

Hierarchical Event Selection for Video Storyboards with a Case Study on Snooker Video Visualization

Matthew L. Parry, Philip A. Legg, David H.S. Chung, Iwan W. Griffiths, and Min Chen

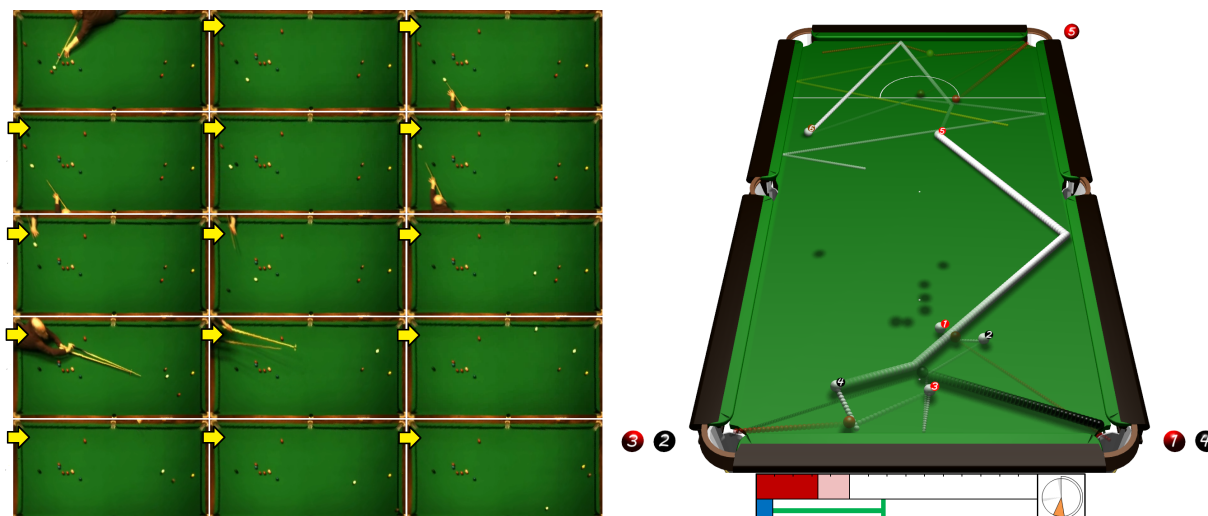


Fig. 1. Left: Keyframes from a 3 minute snooker video segment consisting of 6 shots taken from a complete match of 67 shots. Right: Visualization created from the video data. The visualization displays each shot from the video, whilst introducing event importance shown by ball trajectory emphasis, and event ordering shown by numbered ball icons. Table annotations represent ball pots, player scores, points remaining and the position in the full video where this sequence of events occurs.

Abstract—Video storyboard, which is a form of video visualization, summarizes the major events in a video using illustrative visualization. There are three main technical challenges in creating a video storyboard, (a) *event classification*, (b) *event selection* and (c) *event illustration*. Among these challenges, (a) is highly application-dependent and requires a significant amount of application-specific semantics to be encoded in a system or manually specified by users. This paper focuses on challenges (b) and (c). In particular, we present a framework for hierarchical event representation, and an importance-based selection algorithm for supporting the creation of a video storyboard from a video. We consider the storyboard to be an event summarization for the whole video, whilst each individual illustration on the board is also an event summarization but for a smaller time window. We utilized a 3D visualization template for depicting and annotating events in illustrations. To demonstrate the concepts and algorithms developed, we use Snooker video visualization as a case study, because it has a concrete and agreeable set of semantic definitions for events and can make use of existing techniques of event detection and 3D reconstruction in a reliable manner. Nevertheless, most of our concepts and algorithms developed for challenges (b) and (c) can be applied to other application areas.

Index Terms—Multimedia visualization, Time series data, Illustrative visualization.

1 INTRODUCTION

Video visualization is concerned with the creation of a new visual representation from an input video to reveal important features and events in the video [2]. A *video storyboard*, which is a form of video visualization, typically summarizes a video using a small sequence of keyframes and composite images that are often enhanced by additional

illustrative annotations. Unlike those storyboards used in movie making as a pre-visualization of the planned actions, here we focus on video storyboards as a post-visualization of the events in a video. It helps users primarily as a visual medium to aid discussions and comparison, facilitating memory externalization while removing the burden of viewing videos repeatedly.

Video storyboards have been proposed in the literature (e.g., [9]). However, the technical advances have been largely limited to annotating motions, such as an actor moving from one place to another [9]. The difficulties reside in the pipeline for creating a storyboard from a video, which typically consists of (a) event detection and classification, (b) event prioritization and selection, and (c) event depiction and annotation. One of the major obstacles is the need for encoding some application-specific semantics at some stages of the pipeline, fogging generic approaches with application-specific algorithms. Among the three stages, (a) depends heavily on application-specific semantics and algorithmic encoding (e.g., object and event classification is usually difficult to port from one application to another). Techniques for (c) may require some application-specific specification such as visual de-

- Matthew L. Parry, Philip A. Legg and David H.S. Chung are with the Department of Computer Science, the College of Science, and the College of Engineering, Swansea University, UK.
- Iwan W. Griffiths is with the Department of Sports Science, the College of Engineering, Swansea University, UK.
- Min Chen was with Swansea University, and is now with the Oxford e-Research Centre, University of Oxford, UK.

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.

For information on obtaining reprints of this article, please send email to: twcg@computer.org.

signs and mappings, but are generally more portable as they rarely involve machine learning. Techniques for (b) are potentially more generic and can have many applications. It is necessary to note that although the creation of video storyboard requires the input and encoding of application-specific semantics, this does not undermine its potential as a vital tool for video visualization because many applications, such as sports, have the needs and resources to build such an application-specific pipeline.

In this work, we focus on (b), examining a generic scheme for representing events hierarchically and a recursive algorithm for selecting events for creating storyboards (Section 3). To demonstrate the usability of this scheme, we present a case study of snooker video visualization, where the hierarchical selection approach is particularized (Section 4). The reason to choose this case study is that the definition and importance of events in snooker games are reasonably well-defined and agreeable by most experts and audience. As the filming takes place indoor under a controlled environment, this makes object and event detection and classification more reliable (Section 5), allowing us to focus on (b) without being distracted by application-specific difficulties. We utilized a form of 3D perspective view of a snooker table as the basic template for our visualization, because this is consistent with what users would commonly see on the television. We show that the hierarchical event prioritization and selection can be integrated directly with algorithms for visual mappings (Section 6).

2 RELATED WORKS

Whilst the topic of event visualization is still relatively new, event detection has been studied for many years in the computer vision community. It is widely adopted in surveillance work, as shown in the survey by Hu *et al.* [12]. In particular, it is often used for traffic monitoring [17] or for human behaviour analysis [29]. Another popular application of event detection has been sports broadcasting. Li and Sezan [16] propose to use a Hidden Markov Model to classify events in sports videos (e.g., for American football, baseball and sumo wrestling) into four basic states: start-of-play, in-play, end-of-play and non-play. Sadlier and O’Connor [21] use both audio and visual features to detect events in sports video by using a Support Vector Machine. Pan *et al.* [18] perform event detection based on detecting slow-motion replays that are likely to correspond to major event highlights in broadcast footage.

Video visualization compliments automated video analysis, especially in situations where accurate detection and classification cannot be achieved. Borgo *et al.* [2] conduct a comprehensive survey on this topic. Daniel and Chen [4] use video visualization to summarize CCTV footage. Chen *et al.* [3] propose to improve visualization using both volume (change) and flow (motion) signatures. Their study show that ordinary people can learn to recognise events based on event signatures in static visualization rather than having to view the entire video content. Romero *et al.* [20] use video visualization to add human behavioral analysis. Yeo and Yeung [27] implement a system that uses keyframes to summarize video. Since using keyframes is common practice, this is not a new idea, but a new implementation. Takahashi *et al.* [24] propose to summarize video footage using a ‘video poster’ created from the keyframes and meta-data of sports video. Dony *et al.* [6] and Goldman *et al.* [9] propose to summarize videos using storyboards, which accompany keyframes with annotations (such as motion arrows). Assa *et al.* [1] extract a selection of poses from a short sports video and compose them into a single illustration.

In the context of event-based visualization, Kapler and Wright developed a prototype system ‘GeoTime’ that displays military events in a combined temporal and geo-spatial visualization [13]. They use the 2D x-y space for the geographic space, and the third dimension (z) to represent time in the future and past. Gatalsky *et al.* use the similar concept of the ‘space-time cube’ to visualize spatio-temporal information relating to earthquake events [8]. Wang *et al.* [25] use 3D environment models to contextualize spatially-related videos. Yu *et al.* [28] propose to organize identified events into an event graph to aid the creation of animation of different event sequences.

Finally, for snooker videos, Höferlin *et al.* [11] present a study on using video visualization in snooker coaching and skill training. Rea *et al.* [19] perform video analysis for classifying snooker events, in particular: shot-to-nothing, break building, conservative play and snooker escapes. Denman *et al.* [5] make use of table geometry in processing snooker broadcast videos, and propose to detect the disappearance of objects using histogram analysis at pocket regions. Guo and Namee [10] perform 3D reconstruction using a smallscale toy snooker table with a camera placed directly above. Shen and Wu [22] carry out 3D reconstruction from a camera placed above the centre of the table. They acknowledge the difficulty to capture the entire table because of the altitude of the camera.

3 FACETS OF EVENTS

An *event* can be described as a significant occurrence or happening, which typically has some application-specific meaning in the context where it occurs. A conceptual framework of events may feature several major facets of events. These may include:

- *Hierarchy* — Events may have a hierarchical relationship, that is, an event may be defined as an abstraction or a composition of a series of more elemental events.
- *Importance* — Events may have different levels of importance. Such a quantity can be defined by a group of experts for a specific application.
- *State* — Events may act as a transition, or one of the causes of a transition, from one state to another [23].

It may appear that the specification of events and their above-mentioned facets could be rather arbitrary and might be judged by different individuals in different circumstances. However, in many applications, such as sports games and road traffic, there exists a common understanding. It is such a common understanding that makes event detection and classification possible. On the other hand, the personal and circumstantial variance makes event summarization a challenging task. In comparison with a textual summarization, event visualization has the potential to convey a summary of major events, and their temporal and spatial context in a more effective and efficient manner, while supporting industrials’ reasoning and judgment.

3.1 Event Hierarchy

Let e_w^q denote an event at a time window w in a semantic context q . Without losing generality, we treat a specific point of time t and an ordinal number i as two special cases of w . In the following discussions, we also assume that semantic contexts are organized into a *hierarchy*. Thus, e_t^i implies that it is a level h event occurring at time t . When discussing events at a particular hierarchical level, we may use a specific letter (e.g., a , b) to denote all events at that level.

For example, consider all image frames in a snooker video:

$$E^1 = F = \langle f_1, f_2, \dots, f_t, \dots \rangle.$$

F can be considered as the most elementary level of events. All ball motions and contacts with cues, cushions and pockets form a sequence of events at a higher level,

$$E^2 = C = \langle c_1, c_2, \dots, c_i, \dots \rangle.$$

All shots, from the first contact between the cue and ball to the time when all balls become stationary, fall into the 3rd level,

$$E^3 = W = \langle w_1, w_2, \dots, w_j, \dots \rangle.$$

A collection of consecutive shots that are related forms an event at the 4th level, and all such events belong to the sequence of

$$E^4 = P = \langle p_1, p_2, \dots, p_k, \dots \rangle.$$

All game frames becomes a sequence at the 5th level,

$$E^5 = R = \langle r_1, r_2, \dots, r_l, \dots \rangle.$$

This framework can be extended to include even higher levels such as games, tournaments or seasons. We use symbol \prec to denote the hierarchical order, e.g.,

$$F \prec C \prec W \prec P \prec R \prec \dots$$

Conceptually, events at each level are interleaved with a sequence of *states* at the same level. Each state is unique, and can be determined based on a set of *attributes*. In practice, it is usually more meaningful and efficient to represent states explicitly only at higher levels. We use s_w^q to denote a state at a time window w in a semantic context q . Similar to event definitions, we can have special cases for a specific point in time, order and hierarchy.

3.2 Event Classification

Classification is a process to assign a semantic meaning to an event or state. This is highly application-dependent. For example, in soccer video annotation, it is common for a video analyst to observe an event in a video, create an event mark, and select a type from a list of pre-defined keywords. For snooker videos, it is more feasible to determine a low level event by an automatic classification algorithm.

When considering an event as a record in the computer, we can call an event classification function $\Psi(e)$ to determine the *type* of the event. For example, events in sequence C may be classified as (i) no collision, (ii) ball-to-ball, (iii) cue-to-ball, (iv) ball-to-cushion, (v) ball-near-pocket, and (vi) ball-in-pocket. Events in sequence W may be classified as (i) break building, (ii) conservative play, (iii) snooker escape and (iv) shot to nothing. Similarly, states are classified by a function $\Phi(s)$, such that Ψ and Φ are level dependent.

In general, a higher level event is an abstraction or composition of some lower level events. For instance, a motion event m is determined by examining several consecutive image frames $\dots, f_{i-2}, f_{i-1}, f_i$. A more complex classification scheme may involve processing both events and states at lower levels. In our work, we restrict the classification process only to the level immediately below the current level. Note that this is only a convenient implementation, not a limitation, since one can always ‘copy’ an event from a lower level to a higher level. Hence, given three event sequences at different levels, $X \prec Y \prec Z$, if the definition of an event $z \in Z$ depends on some events in X and some in Y , we can always add new events in Y to mirror those relevant events in X .

We thereby have the following classification dependency:

$$\Psi : (E^h)^d \times (S^h)^d \mapsto E^{h+1} \quad \Phi : (E^h)^d \times (S^h)^d \mapsto S^{h+1}$$

where d is the small constant representing the number of events and states in a sequence of transitions that a classification scheme will consider. As almost all events and states are type-defined by Ψ or Φ (except the image frames in F), event detection and classification conform to the general problem of pattern recognition in text strings [14].

For instance, given a sequence of event-state transition, $A = \langle e_1, s_1, e_2, s_2, \dots, e_d, s_d \rangle$, we can compute a corresponding string of types as:

$$T_A = \langle \Psi(e_1), \Phi(s_1), \Psi(e_2), \dots, \Psi(e_d), \Phi(s_d) \rangle$$

We then search for a particular type-signature sequence in T_A .

3.3 Event Importance and Selection

Given an event record e , we can also determine the *importance* of the event by using an importance function $\Gamma(e)$. Unlike event and state classification functions, $\Gamma(e)$ operates on the events and states at the same level. A simple importance function can simply be a mapping from event type to a scalar value representing the importance. A slightly sophisticated function may take the preceding state and succeeding state into account. A more complex function may be a function of a sequence of event-state transitions in a manner similar to the processing of T_A in Section 3.2.

In any form of event summarization, including video storyboarding, we need to select a set of events to be communicated to the users. The

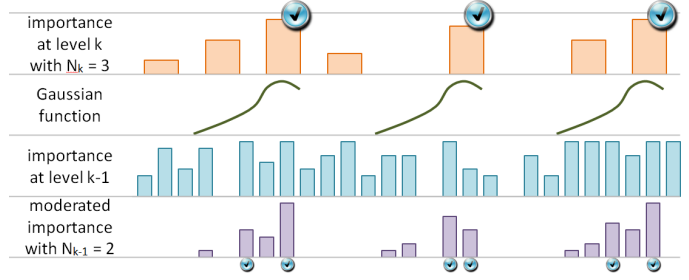


Fig. 2. Example to show event importance and selection. Events to be selected for video storyboard at level k and $k - 1$ are shown by the blue ‘tick’ icon. This illustrates how the Gaussian function introduces a temporal measure into the moderated importance criteria.

importance of a higher level event will have significant influence on the selection of lower level events. In many ways, this is similar to storytelling. The selection of a lower level event (e.g., pick up a knife) for the story would depend heavily on the importance of the related event at a higher level (e.g., fighting or cooking).

Once importance for each event is computed, we can start to select events for event summarization. In this work, a summarization is also organized hierarchically. Again, similar to storytelling, a story consists of a number of sub-stories. Each sub-story consists of a number of sub-sub-stories. There is an overall control about the coverage in breadth (i.e., how many sub-stories are allowed) and depth (i.e., how many levels of details).

Our *event selection* algorithm assumes the input of the following initial conditions from the user: the particular levels of events to be included in the summary, the maximum number of events at the highest level to be included in the summary, the time period covered, and a few other parameters to be given later in the relevant context.

Let $H_1 \prec H_2 \prec \dots \prec H_k$ be the k levels of events to be visualized. N_k be the maximum number of events to be depicted at level H_k . The selection algorithm first selects N_k events at level H_k with the highest importance values, resulting in a sequence:

$$A = \langle a_1, a_2, \dots, a_n \rangle$$

where each event a_i has an importance value α_i .

For each event $a_i \in A$ at level k , we identify a sequence of relevant events at level $k - 1$. The relevance is defined by a time period, $[t - \delta_1, t + \delta_2]$ where t is the point in time of a_i , and $\delta_1, \delta_2 \geq 0$. Often, we set $\delta_2 = 0$ as the depiction of an event usually involves showing more information about the sub-events leading the event.

Let $B_i = \langle b_{i,1}, b_{i,2}, \dots, b_{i,m} \rangle$ be a sequence of such sub-events at level $k - 1$. Each $b_{i,j}$ is defined with its importance β_j and time u_j .

The importance value β_j is moderated by a Gaussian function $G(x)$ where $x = |u_j - t|$. The parameter σ of $G(x)$ is pre-defined, and we set $\sigma = 2$ as a default in our system. This Gaussian function gradually reduces the importance of events in B_i if they are further away from t , which is the temporal focus of a_i . Let $\beta'_1, \beta'_2, \dots, \beta'_m$ be the resultant importance values after Gaussian moderation. Let N_{k-1} be a local maximum number of events at level $k - 1$ to be visualized for each event a_i at level k . We select N_{k-1} events from B_i based on with the highest importance values in $\langle \beta'_1, \beta'_2, \dots, \beta'_m \rangle$. Fig. 2 illustrates this moderation process, which can continue recursively.

4 CASE STUDY: SNOOKER VIDEO VISUALIZATION

We now present the proposed framework in application to sports video event visualization. We choose snooker as an example since it consists of well-defined events that can be clearly represented. Snooker is played on a large green baize-covered table (of size 12×6 ft) with pockets in each of the four corners and in the middle of each of the long side cushions [7]. It is played using a cue and a set of snooker balls, consisting of one white cue ball, 15 red balls (worth one point each), and a series of coloured balls: yellow (worth two points), green (worth

E^4	Break Phase		Safety Exchange		Tactical Play	
E^3	Break Building	Conservative Play	Snooker Escape	Shot-to-Nothing	Foul	
E^2	Ball Motion		Ball Collision		Ball Pot	

Fig. 3. Snooker event hierarchy. This shows low-level events that occur within a particular shot (E^2), higher-level events that define a single shot (E^3) and higher-level events that group a collection of shots (E^4).

three points), brown (worth four points), blue (worth five points), pink (worth six points) and black (worth seven points). The aim of the game is to score as many points as possible by striking the cue ball with the cue in order to pot the red and coloured balls in sequential order.

4.1 Snooker Event Hierarchy

Fig. 3 gives the event hierarchy that was adopted for snooker storyboarding. The hierarchy shows three event levels, where $E^2 \prec E^3 \prec E^4$. Given each ball on the table we define the detectable low-level events into the following classes $\Psi(c) = \{\text{moving, collision, pot}\}$, $c \in E^2$. It is assumed that at the start and end of each shot that each ball will take the event $\psi_{stopped}$. Let us now consider the series of low-level events E^2 that would make a typical shot in snooker. The player strikes the cueball (triggering the cueball event *moving*) and collides with another object ball (triggering the cueball event *collision*). If this results in the object ball event *pot* being triggered then the player takes another shot, otherwise the opponent takes their turn.

Each shot is clearly bounded between the sequence of events that results in each ball object being stationary on the table. Hence we can begin to formulate higher level events E^3 by collecting the events and states occurring between this interval. As discussed in Section 3, higher-level events are made up of lower level events, subject to satisfying the condition of states. By searching for a particular signature contained in E^3 , we assign each shot to one of the following types: *break building*, *conservative play*, *snooker escape*, *shot-to-nothing* and *foul*.

In order to define the states between events, we consider a set of attributes α , each of which can take a particular value. Then any state can be defined as $s = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$. For snooker we use the following attributes: *player1_score*, *player2_score*, *current_break*, *points_remaining_on_table*, *cueball_safe*, *valid_shot_played*. Likewise, these attributes may have additional parameters associated with them (e.g., *cueball_safe* depends on *cueball_position* and *distance_to_closest_red*). At any time window w , the set of attributes can be assessed to determine the current state. Due to the nature of snooker, it is reasonable to consider these attributes for states at level E^3 which coincides with each independent shot in the match. After each shot at event level E^3 , each attribute is updated to determine the new state before the next shot is played.

Let us consider the events at level E^3 . Break building is defined where $\alpha_{valid_shot_played}$ is true, and the event ψ_{pot} occurs. Similarly, a shot where $\alpha_{valid_shot_played}$ is true but the event ψ_{pot} does not occur can be considered to be conservative play, subject to $\alpha_{cueball_safe}$ being true. A shot-to-nothing can be defined where ψ_{pot} occurs with attribute $\alpha_{cueball_safe}$ being true. We deduce a snooker-escape by looking for the occurrence of a collision with the cushion before making contact with the ball, subject to $\alpha_{cueball_safe}$ being true. Finally, a foul occurs when $\alpha_{valid_shot_played}$ returns false.

For our highest event level E^4 , we begin to cluster together relevant shots based on the context of a regular game of snooker. A collection of break building shots can be grouped to give a break phase where a player has potted multiple balls. Likewise, a collection of shots classified as conservative play could be grouped to be a safety exchange. In some cases, it may also be important to know the shot that came before or after an event (e.g., poor cueball positioning that allowed a break to be scored, or a missed pot that ended a break). Hence we type define ψ_{break_phase} as a sequence of consecutive break building shots along with the shots that precede directly before and after. We use the

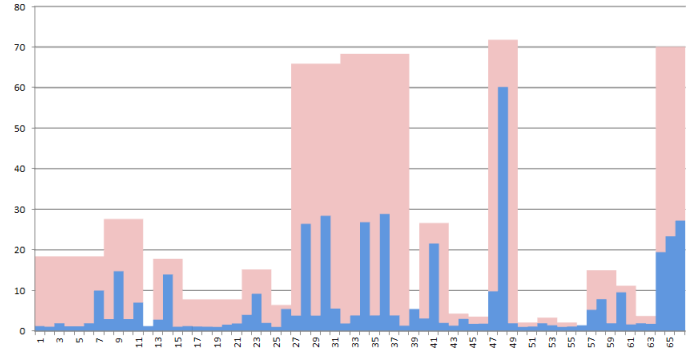


Fig. 4. A bar chart of event importance (X-axis gives shot number compared against Y-axis for importance). Events at level E^3 are shown in blue and events at level E^4 are shown in red. Four significantly large peaks occur for E^4 that represent the key events for that level.

term ‘tactical play’ to categorize shots that involve escaping from a snooker. Again, we consider the preceding shot in order to establish the action that took place that results in the snooker.

4.2 Snooker Event Importance

Fig. 4 shows a bar chart that represents the event importance for a snooker match, illustrating the importance of events at levels E^3 and E^4 . The 4 peaks in E^4 corresponds to the 3 largest breaks in the game and the phase when the frame ball was potted. Within each region, we can see local peaks that show the more important individual shots (e.g., potting the black ball) that are derived from the general form:

$$\Gamma(\psi_{type}) = b + v(\alpha) \quad (1)$$

where b is a constant that gives the baseline importance value for each event type ψ_{type} and $v(\alpha)$ denotes the variable importance computed based on the current attributes in the state. Given the higher level events and the state attributes, we define our following importance functions based on our general definition:

$$\Gamma(\psi_{conservative_play}) = 1 + (1 - |\alpha_{cueball_p_osition}|) \quad (2)$$

$$\Gamma(\psi_{break_building}) = \beta_c + \beta_c \times \ln(\alpha_{current_break}) \quad (3)$$

$$\Gamma(\psi_{snooker_escape}) = 2 + \frac{1}{e^{1/\eta}} \quad (4)$$

$$\Gamma(\psi_{shot_to_nothing}) = 4 + |\alpha_{cueball_p_osition}| \quad (5)$$

where $\eta \in \mathbb{N}$ is the number of cue ball collisions before contact with object ball and β_c corresponds to the score of potted ball colour (where $c = 1 \dots 7$).

From Eq. 2, the importance value for a typical conservative shot is bounded between the interval $[1, 2]$ where $\Gamma(\psi_{conservative_play}) \rightarrow 2$ as $\alpha_{cueball_p_osition} \rightarrow 0$. A good conservative shot generally involves leaving the cue ball near the baulk end of the table which in our case is when the cue ball x-position is at its maximum. As a general frame of snooker can contain a large number of good conservative shots, we indicate the bad conservative shots (i.e., when ξ tends to zero) with higher importance as they occur less frequently. Similarly, this is also applied in computing the importance for ‘shot-to-nothing’ in Eq. 5. However, in this case we accentuate that the pot is more difficult when the cue ball is closer to the top cushion (when $\alpha_{cueball_p_osition} \rightarrow 1$).

Break building is derived using a two-stage iteration. We use the value of the potted ball as the base score followed by an additional factor of $\beta_c \times \alpha_{current_break}$. The higher the break that a player makes, the more important that particular break becomes. We use a natural logarithmic function to model the behaviour of a break being gradually less important once a player has completed a break to secure a win.

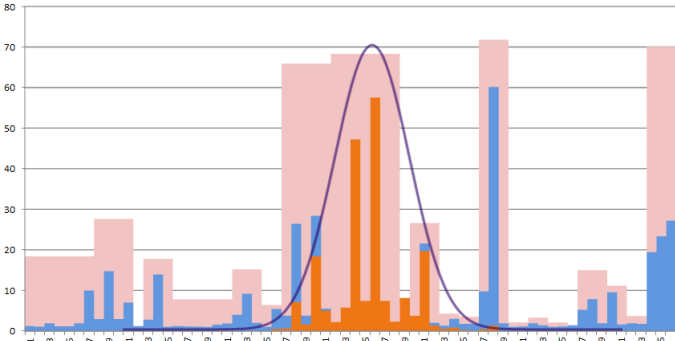


Fig. 5. Applying Gaussian moderation as a weighting parameter for event importance. The adjusted importance parameters are shown in green. This introduces a temporal focus surrounding the key local event that occurs within the grouping given by the global importance.

The most important shot in a frame is considered to be ‘frame ball’. This is where the difference between the player scores is greater than the number of points remaining on the table. The importance of frame ball is defined as:

$$\frac{147}{\alpha_{points_remaining_on_table}} \times \frac{\max(\alpha_{player1_score}, 1) \max(\alpha_{player2_score}, 1)}{|\alpha_{player1_score} - \alpha_{player2_score}|} \quad (6)$$

Note that frame ball is only considered when:

$$|\alpha_{player1_score} - \alpha_{player2_score}| > \alpha_{points_remaining_on_table}. \quad (7)$$

A frame ball is more significant if the remaining score on the table is small (this will occur if the game happens to be close) or if the score difference between both players is small.

Fig. 5 shows Gaussian moderation being applied to the original event importance (Fig. 4) as discussed in Section 3. For a particular higher-level event, the Gaussian curve is centred on the greatest low-level event within this set. The Gaussian curve is used to apply a weighting to the importance in order to provide a temporal focus surrounding the key event.

5 SYSTEM OVERVIEW

The system can be described as a three-stage process, (a) event detection and classification, (b) event prioritization and selection, and (c) event depiction and annotation. In (a), the input video is processed to detect the snooker table and ball objects, and based on the tracking data, detects the occurrence of low-level events. From this, the system is able to produce the hierarchical event classification. Given the event hierarchy, (b) computes the importance of each event that occurs. From this, the system selects the shots of greatest importance. Finally, (c) generates the visualization by using the event importance and selection data in conjunction with the tracking data.

We capture video footage from the snooker table using a single camera that is mounted above the table. The process of detecting, classifying and tracking each ball object within the captured scene (Fig. 6) is given in [15]. For each shot in the match, and each frame of video in the shot, we obtain the following data for each ball: ID, colour, position, speed, direction. This provides enough data to generate a 3D reconstruction of the captured scene. Extending on our previous work, each ball will also have three event tags that correspond to the three low-level events: *moving*, *collision* and *pot*. A ball is detected as moving if the speed is greater than a fixed parameter, based on the position between subsequent frames in the video. A collision is detected where a ball starts moving, subject to another ball moving close to this. Finally, a pot is detected where a ball disappears from the table, subject to being close to a pocket region. These make up the low-level events that are used for further processing by the event selection tool.



Fig. 6. Conversion from video view to 3D reconstructed view.

6 VISUALIZATION DESIGN

A *storyboard* consists of a series of *illustrations*. Each illustration represents a major event selected by the event selection algorithm described in Section 3. Unlike the conventional storyboard used in computer animation, each of our illustrations is not just a keyframe, but a visualization of several sub-events related to the major event associated with this illustration. As described in Section 3, the selection of these sub-events is based on Gaussian-moderated importance values. In this section we shall discuss the visualization design used to generate the illustrations, and how these form the video storyboard.

It is important that our visualization design follows best practice guidelines. There are existing guidelines for how to use different visual cues such as colour, thickness/size, opacity, lines, texture and symbols when producing visualizations [26]. The visualization should clearly represent the action from the video data, whilst maintaining temporal information. It should address the concept of event hierarchy, and emphasize the key events at each level of the hierarchy by importance. The visualization should be intuitive for the user and provide faster interpretation of events than watching the video in real-time.

As a basis for the visualization we use a 3-dimensional model of a snooker table to give clear contextual representation of the data. Obviously other event-based systems would utilize a different environment, based on the application area. The system will incorporate the tracking data for each ball object as performed in Section 5. However, due to the wealth of information that is present in the snooker video content, the challenge in this system is quickly apparent. At most, there will be 22 ball objects on the table during a match. For each shot, the cueball will move around the table which will collide with a number of other balls, causing these to move also. A typical match may have 50-60 shots played. To represent this information on a single image will result in an exceptionally confusion representation due to over-cluttering. Therefore we use a video storyboard to represent the video using a user-defined number of static visualizations. We use event prioritization and selection (from Section 4) to determine the key event sequences from the video data for the storyboard.

Table 1 gives an overview of some of the different visual cues that could be utilised in our visualization design. It is important that for each visual cue we consider how this may be constrained by the application context, and how this could be used to introduce additional information to the scene. In snooker, the obvious constraint is colour since this is already used extensively to represent different balls on the table. Since colour is so commonly adopted as a visual cue in visualization this poses an additional challenge to our work. Other cues that are restricted by the data space are the size of the ball objects and the line length of the trajectory paths. Just as with the data space, there are constraints introduced in the rendering space for representing information on the 3-dimensional snooker table. Colour, ball size and trajectory line length remain constrained due to the data space being rendered. The main constraint introduced in the rendering is lighting, since this is required to give a realistic representation of the scene. Therefore, ball and table textures, along with shadow effects, are also constrained by the rendering process. Finally, since this is a 3-dimensional model that can be viewed from any arbitrary viewpoint, the perspective viewpoint of the visualization will affect the size of

Visual Cues	Constraints due to current use in		Design options of additional event annotation to show	
	Existing Data Space	Existing Rendering Space	Event Order	Key Event
Luminance	<i>n/a</i>	<i>n/a</i>	Diffuse or specular highlight	Trajectory of key event only
Colour	Ball colours	Constrained by data	Varying trajectory colours	Varying trajectory colours
Opacity	<i>n/a</i>	<i>n/a</i>	Use to show ball motion	Use for relative shot comparison
Size	Ball size	Constrained by data	Increment size based on order	Larger ball object for key event
Shadow	<i>n/a</i>	Shadow of ball objects	Increase darkness based on order	Shadow only for key events
Line				
- length	Ball trajectory	Constrained by data	Constrained by data space	Constrained by data space
- thickness	<i>n/a</i>	Based on 3-D perspective	Increase line thickness	Key events shown thicker
Texture				
- ball	<i>n/a</i>	Ball luminance	Constrained by rendering space	Constrained by rendering space
- trajectory	<i>n/a</i>	<i>n/a</i>	Texture a proportion based on order	Use ball motion for key events
- table	<i>n/a</i>	<i>n/a</i>	Rings around cueball to show order	Text or symbol to shown key events
Annotation				
- inside frame	<i>n/a</i>	<i>n/a</i>	Text-based label (may clutter)	Text-based label (may clutter)
- outside frame	<i>n/a</i>	<i>n/a</i>	Show potted balls by pocket	Speech bubble pointing to key event

Table 1. Table to show the different possible visual cues that could be used within the visualization scheme. For each of the visual cues given we assess the constraints that are imposed by the data space (i.e. visual cues that are already used in snooker), and the constraints that are imposed in creating a realistic rendering of the scene. We then give design options based on each of the visual cues that could be used to introduce additional information to the visualization.

both ball objects and line trajectories.

Table 1 also gives possible suggestions as to how visual cues can be used to incorporate event ordering into the visualization. This is important to provide temporal relevance to the information presented in the visualization. In particular, this would indicate the ordering for a sequence of shots on the snooker table. It is clear that some visual cues will not offer significant benefit to illustrate such information, for instance, luminance. Whilst other cues could technically be employed, such as size, they are perhaps not particularly intuitive to a user whilst maintaining a clear visualization style. Finally, suggestions are given in the table for key event representation. This should introduce the notion of event importance as discussed in Section 3.3. The visualization should clearly indicate that particular shots are more significant than others. Again, some of the visual cues presented may be unsuitable for our task (e.g., colour) or may not provide intuitive representation of the key events (e.g., annotation outside the frame).

6.1 Visual Language

Fig. 7 shows the development of the visual language used for our visualization scheme, based on the initial design ideas. To illustrate this, we use an example visualization from the video storyboard, generated from real match data. The visualization represents a ‘break phase’ event from level E^4 , that consists of 6 ‘break building’ shots from level E^3 , each of which consist of events (e.g., motion) from level E^2 . As we have previously discussed, we know that ball colours, ball size and ball motion trajectories are three key elements confined by the data space and so are preserved. Fig. 7(a) gives the initial visualization for the video storyboard. In this we show each of the ball objects on the table, along with the associated ball motion as depicted by coloured lines on the table. This is done for each of the six shots. This initial design suffers from a number of flaws: it does not indicate the key event, time information is not explicit for neither a shot or the entire sequence, and the sense of action and movement that a video offers is lost from the static representation. We address these concerns in the following revisions of the design.

Firstly we replace the line trajectories with ball objects (Fig. 7(b)). Using ball objects to show the trajectory path provides a more intuitive cue due to a greater sense of ball motion. We then replace any static balls that are not directly involved in play with only the ball shadow (Fig. 7(c)). This choice was made to draw the viewer’s attention away from ball objects of low interest, whilst not removing the information entirely. Only balls that move, or balls that contribute to the state of play (i.e. in the case of a snookered position), remain in full view. To enhance the sense of action in the visualization, we use semi-transparent trajectories (Fig. 7(d)). The trajectory begins with a low alpha value that increases as the ball travels. This gives directional

information, and helps emphasize the starting position for each shot.

Event importance is introduced into the visual design at the shot level E^3 . To do this we use both opacity and trajectory width (Fig. 7(e)). Before, opacity was applied only to the ball trajectory, within the range $[0.05, 1]$. Here, the maximum opacity value is defined by the event importance, and applied to both the trajectory path and the ball object, by the formula:

$$\frac{v}{e^m}, \quad m = \frac{\alpha_{max} - \alpha}{k^2} \quad (8)$$

where α_{max} is the maximum local importance for the series of shots, α is the local importance for that particular shot and k is a user-defined constant that influences the steepness of the function. For computing the maximum opacity, $v = 1$ since this is the upper bound that the parameter can take. The formula is also used to calculate the trajectory width, where the upper bound for width is $v = 50$. From Fig. 7(e), the shot that results in the black ball being potted in the bottom right pocket has the greatest alpha value associated with it, and also is the thickest trajectory, making this the most important shot in the series.

We present two approaches for incorporating temporal information in the visual design. The first shows coloured rings on the table at the cueball start position for each shot (7(f)). The number of rings indicate the shot number, and the colour of each ring indicates the target ball colour for that shot. The second approach places icons on the cueball at each start position (7(g)), where the icon number refers to the shot number and the icon colour refers to the target ball colour. Whilst both approaches seem reasonable, the number-based icons are more explicit and so we choose this for our final design (Fig. 8(a) gives a close-up view of the icons used). By using icons we can also link this with the additional table annotations for ball potting information (discussed in Section 6.2). Fig. 7(h) presents the final visual design, whereby the key event from the series is given an emphasized shadow to make this stand out clearer from the other shots on the table.

6.2 Table Annotations

We have presented the visual language that will be used to construct the illustrations for the video storyboard. In addition to this, we also use annotation to display key information about the series of events.

Fig. 8 gives examples of the annotations used in the snooker video storyboard. Fig. 8(a) shows the icons used on the cueball. As previously discussed, the number indicates which shot this is in the keyframe and the colour indicates the object ball that the cueball collides with first. Fig. 8(b) uses the same iconic notation to show the ball pots at each pocket within the displayed sequence. Fig. 8(c) shows the ‘dashboard’ that represents the key information about the match for the displayed sequence. The red and blue bars correspond to player

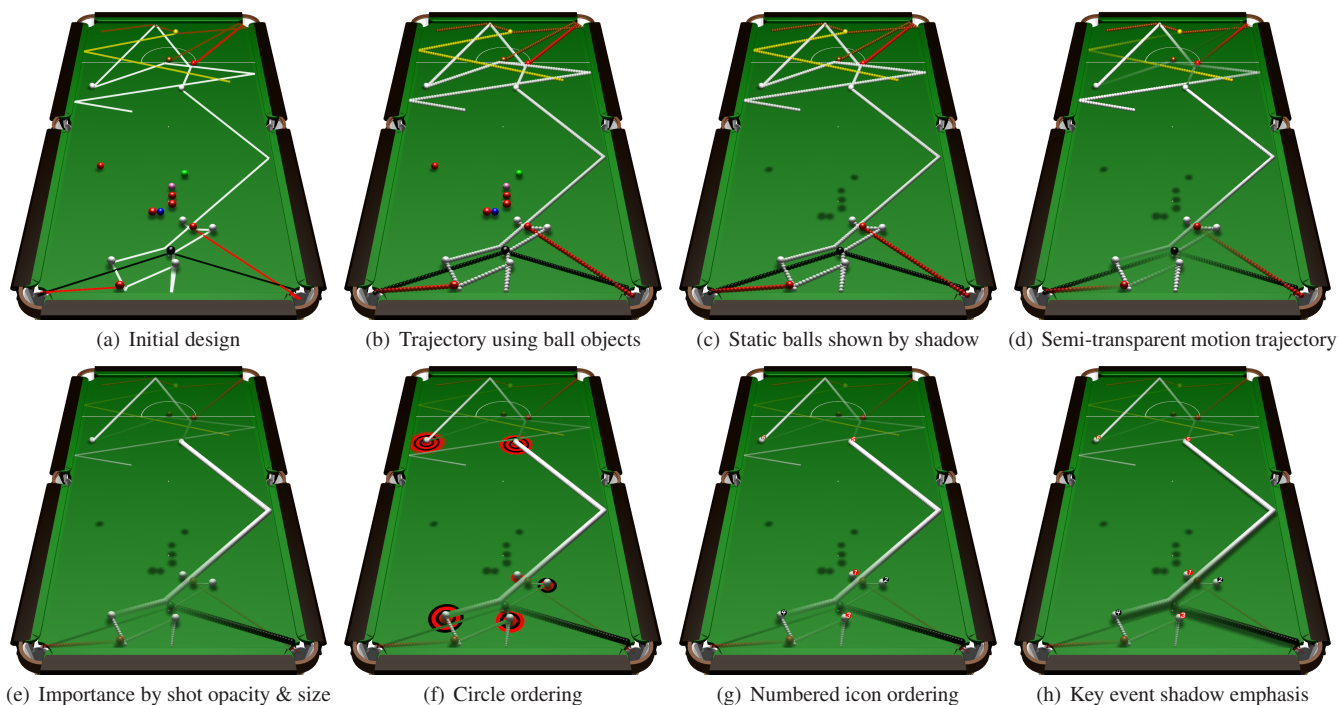


Fig. 7. Development of the visualization language used for snooker video storyboarding. Shot 4 is the most important shot in this sequence, where the black is potted in the bottom-right pocket and the cueball travels up the table. 7(a) shows the original representation with all ball objects shown and trajectory data shown by lines. 7(b) replaces the trajectory lines for each ball with ball objects. 7(c) replaces stationary balls with shadow objects. 7(d) introduces a semi-transparent trajectory to represent direction of ball motion. 7(e) incorporates shot importance, based on opacity (applied to both the ball object and the trajectory) and the trajectory width. 7(f) shows event ordering using ring notation, where the number of rings indicates shot number and the outer ring shows object ball colour. 7(g) shows event ordering using numbered icons, where the number indicates shot number and the icon colour shows object ball colour. 7(h) highlights the key event in the sequence (shot 4) using emphasized shadow.

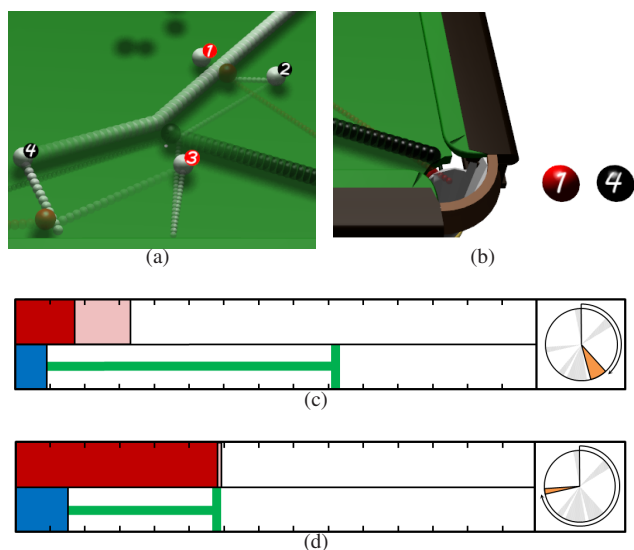


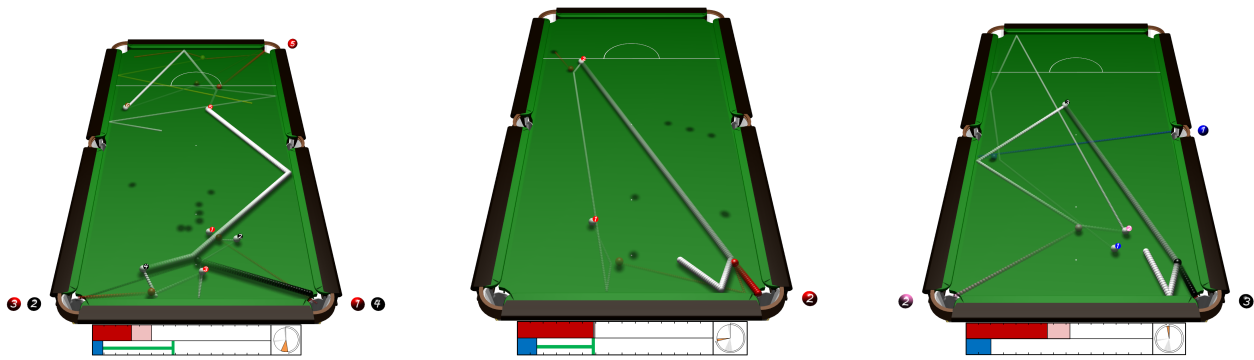
Fig. 8. Snooker visualization annotations. 8(a) shows the numbered icons used for each shot on the table. 8(b) shows the ball pot icons at the side of each pocket which correspond to the shot icons. 8(c) and 8(d) show the 'dashboard' that represents scoring and video timing information. 8(c) shows the state of play early in a match whilst 8(d) indicates a key moment, 'frame ball'.

one and two respectively, whilst the width of each bar represents the score. The solid region shows the scores prior to the displayed sequence, whilst the lighter region shows the scores as a result of the sequence. An important factor in snooker is the remaining points on the table, since if this is less than the difference between the player scores then the highest scoring player has won. A green bar is displayed on the lowest scoring player to show this value as a result of the visualized sequence. Fig. 8(d) shows where the remaining points is less than the player score difference, indicating the state 'frame ball'. Finally, the dashboard also shows timing information from the video data. The clock on the right represents the full length of the video, starting from the up-most position and moving around clockwise. The orange segment indicates the time period for the visualized sequence being shown. The gray segments indicate the other sequences used within the current video storyboard.

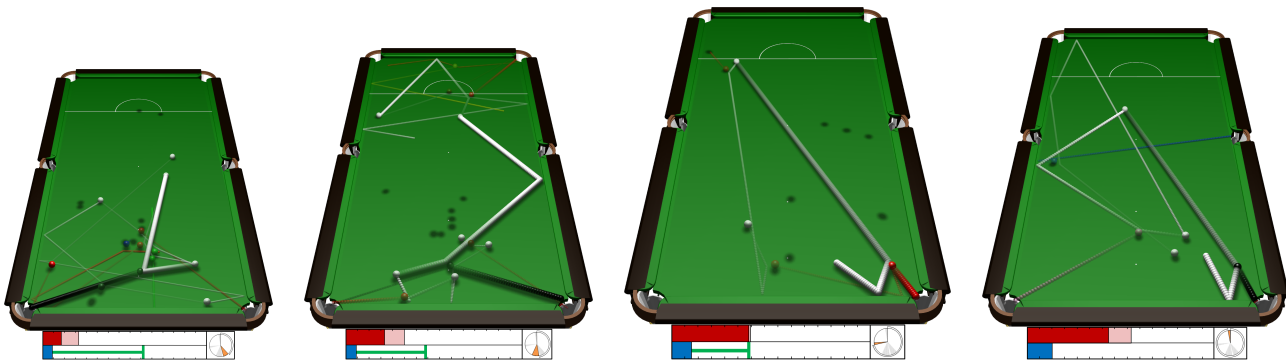
6.3 Video Storyboarding

We now generate the video storyboard for a typical game of snooker, based on the keyframe visualizations. Our example match was played in 66 shots, with each shot being recorded in 30 second segments (total length 33 minutes). From our initial case study, Fig. 4 gives the computed event importance for this match.

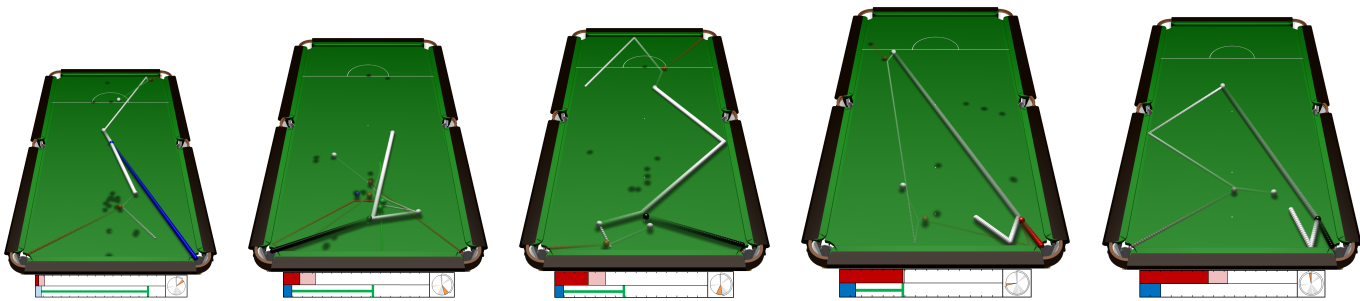
Fig. 9 presents 4 different video storyboards that are generated for the snooker match, using (a) 3, (b) 4, (c) 5 or (d) 6 illustrations. The video storyboard is depicted from left-to-right in sequential order, and maintains the event hierarchy presented in Section 4. Each illustration represents an event at level E^4 . Each shot depicted on the table represents an event at level E^3 . Each shot is made up of detected motion that represent events at level E^2 . We have discussed how importance is depicted for events at level E^3 , however by using video storyboarding we also introduce this for higher levels. The importance of E^4 events is shown by the relative size of each illustration. Frame ball has the



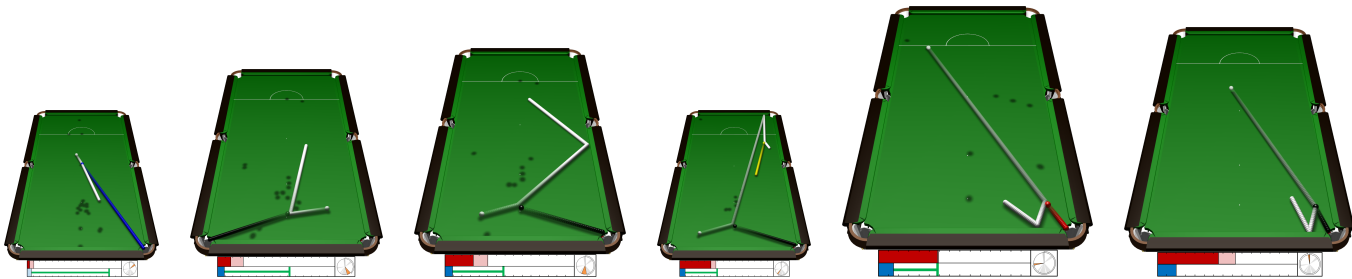
(a) Using 3 illustrations: the greatest level of detail, including all shot trajectories, shot number identifiers and corresponding pot identifiers.



(b) Using 4 illustrations: level of detail is reduced to remove outside annotation from the visualizations, however main table is preserved.



(c) Using 5 illustrations: level of detail is reduced so that less significant shots are no longer shown.



(d) Using 6 illustrations: more global events can be shown but at much lower level of detail. This results in only a single shot trajectory per visualization.

Fig. 9. Video storyboard based on hierarchical event visualization. 4 examples are shown using (a) 3, (b) 4, (c) 5, or (d) 6 illustrations (this parameter can be defined by the user). Each keyframe shows an event at level E^4 and shows importance based on relative illustration size. Each keyframe consists of a number of shots from event level E^3 . The importance of events at level E^3 is depicted through the visual design (using shot opacity and size), with the key event being depicted using emphasized shadow. Event ordering is shown using illustration ordering, trajectory paths, cueball icons, and the dashboard annotation. The number of illustrations used also impacts on the detail shown in each keyframe.

greatest importance and so the illustration appears largest in the storyboard (illustrations 2, 3, 4 and 5 for (a)-(d) respectively).

We also show how the number of illustrations can be used to impact the event importance. When 3 illustrations are used, the full visualization is shown with all shots portrayed on each illustration. As the number of illustrations increases, it may be that the level of detail required by the user becomes less. In this example, we choose to show only the key event from level E^3 when 6 illustrations are used.

7 EVALUATION

To evaluate the work in this paper, we organized two consultation sessions. In the *first consultation session*, we invited 10 participants (7 males and 3 females) with varied familiarity with the game of snooker. The feedback from this consultation session indicates clearly that the storyboard is a much more time-efficient method of understanding the most important events in the video. The full video requires 33 minutes to watch, whereas the participants required (on average) 2 minutes 44 seconds to view the storyboard. Additionally, several participants commented that viewing the video became tedious because of the large number of events that were of little interest, and ‘non-action’ moments. The feedback also indicates that the storyboard has the clear advantage over basic keyframes in helping viewers identify events. This is because each illustration in a storyboard captures several shots and depicts a series of motions and actions, while each keyframe shows a static temporal instant, from which it is difficult to infer the actual event. However, keyframes are more intuitive to depict the order of events as long as there are a sufficient number of them.

Following the initial consultation, we decided to conduct an in-depth study on the event selection algorithm by holding a *second consultation session*. This time, we invited 5 participants all with a good understanding of the game of snooker, but without any knowledge about our event selection algorithm. The goal of this consultation is to establish how close our event selection algorithm would match the expectation of the participants. After a brief introduction, the 5 participants watched the 33 minute video of the snooker match. As instructed, when watching the video, the participants paid their attention to important events of the video. They were allowed to make notes during this time. Immediately after watching the video, we asked the participants ‘what would you consider to be the most significant moments in the video?’ Participants were given a set of 66 keyframes, one per shot, with textual annotation describing the shots.

As shown in Fig. 10(a), the 5 participants selected a diverse collection of shots that are considered to be important. The participants often named a group of shots as important, and it was unavoidable to have some numbering errors (± 1). Taking these facts into account, the algorithmically determined top level event importance (i.e., the pink regions) correlates well with that suggested by the participants, except for the region of shots 47-49. In fact, shot 48 is a critical point of the match (frame ball). As the video does not show the current scores, most participants did not identify such a key event. On the other hand, our event selection process is capable of computing the numerical scores, thus taking such considerations into account.

We then provided the participants with the summary information of their assessment of the importance, highlighting the fact that the region of shots 27-42 were considered to be important by most. Some participants identified this range as two individual events that occur between 27-31 and 33-38, whilst others listed a number of shots that occur within the range 27-42 with (± 1) shot-shifting. We then asked the participants to write 5 sentences to describe that particular period of the video. We collected these writings and associated each sentence with the relevant shot numbers. The results are shown in Fig. 10(b).

The objective is to examine the usefulness of the Gaussian-moderation in selecting events for each illustration in the storyboard. Again we take into account the facts of naming groups of shots and the (± 1) shot-shifting errors. Let us focus on shots 28, 30, 34, 36 and 41. The ordinal ranking of the five events are $high[30/36, 28/34, 41]$ low . The Gaussian moderated-ranking is $high[36, 34, 41, 30, 28]$ low . The participants’ collective ranking is $high[36, 34/41, 30, 28]$ low , which is reasonably close to the Gaussian-moderated ranking.

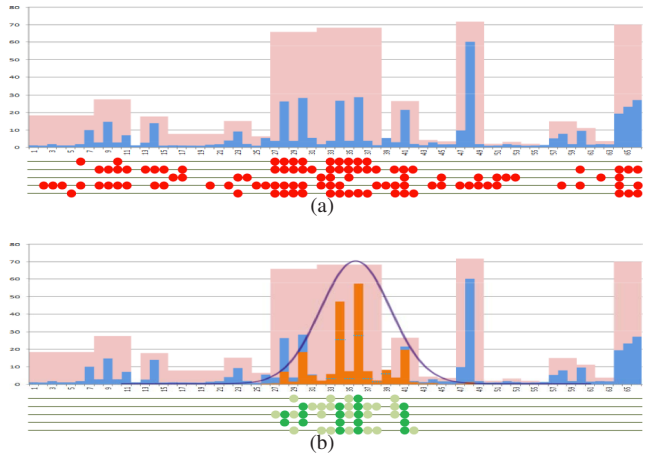


Fig. 10. Comparing the importance determined by our event selection algorithm to that of the five participants in a consultation session. (a) Each participant corresponds to a line below the bar chart of E^4 and E^3 importance, and the shot considered to be important by the participant is marked with a red circle. (b) Each participant corresponds to a line below the Gaussian-moderated importance chart. The shot considered to be important by the participant is marked with a green circle.

The second consultation session has offered a more informative evaluation about our event selection algorithms. It has shown that our top-level importance classification is consistent with the collective views of potential users, and the Gaussian moderation is highly useful in selecting events for each illustration. The two consultation sessions have also revealed that the assessment of importance varies noticeably between users. Storyboarding can potentially be used by coaches to help players make more objective and consistent assessment.

8 CONCLUSION

In this work, we have studied the use of video storyboarding to visualize events in video footage. We have presented a hierarchical framework for event organization, where higher level events are defined upon more detectable low-level events. A video storyboard, which is also organized in a hierarchical manner, consists of a set of illustrations. While each illustration corresponds to a major high-level event, it also depicts a number of events at a lower level. In the context of a sport application, we have developed a software pipeline from object detection to event classification, and from 3D reconstruction to visualization design. The most important contribution of this work is a novel method for hierarchical event selection. This method can easily be deployed in many other applications of event visualization. We have conducted two consultation sessions to evaluate our approach. The results have confirmed the usefulness of video storyboarding in general and the merits of our event selection algorithm in particular.

Through an application in snooker, we have demonstrated that a storyboard can provide an effective video visualization tool that facilitates memory externalization and reduces the needs for watching videos repeatedly. Such a tool can be used to analyse events in a game or a training session. A local snooker club has expressed that using such a video storyboard for training summarization would be greatly beneficial to players and coaches. The time required to interpret the visualization is significantly less than viewing the video footage. This approach may also be applicable to other sporting scenarios, along with other application areas, which remains the topic of future work.

ACKNOWLEDGMENTS

Many thanks to Terry Griffiths Matchroom for the use of their facilities and to the Welsh Assembly Government for funding this research. Thanks also goes to Adrian Morris for managing the industrial collaboration, and to Dave Marshall (Cardiff) and Richard Jiang (Swansea) for their involvement in the early developments prior to this work.

REFERENCES

- [1] J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: pose selection and illustration. *ACM Transactions on Graphics*, 24(3):667–676, 2005.
- [2] R. Borgo, M. Chen, B. Daubney, E. Grundy, H. Jänicke, G. Heidemann, B. Höferlin, M. Höferlin, D. Weiskopf, and X. Xie. A survey on video-based graphics and video visualization. *Eurographics (State of the Art Reports)*, Llandudno, UK, pages 1–23, 2011.
- [3] M. Chen, R. P. Botchen, R. R. Hashim, D. Weiskopf, T. Ertl, and I. M. Thornton. Visual signatures in video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1093–1100, 2006.
- [4] G. W. Daniel and M. Chen. Video visualization. In *Proc. IEEE Visualization 2003*, Seattle, WA, pages 409–416, October 2003.
- [5] H. Denman, N. Rea, and A. Kokaram. Content based analysis for video from snooker broadcasts. *Image and Video Retrieval*, 2383:198–205, 2002.
- [6] R. D. Dony, J. W. Mateer, and J. A. Robinson. Techniques for automated reverse storyboarding. *IEE Journal of Vision, Image and Signal Processing*, 152(4):425–436, 2005.
- [7] C. Everton. *Snooker and Billiards: Techniques, Tactics and Training (Crowood Sports Guides)*. The Crowood Press, 1991.
- [8] P. Gatalsky, N. Andrienko, and G. Andrienko. Iterative analysis of event data using space-time cube. In *8th International Conference on Information Visualisation*, pages 145–152, 2004.
- [9] D. B. Goldman, B. Curlless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics*, 25(3):862–871, 2006.
- [10] H. Guo and B. M. Namee. Using computer vision to create a 3d representation of a snooker table for televised competition broadcasting. In *Proceedings of the 18th Irish Conference on Artificial Intelligence and Cognitive Science*, pages 220–229, 2007.
- [11] M. Höferlin, E. Grundy, R. Borgo, D. Weiskopf, M. Chen, I. W. Griffiths, and W. Griffiths. Video visualization for snooker skill training. *Computer Graphics Forum*, 29(3):1053–1062, 2010.
- [12] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE transactions on systems, man and cybernetics - Part C: Applications and reviews*, 34(3):334–352, 2004.
- [13] T. Kapler and W. Wright. Geotime information visualization. In *INFOVIS '04 Proceedings of the IEEE symposium on Information Visualization*, pages 25–32, 2004.
- [14] D. E. Knuth, J. H. Morris, and V. R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.
- [15] P. A. Legg, M. L. Parry, D. H. S. Chung, R. M. Jiang, A. Morris, I. W. Griffiths, D. Marshall, and M. Chen. Intelligent filtering by semantic importance for single-view 3D reconstruction from snooker video. *To appear in 18th International Conference on Image Processing (ICIP), Brussel, Belgium*, 2011.
- [16] B. Li and M. I. Sezan. Event detection and summarization in sports video. In *Content-Based Access of Image and Video Libraries, 2001. (CBAIVL 2001). IEEE Workshop on*, pages 132–138, 2001.
- [17] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):873–889, 2001.
- [18] H. Pan, P. van Beek, and M. I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages 1649–1652, 2001.
- [19] N. Rea, R. Dahyot, and A. Kokaram. Modelling high level structure in sports with motion driven HMMS. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, pages 621–624, 2004.
- [20] M. Romero, J. Summet, J. Stasko, and G. Abowd. Viz-a-vis: Towards visualizing video through computer vision. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1261–1268, 2008.
- [21] D. A. Sadlier and N. E. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(10):1225–1233, 2005.
- [22] W. Shen and L. Wu. A method of billiard objects detection based on snooker game video. In *International Conference on Future Computer and Communication (ICFCC)*, 2010.
- [23] C. S. Smith. Activities: States or events? *Linguistics and Philosophy*, 22(5):479–508, 1999.
- [24] Y. Takahashi, N. Nitta, and N. Babaguchi. Video summarization for large sports video archives. In *IEEE International Conference on Multimedia and Expo*, pages 1170–1173, 2005.
- [25] Y. Wang, D. M. Krum, E. M. Coelho, and D. A. Bowman. Contextualized videos: Combining videos with environment models to support situational understanding. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1568–1575, 2007.
- [26] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2000.
- [27] B.-L. Yeo and M. Yeung. Retrieving and visualizing video. *Communications of the ACM*, 40(12):43–52, 1997.
- [28] L. Yu, A. Lu, W. Ribarsky, and W. Chen. Automatic animation for time-varying data visualization. *Computer Graphics Forum*, 29(7):2271–2280, 2010.
- [29] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages 123–130, 2001.