive features. Fig. 6 is the result when the user selects Node 1. All the edges starting from this node are highlighted, each with a small character performing the action on it.

5. Action-based graph

The action-based graph focuses on evaluating the transition probability from one action class to another. In such a graph, the nodes represent groups of action with similar activity segments. The edges represent the transition probability between two action groups. It allows us to evaluate the pattern of launching actions and extract the strategy of the boxer.

5.1. Graph construction

We use the hidden Markov model (HMM) to organize the captured motion, as it has been shown effective in modelling human motion. In the domain of character animation, HMM has been mostly used in the posture level to create motion graphs [12]. We adapt the graph into the action level such that we can visualize the transition probability among actions.

The nodes of the graph represent different action groups. We apply Eq. (2) to group the captured actions into a number of action groups with k-means clustering. The process is similar to that in Section 4.1, in which we define a threshold based on expert knowledge, and then incrementally increase the number of classes until the threshold is met. We denote k' as the total number of groups, $|G_i|$ as the number of actions in the ith action group (which is used in the visualization system for visualizing the fatness and the placement of the node and will be described later).

The edges of the graph represent transitional probability from one action group to another. To obtain the transitional probability, we go through the sequence of actions in the captured motion and count the number of occurrences for an action belonging to group i to be followed by another belonging to group j, which is denoted as c_{ij}. The transition probability of action group i to action group j

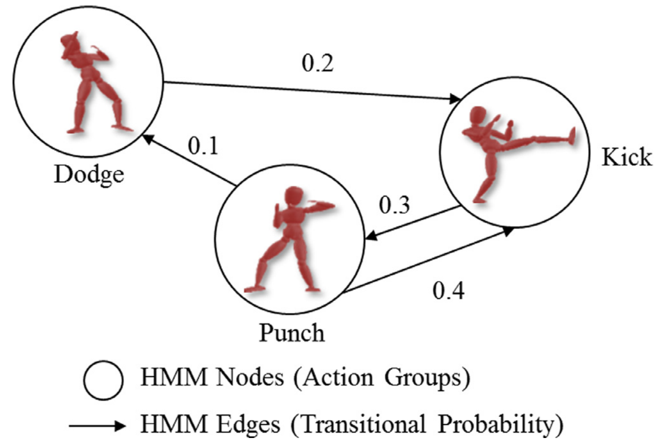

Fig. 7. The three HMM nodes represent action groups. The HMM edges represent transitional probability between them.

is defined as

$$T_{ij} = \frac{c_{ij}}{\sum_{m=1}^{k'} \sum_{n=1}^{k'} c_{mn}} \tag{4}$$

where the denominator represents the total number of transition in the whole motion. Notice that i may be equal to j. In such a case, two actions of the same action group are launched successively.

The concept of the action-based graph is shown in Fig. 7. In general, experienced boxers tend to have a more evenly distributed transitional probability across all actions, which means that there should be edges connecting all the nodes. This indicates that the boxer's pattern is dynamic and cannot be easily predicted by an opponent. Conversely, novice boxers may have limited edges and some thick edges connecting two nodes, which means a high probability to launch those two groups action consecutively. An opponent may discover such a pattern and counter-act in advance when the first action is observed.

5.2. The action strategy index

In many sports, the unpredictability of action patterns is an important skill indicator. Experienced players would diversify their action patterns such that their opponents cannot predict the next action. However, novice players tend to perform actions based on predictable patterns (i.e. the sequence of actions to be launched continuously), which can be easily identified. For example, a novice boxer usually perform two straight punches successively. This is because the boxer is not able to link different types of punches fluently, and therefore would perform the simplest punches again and again. The proposed action-level graph allows easy observation of boxing patterns, as we can visualize the transitional probability among actions. We further propose the *Action Strategy Index*, which evaluates the unpredictability of action pattern. We obtain the number of outgoing HMM edges for each HMM node, forming a set that is denoted as $e = \{e_i\} \; \forall \, i \in [1, k']$, where k' is the total number of HMM nodes. Skillful players would have similar values in the e set, while novice players would have very different values. We therefore define the Action Strategy Index as the precision of e, that is, the reciprocal of its standard deviation

$$ASI = \frac{1}{\sigma(e)} \tag{5}$$

where σ represents the standard deviation operator. A high *ASI* value indicate that the player's action patterns are more unpredictable, which indicates a higher skill level.

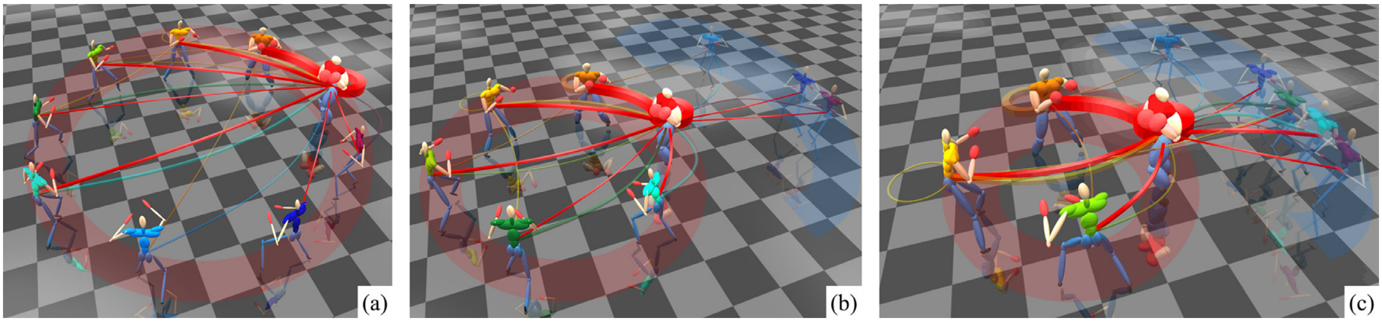

Fig. 8. Action-based graphs of the same boxer generated by setting the frequency threshold as (a) 0, (b) 1 and (c) 2. The red shade indicates the inner circle covering nodes of the frequent class, and the blue shade indicate the outer circle covering nodes of the rare class.

5.3. Visualization system

Here, we explain the visualization system for the action-level graph. The system allows easy observation of the preference of action and the boxing pattern. Both are very important aspects to evaluate the high-level strategy of a boxer.

5.3.1. Visualizing HMM Nodes

Each action group is represented by its corresponding median action, which is the action that is the closest to the mean value of the action group during k-means clustering. We render the nodes using human characters with the starting posture of the median action. The number of actions in each action group is visualized using the fatness of the corresponding character. The color of the nodes are randomized.

As mentioned in Section 4.1.2, we observe that some boxers, especially novices, may produce random actions that are not repeatable. Such actions may generate a large number of thin nodes, which distract the user from evaluating the actions that are often launched. Therefore, we classify the action groups with $|G_i| > a$ into the *frequent class*, and groups with $|G_i| \leq a$ into the *rare class*, where G_i is the number of member actions in a node as defined in Section 5.1, a is a preset frequency threshold. Fig. 8 shows the result of setting different values of a. We find that setting $a = 2$ generates the best results.

We place the nodes belonging to the frequent class at an inner circle, and those belonging to the rare class at an outer circle, such that the user can identify them easily and decide what to focus on. For the inner circle, nodes are ordered according to the corresponding value of $|G_i|$, and are placed evenly at a circle with a smaller radius. For the outer circle, to minimize edge crossing, we place the nodes at a position on a circle with a larger radius that is the closest to the nodes with incoming and outgoing edges. To implement this, we develop a simple optimization algorithm that optimizes the position of the nodes. During the optimization, we constrain the position to be at the circle and not overlapping with existing nodes. We then minimize the sum of distance with respect to the nodes connecting to the current one.

By default, we render the HMM node belonging to the frequent class with solid colors, and those belonging to the rare class in semi-transparent colors. This further avoids the user being distracted by the rarely performed actions.

5.3.2. Visualizing HMM Edges

We visualize the edges using 2D curves. While we can render the edges with straight lines, the resultant group would be difficult to observe as the lines overlap significantly. We augmented the edges with a small random curvature to solve the problem. We also render the edges as semi-transparent such that the users can see through partially overlapped edges. The thickness of the edge is proportional to T_{ij} calculated in Eq. (4). As a result, a thicker edge connecting node i to node j indicates that the boxer often launches action group j after action group i. The color of the edges are decided based on that of the source node. This helps the user to identify which action groups the boxer may launch after a particular one.

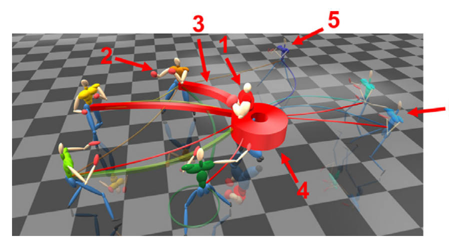

Fig. 9. The action-based graph of the Boxer S. 1, 2 are HMM nodes belonging to the frequent class. 3, 4 are outgoing HMM edges from the node 1. 5, 6 are HMM nodes belonging to the rare class.

5.3.3. Interactive features

We also implement some interactive features such that the user can select what to view. The most important component of the action-based graph is the action itself. Therefore, we implement an interactive system such that when a user clicks on a particular HMM node, the median action of the corresponding action group is displayed. We also highlight the outgoing edges from such a node. This allows the user to examine individual action group together with the transition probability to the next groups. The information of the node, such as the number of member actions and the number of out-going HMM edges, are displayed on a separate window.

As an example, in Fig. 9, there are 5 HMM nodes belonging to the frequent class including node 1 and 2. These nodes are visualized with more muscular characters, meaning that the boxer performs them more frequently. There are 3 HMM nodes belonging to the rare class including node 5 and 6, which are visualized with thinner characters. Node 1 has 5 outgoing HMM edges, in which edge 3 point towards another node, while edge 4 is a self-connecting edge. Edge 4 is thicker than the others, indicating that the boxer performs successive actions belonging to node 1 very frequently. The screen is captured when the user selects node 1, and as a result, all outgoing edges of node 1 are highlighted, and the character representing node 1 performs the corresponding median action.

6. Experimental results

In this section, we present experimental results. We captured the motions of four boxers with varying skill levels. We first give

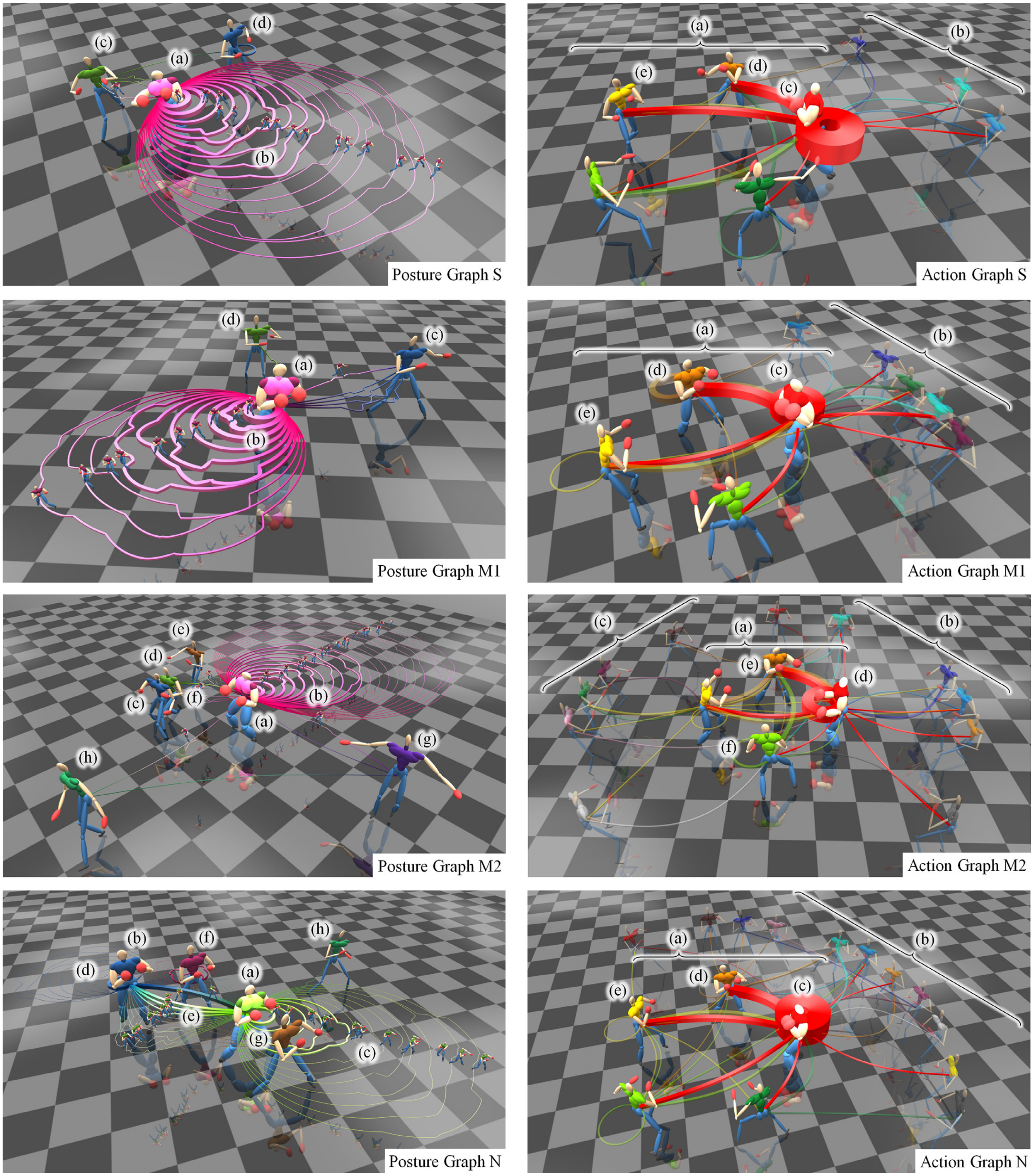

Fig. 10. (Left) The posture-based graphs and (Right) the action-based graphs for boxer N, M1, M2 and S (top to bottom), respectively.

detailed motion analysis and visualization of individual motions, and then compare them side by side using the proposed indexes. This demonstrates 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 refer the readers to the supplementary video for more details.

The four boxers chosen have different skill levels. As a ground truth, their skills were evaluated by a professional boxing coach as skillful, medium, medium and novice respectively, and were denoted as S, M1, M2 and N.

6.1. Boxer evaluation

The boxers' posture-based and action-based graphs are shown in Fig. 10, in which letter annotations are given to help explain the graphs. These graphs allow the users to assess boxing skills even if they are not familiar with boxing.

6.2. Boxer S

The first row of images in Fig. 10 shows the graphs of boxer S. The posture-based graphs shows a main standard posture (a) to start and end actions, which is good for boxing as it allows the boxer to transit from one action to another effectively through the standard posture. A large variety of actions (b) can be produced from such a posture. There is a secondary posture in which the arms are further apart (c). This should be avoided as such a posture is weak in blocking attacks. Posture (d) is generated because the boxer trips over during the training. Our system can pick up and visualize such a mistake accurately.

The action-based graph of boxer S shows there there are many actions in the frequent group (a) and only a few in the rare group (b). This shows that the boxer is experienced and his actions are consistent. There is a major movement action (c) in the frequent group (a), and such an action has good connections to many of the others. This is good as experienced boxers typically use movement actions to adjust their position relative to their opponent, and launch attacks when the time is right. Other actions in the frequent group (a) are variations of attacks. For example, the more frequently used action (d) is a right-left combo and action (e) is a single right punch, which show that the boxer tends to start an attack with the right punch. It is good to see that attacking actions may connect to each other, which enhance the unpredictability of the boxer.

6.3. Boxer M1

Next, we evaluate the posture-based graph of boxer M1. The boxer has a main standard posture (a) to launch most of the actions (b). However, he has a secondary posture (c) for launching some attacks, and another (d) for launching a turning action. In both postures, the arms are in a low position and cannot guard the boxer well from the opponent. More importantly, the relatively more frequently used secondary posture (c) is performed with the foot distance much wider than the shoulder width. This means the boxer has limited mobility in this posture, as the legs must move towards each other before another stepping action can be performed. These observations show that the boxer is not as experienced and consistent as boxer S.

The corresponding action-based graph shows that there are fewer frequent class actions (a) but more rare class ones (b) compared to boxer S. This means that that the boxing action of boxer M1 is less consistent. The boxer has a large number of movement actions (c) that are connected to all the rest of the action nodes. He also has a variety of attack actions as shown in other actions in the frequent class (a). In particular, action (d) is a left-right combo and action (e) is a left punch, showing that the boxer tends to start an attack with the left punch. Overall, there is an acceptable number of connections among attacks, demonstrating the acceptable unpredictability of the boxer.

6.4. Boxer M2

For boxer M2's posture-based graph, there is a main standard posture (a) launching the majority of actions (b). There are, however, a number of secondary postures (b)–(d). These postures are all performed sub-optimally with his arms not guarding the head, and should be avoided. Looking closely to the edges (f) going to posture (c), we can find that the posture is performed as a subtle movement to prepare various left punches. This should be avoid as the opponent can tell the moves whenever seeing such a posture. Postures (g) and (h) are very different from the rest, and are geometrically far from the other postures. These two postures are performed because the boxer unintentionally raises the arms during the capture. Our system can pick up the mistake and visualize it in the graph.

Table 1
Statistics of the boxing motions. SL: Skill Level evaluated by a professional boxing coach. PN: Posture Number (for starting and ending actions). AN: Action Number.

	Boxer S	Boxer M1	Boxer M2	Boxer N
SL	Skillful	Medium	Medium	Novice
PN	138	160	112	176
AN	69	80	56	88

From boxer M2's action-based graph, it can be observed that there are relatively fewer actions in the frequent class (a), but a large number of actions in the rare class combining (b) and (c). This shows that the boxer is quite inconsistent in the boxing actions, and could be because of the lack of training and experience. Different from the boxers discussed, boxer M2 has a largest action node (d) of left punch. The second largest action node (e) is a double left punch. The movement action node (f) is relatively small. This shows that boxer M2 has a different boxing style to use left punch as a major action to connect to other actions and his left punch is dominant. Such a boxing style is not advised as a punching action, comparing to a movement one, consume more energy and expose a larger risk of being attacked.

6.5. Boxer N

In the posture-based graph of the novice boxer N, there are two major standard postures (a) and (b) instead of one. There are a large number of self-connecting actions (c) and (d) for both postures, as well as a lot of actions (e) connecting the two. This shows that the boxer is highly inconsistent in the boxing postures. Posture (a), the more relatively frequently used one, is inferior to posture (b), due to its wider foot distance. It does not allow the boxer to step freely. Posture (f), (g) and (h) are all secondary postures with different posture variations. They are all not well performed due to the low arm positions limiting blocking capability, and the wide foot width limiting movement capability.

The corresponding action-based graph shows some actions in the frequent class (a) but a large number of actions in the rare class (b). This means that the novice boxer cannot perform actions consistently. The action in the rare class (b) are mainly very long combo that are randomly combined and cannot be reproduced. The main action (c) is a movement action. Such an action cannot connect to a number of others in the rare class (b), and many actions in the rare class (b) are not well connected. This means that the boxer's action is more predictable, which is bad in a match as the opponent can guess what the boxer may launch next. The two more frequently used attack action (d) and (e) are left-right combo and left punch respectively, showing that the boxer tends to start an attack with a left punch.

6.6. Statistical analysis

Here, we give some statistics about the proposed system.

Table 2
Statistics of the boxing motions. FNN: Fat Node Number (brackets show numbers after removing accidentally created nodes). FEN: Fat Edge Number (brackets show numbers of consistently performed edges). CI: Connectivity Index.

	Boxer S	Boxer M1	Boxer M2	Boxer N
FNN	3 (2)	6 (4)	3 (3)	5 (5)
FEN	20 (10)	36 (12)	16 (7)	57 (8)
CI	5.0	3.0	2.3	1.6

Table 3
Statistics of the boxing motions in the Aciton Graphs. NN: Node Number. NNFC: Node Number for Frequent Class. NNRC: Node Number for Rare Class. EN: Edge Number. ASI: Action Strategy Index.

	Boxer S	Boxer M1	Boxer M2	Boxer N
NN	7	11	9	16
NNFC	4	3	4	5
NNRC	3	8	5	11
EN	16	27	20	38
ASI	0.572	0.448	0.426	0.378

Table 1 shows the skill level assessed by a professional boxing coach, as well as the number of postures and actions, for each of the boxers considered.

Table 2 shows the statistics related to the posture-based graph, including the number of fat nodes and fat edges, as well as the Connectivity Index calculated with Eq. (3). The index evaluates the richness of actions and the flexibility of transitions. It aligns with the boxers' skill level and more skillful boxers have higher Connectivity Indexes.

Table 3 shows the statistics related to the action-based graph, including the number of HMM nodes (which is further separated into the number for the frequent class and the rare class respectively) and HMM edges, as well as the Action Strategy Index calculated with Eq. (5). It indicates the unpredictability of a boxer, and more skillful boxers are generally more unpredictable. Again, it aligns with the boxer' skill level and more skillful boxers have higher Action Strategy Indexes.

In terms of the computational cost, we run the proposed system on a laptop computer with a Core i7-6820HQ CPU, 16GB of RAM and a NVIDIA Quadro M1000M graphic card. The computational time to analyze the captured motion (Section 3.2) and computing the graphs (Sections 4 and 5) ranges from 6 to 9 s. The variation of computational time is mainly due to the iterative k-means clustering algorithm for both postures and actions, as a larger k requires longer computational time. The run-time cost is low and we achieve frame rate higher than real-time (i.e. 60Hz). The frame rate tends to be lower when there are more characters shown in the graphs.

7. Conclusion and discussions

In this paper, we proposed a method to visualize the high-level skills of boxers using an automatic motion analysis and visualization framework. The proposed posture-based graph is a customized Fat Graph that allows us to evaluate the quality of standard postures for launching and finishing actions. The action-based graph is a customized Hidden Markov Model that visualizes the transition probability among actions. We further introduce the Connectivity Index that is deduced from the posture-based graph and allows evaluation of the richness of actions and the flexibility of transitions, as well as the Action Strategy Index that is deduced from the action-based graph and allows evaluation of the unpredictability of action patterns. The system is applied on the motion captured from 4 boxers with varying skill levels. The evaluations from our system aligns with that of a professional boxing coach.

Although we use boxing as our target sport in the experimentation section, the underpinning theoretical development can be applied to most sports that require swiftness, flexibility and creativity, such as tennis, fencing and basketball. The adaptation of the proposed system to these sports and the comparison of the system performances on different sports remain as future work.

We focus on analyzing the skill level of the boxers in terms of high-level motion behaviour such as the richness of the action, the transition of action and the unpredictability of boxing patterns. 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 key actions. Admittedly, not all sports follow this pattern. Second, the visualization and skill assessment is based on an individual athlete, not considering skills related to collaborations such as those in group sports, in which the assessment might need to employ different criteria.

We argue that novice boxers tend to have different posture-based graphs, while experienced boxers tend to have graphs of a similar topology. This is because unlike experience boxers who have only 1 to 2 main postures nodes, novice boxers tend to have more nodes, resulting in a much larger variation on the graph topology. As a future work, we would like to utilize the stem to evaluate a large number of boxers in different skill levels to verify this argument.

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.

Acknowledgment

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1) and the Royal Society (Ref: IE160609).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cag.2017.09.007

References

[1] Komura T, Kuroda A, Shinagawa Y. Nicemeetvr: facing professional baseball pitchers in the virtual batting cage. In: Proceedings of the 2002 ACM symposium on applied computing. In: SAC '02. New York, NY, USA: ACM; 2002. p. 1060–5. ISBN 1-58113-445-2. doi: 10.1145/508791.509000.
[2] Bideau B, Kulpa R, Menardais S, Fradet L, Multon F, Delamarche P, et al. Real handball goalkeeper vs. virtual handball thrower. Presence: Teleoper Virtual Environ 2003;12(4):411–21.
[3] Molet T, Aubel A, Capin T, Carion S, Lee E, Magnenat-Thalmann N, et al. Anyone for tennis? Presence: Teleoper Virtual Environ 1999;8(2):140–56. doi:10.1162/105474699566134.

[4] Hyun K, Lee K, Lee J. Motion grammars for character animation. Comput Graph Forum 2016;35(2):103–13. doi:10.1111/cgf.12815.

[5] Wampler K, Andersen E, Herbst E, Lee Y, Popović Z. Character animation in two-player adversarial games. ACM Trans Graph 2010;29(3):26-1–26-13. doi:10.1145/1805964.1805970.

[6] Shin HJ, Oh HS. Fat graphs: constructing an interactive character with continuous controls. Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on computer animation. SCA '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association; 2006. p. 291–8. ISBN 3-905673-34-7, Vienna, Austria.

[7] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;77(2):257–86. doi:10.1109/5.18626.

[8] Shum HPH, Wang H, Ho ESL, Komura T. Skillvis: a visualization tool for boxing skill assessment. In: Proceedings of the 2016 international conference on motion in games. In: MIG '16. New York, NY, USA: ACM; 2016. p. 145–53. ISBN 978-1-4503-4592-7. doi: 10.1145/2994258.2994266.

[9] Yeadon M. The simulation of aerial movement- iv. a computer simulation model. J Biomech 1990;23(1):85–9.

[10] Yeadon M. The biomechanics of the human in flight. Am J Sports Med 1997;25(4):575–80.

[11] Arikan O, Forsyth D. Motion generation from examples. ACM Trans Graph 2002;21(3):483–90.

[12] Lee J, Chai J, Reitsma PSA, Hodgins JK, Pollard NS. Interactive control of avatars animated with human motion data. ACM Trans Graph 2002;21(3):491–500.

[13] Kovar L, Gleicher M, Pighin F. Motion graphs. ACM Trans Graph 2002;21(3):473–82.

[14] Grochow K, Martin SL, Hertzmann A, Popovic' Z. Style-based inverse kinematics. ACM Trans Graph (TOG) 2004;22(3).

[15] Shin HJ, Lee J. Motion synthesis and editing in low-dimensional spaces. Comput Animat Virtual Worlds (Special Issue: CASA 2006) 2006;17(3–4):219–27.

[16] Min J, Chai J. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. ACM Trans Graph 2012;31(6):153.

[17] Beaudoin P, Coros S, van de Panne M, Poulin P. Motion-motif graphs. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on computer animation. In: SCA '08. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association; 2008. p. 117–26. ISBN 978-3-905674-10-1

[18] Arikan O, Forsyth DA, O'Brien JF. Motion synthesis from annotations. ACM Trans Graph 2003;22(3):402–8. doi:10.1145/882262.882284.

[19] Li Y, Wang T, Shum H-Y. Motion texture: a two-level statistical model for character motion synthesis. ACM Trans Graph 2002;21(3):465–72. doi:10.1145/566654.566604.

[20] Reitsma PSA, Pollard NS. Evaluating motion graphs for character animation. ACM Trans Graph 2007;26(4). doi:10.1145/1289603.1289609.

[21] Heck R, Gleicher M. Parametric motion graphs. In: Proceedings of the 2007 symposium on interactive 3D graphics and games. In: I3D '07. New York, NY, USA: ACM; 2007. p. 129–36. ISBN 978-1-59593-628-8. doi: 10.1145/1230100.1230123.

[22] Xia S, Wang C, Chai J, Hodgins J. Realtime style transfer for unlabeled heterogeneous human motion. ACM Trans Graph 2015;34(4):119-1–119-10. doi:10.1145/2766999.

[23] Safonova A, Hodgins JK. Construction and optimal search of interpolated motion graphs. ACM Trans Graph 2007;26(3). doi:10.1145/1276377.1276510.

[24] Shum HPH, Komura T, Shiraishi M, Yamazaki S. Interaction patches for multi-character animation. ACM Trans Graph 2008;27(5):114-1–114-8. doi:10.1145/1409060.1409067.

[25] Shum HPH, Komura T, Yamazaki S. Simulating multiple character interactions with collaborative and adversarial goals. IEEE Trans Vis Comput Graph 2012;18(5):741–52. doi:10.1109/TVCG.2010.257.

[26] Ho ESL, Komura T. A finite state machine based on topology coordinates for wrestling games. Comput Animat Virtual Worlds 2011;22(5):435–43. doi:10.1002/cav.376.

[27] Ho ES, Komura T. Character motion synthesis by topology coordinates. Comput Graph Forum 2009. doi:10.1111/j.1467-8659.2009.01369.x.

[28] Plantard P, Shum HPH, Multon F. Filtered pose graph for efficient kinect pose reconstruction. J Multimed Tools Appl 2017;76(3):4291–312. doi:10.1007/s11042-016-3546-4.

[29] Plantard P, Shum HPH, Pierres A-S L, Multon F. Validation of an ergonomic assessment method using kinect data in real workplace conditions. Appl Ergon 2016. doi:10.1016/j.apergo.2016.10.015.

[30] Shum HPH, Komura T, Shiratori T, Takagi S. Physically-based character control in low dimensional space. In: Proceedings of the third international conference on motion in games. In: MIG '10, vol. 6459. Berlin, Heidelberg: Springer-Verlag; 2010. p. 23–34. ISBN 978-3-642-16957-1. doi: 10.1007/978-3-642-16958-8_3.

[31] Shum HPH, Ho ESL, Jiang Y, Takagi S. Real-time posture reconstruction for microsoft kinect. IEEE Trans Cybern 2013;43(5):1357–69. doi:10.1109/TCYB.2013.2275945.

[32] Ho ESL, Shum HPH, Cheung Y-m, Yuen PC. Topology aware data-driven inverse kinematics. Comput Graph Forum 2013;32(7):61–70. doi:10.1111/cgf.12212.

[33] Lawrence ND. Gaussian process latent variable models for visualisation of high dimensional data. In: Proceedings of the 2003 neural information processing systems; 2003.

[34] Wang JM, Fleet DJ, Hertzmann A. Gaussian process dynamical models for human motion. IEEE Trans Pattern Anal Mach Intell 2008;30(2):283–98. doi:10.1109/TPAMI.2007.1167.

[35] Holden D, Saito J, Komura T. A deep learning framework for character motion synthesis and editing. ACM Trans Graph 2016;35(4):138-1–138-11. doi:10.1145/2897824.2925975.

[36] Liu Z, Zhou L, Leung H, Shum HPH. Kinect posture reconstruction based on a local mixture of gaussian process models. IEEE Trans Vis Comput Graph 2016. doi:10.1109/TVCG.2015.2510000.

[37] Hara K, Omori T, Ueno R. Detection of unusual human behavior in intelligent house. In: Proceedings of the 12th IEEE workshop on neural networks for signal processing; 2002. p. 697–706. doi:10.1109/NNSP.2002.1030081.

[38] Françoise J, Roby-Brami A, Riboud N, Bevilacqua F. Movement sequence analysis using hidden Markov models: a case study in tai chi performance. In: Proceedings of the 2nd international workshop on movement and computing. In: MOCO '15. New York, NY, USA: ACM; 2015. p. 29–36. ISBN 978-1-4503-3457-0. doi: 10.1145/2790994.2791006.

[39] Brand M, Hertzmann A. Style machines. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques. In: SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 2000. p. 183–92. ISBN 1-58113-208-5. doi: 10.1145/344779.344865.

[40] Tanco LM, Hilton A. Realistic synthesis of novel human movements from a database of motion capture examples. In: Proceedings of workshop on human motion; 2000. p. 137–42. doi:10.1109/HUMO.2000.897383.

[41] Ren L, Patrick A, Efros AA, Hodgins JK, Rehg JM. A data-driven approach to quantifying natural human motion. ACM Trans Graph 2005;24(3):1090–7. doi:10.1145/1073204.1073316.