

Method and System for Event Detection in a Video Stream

FIELD OF INVENTION

This invention relates to a method and system for event detection in a video stream.

BACKGROUND OF INVENTION

The event broadcasting industry is becoming one of the most profiting businesses. For example, the amount of sports video content is increasing rapidly due to the large number of audience. However, the time available for audience to watch the events such as sports video is decreasing. Thus, there is a demand for extracting highlights from e.g. sports videos to facilitate the audience to watch only the exciting or interesting parts of the sports video. Events highlights are also demanded by broadcasters e.g. for presentation during breaks of an event or for compiling an event summary for news breaks.

Currently, generating highlights from a live event is a labor-intensive and inflexible process. For example, in analyzing a sports video using only Audio/Visual (A/V) features it is difficult to identify the semantics of an event such as the scoring of a soccer goal. For example, in a soccer video, although the "goal" event can be detected using A/V features through analyzing the video frames, the details such as who scored the goal cannot be evaluated using A/V features easily. Secondly, some events are almost impossible to be detected using only A/V features, e.g. "yellow/red cards" events. Therefore, good sports video highlight generation requires much skill and experience and is a difficult process. The low-level A/V feature based bottom-up approaches tend to capture the syntactic information rather than rich user semantics. The view composition from multiple cameras tries to help viewers reconstruct activities but often misses important visual clues to infer events. Furthermore, the assessment of the quality of sports highlights is a strong subjective task.

A need therefore exists to provide a method and system for live event detection in video that addresses at least one of the above-mentioned problems.

SUMMARY OF INVENTION

In accordance with one aspect of the present invention, there is provided a method of event detection in a video stream, comprising the steps of: detecting a text event in a casting text stream external to the video; and identifying a corresponding video event in the video stream based on the detected text event.

The step of identifying of the corresponding video event may comprise time aligning the casting text stream with the video stream.

The step of time aligning may comprise: detecting a time stamp in the casting text stream associated with the text event; detecting a digital clock region in frames of the video stream; extracting time information from the digital clock region; and aligning the text event with video event based on the time stamp and the extracted time information.

The step of extracting the time information from the digital clock region may comprise: utilising periodic pattern change detection to identify digit regions within the digital clock region; collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and recognising the digit characters displayed in the digital clock region utilising the pattern templates.

The step of extracting the time information from the digital clock region may comprise: utilising periodic pattern change detection to identify digit regions within the digital clock region; identifying a first change in a pattern of one of the digit regions; and extracting the time information based the identified first change in the pattern of the one digit region.

The identifying of the corresponding video event may comprise defining a search range for identifying the video event in the video stream aligned with the casting text stream, the search range being around the temporal location of the time stamp in the aligned casting text stream and video stream.

The method may further comprise utilising a model with a "noise-event-noise" structure for identifying the video event within the search range.

The "noise-event-noise" structure may comprise a self-jump structure.

In accordance with another aspect of the present invention, there is provided a system for event detection in a video stream, the system comprising: a detector for detecting a text event in a casting text stream external to the video; and an identification unit for identifying a corresponding video event in the video stream based on the detected text event.

The identification unit may identify the corresponding video event in the video stream based on the detected text event by time aligning the casting text stream with the video stream.

The identification unit may comprise: a detection unit for detecting a time stamp in the casting text stream associated with the text event and detecting a digital clock region in frames of the video stream; an extractor for extracting time information from the digital clock region; and a time aligning unit for aligning the text event with video event based on the time stamp and the extracted time information.

The extractor unit may extract time information from the digital clock region utilising periodic pattern change detection to identify digit regions within the digital clock region; collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and recognising the digit characters displayed in the digital clock region utilising the pattern templates.

The extractor unit may extract time information from the digital clock region utilising periodic pattern change detection to identify digit regions within the digital clock region; identifying a first change in a pattern of one of the digit regions; and extracting the time information based the identified first change in the pattern of the one digit region.

The identification unit may identify the corresponding video event in the video stream based on the detected text event by defining a search range for identifying the video

event in the video stream aligned with the casting text stream, the search range being around the temporal location of the time stamp in the aligned casting text stream and video stream.

The identification unit may utilise a model with a "noise-event-noise" structure for identifying the video event within the search range.

The identification unit may utilise a model with a "noise-event-noise" structure comprising of a self-jump structure.

In accordance with yet another aspect of the present invention, there is provided a method of extracting time information from a video stream, the method comprising: detecting a digital clock region in frames of the video stream; and utilising periodic pattern change detection to identify digit regions within the digital clock region.

The method may comprise detecting the digital clock region in the frames of the video stream; and utilising the periodic pattern change detection to identify the digit regions within the digital clock region; collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and recognising the digit characters displayed in the digital clock region utilising the pattern templates.

The method may comprise: detecting the digital clock region in the frames of the video stream; utilising the periodic pattern change detection to identify the digit regions within the digital clock region; identifying a first change in a pattern of one of the digit regions; and extracting the time information based the identified first change in the pattern of the one digit region.

In accordance with a further aspect of the present invention, there is provided a system of extracting time information from a video stream, the system comprising a detection unit for detecting a digital clock region in frames of the video stream by utilising periodic pattern change detection to identify digit regions within the digital clock region.

The detection unit may detect the digital clock region in frames of the video stream by utilising the periodic pattern change detection to identify the digit regions within the digital clock region; collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and recognising the digit characters displayed in the digital clock region utilising the pattern templates.

The detection unit may detect the digital clock region in frames of the video stream by utilising the periodic pattern change detection to identify the digit regions within the digital clock region; identifying a first change in a pattern of one of the digit regions; and extracting the time information based the identified first change in the pattern of the one digit region.

In accordance with another aspect of the present invention, there is provided a data storage medium having stored thereon code means for instructing a computer system to execute a method of event detection in a video stream, comprising the steps of: detecting a text event in a casting text stream external to the video; and identifying a corresponding video event in the video stream based on the detected text event.

In accordance with yet another aspect of the present invention, there is provided a data storage medium having stored thereon code means for instructing a computer system to execute a method of extracting time information from a video stream, the method comprising: detecting a digital clock region in frames of the video stream; utilising periodic pattern change detection to identify digit regions within the digital clock region.

BRIEF DESCRIPTIONS OF THE DRAWINGS

These and other features, objects and advantages of the present invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, used in conjunction with the attached drawings, in which:

Figure 1 illustrates a conceptual block diagram of event detection from a live video.

Figure 2 illustrates an example of text broadcasting.

Figure 3 illustrates the keywords definition for soccer events.

Figure 4 illustrates examples of A/V features and the associated analysis.

Figure 5 illustrates a far-view shot and close-up shot in a live video.

Figure 6 illustrates a digital clock with other text contents overlaid in the live video.

Figure 7 shows a flow chart illustrating two methods of digital clock detection and game time recognition.

Figure 8 illustrates the periodic pattern change of a SECOND digit character of the digital clock.

Figure 9 illustrates M cycles of the SECOND digit character of the digital clock.

Figure 10 illustrates a Temporal Neighbouring Pattern Sequence (TNPS) in a region of a video image without the overlaid digital clock and a TNPS in a region of a video image with the overlaid digital clock.

Figure 11 illustrates possible recognition results of digit character numerals in consecutive 25 video frames.

Figure 12 illustrates clock digit recognition accuracy of a video digital clock time method.

Figure 13 illustrates a TNPS of the MINUTE digit character of the digital clock.

Figure 14 illustrates clock digit recognition accuracy of a video digital clock time method based on temporal periodic pattern change of digit characters.

Figure 15 illustrates the time relationship of A/V stream and text stream.

Figure 16 illustrates the Hidden Markov Model (HMM) grammars for alignment of text event and A/V features.

Figure 17 illustrates the probability score combination scheme.

Figure 18 illustrates a probability scores graph for a "card" event.

Figure 19 illustrates a schematic drawing of a computer system for implementing the method and system described.

Figure 20 is a flow diagram of a method for video indexing.

Figure 21 is a flow diagram of a method for personalised video generation.

Figure 22 is a schematic diagram of a system for indexing video and generating personalised video.

Figure 23 is a schematic diagram of the indexing and classification process.

Figure 24 is a flow diagram of "play keywords" extraction from Sports Web casting Text.

Figure 25 is a flow diagram of "play keywords" extraction from action description tokens.

Figure 26 is a table of an example of Sports Web casting Text.

Figure 27 is a table including a sample of Sports Web casting Text for a goal event.

Figure 28 is a flow diagram of parsing input video stream into play, replay and break video segments, and commercials.

Figure 29 is a flow diagram of "video keywords" extraction from the play/replay/break video segments.

Figure 30 is a flow diagram of "audio keywords" extraction.

Figure 31 is a flow diagram of a method for automatic video summary creation from user preferences.

Figure 32 is a flow diagram of a method for automatic video summary creation from a text summary.

Figure 33 is a flow diagram of replay video segment detection, parsing and classification.

Figure 34 is a flow diagram of a method of learning weighting for different types of replays from human production directors.

Figure 35 is a flow diagram of an algorithm for the soccer or football ball detection.

Figure 36 is a flow diagram of an algorithm of the real time detection of the goalmouth.

Figure 37 is a diagram of the three streams of footage annotated with semantic content.

Figure 38 is a diagram of the three streams of footage annotated with semantic content to create a personalized video summary.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "detecting", "identifying", "calculating", "generating", or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a conventional general purpose computer will appear from the description below.

Figure 1 is a block diagram illustrating the steps for aligning text describing a video event to the time of occurrence of the video event according to an example embodiment of the present invention. There are two concurrent input sources relating to the same video event, a live broadcast sports video 102 and a live web-casting text 104. In the example embodiment, the input sports video 102 has a digital clock overlaid in the video. The digital clock typically indicates the amount of time passed in the sports game. In the example embodiment, the input sports video 102 in MPEG format is a broadcast soccer game video. It is appreciated that the input sports video 102 may be extended to other sports or event domains.

At step 108, keywords are defined. The keywords are defined by selecting words of the input web-casting text 104 and associating the words with predefined events of interest. For example, keywords such as "red card" describe the event that a soccer player is sent off the field.

Once keywords are defined at step 108, the occurrence of an event of interest can be detected by matching the words in the text description of the input web-casting text 104 with the defined keywords at step 110. An event detected from the input web-casting text 104 is hereinafter referred to as a text event.

At step 112, the semantics of the text event are extracted. For instance, a semantic such as the name of the person who scored a soccer goal is extracted.

At step 114, the time tag associated with the input web-casting text 104 is extracted. The time tag contains the time occurrence of the event of interest that is described by the input web-casting text 104, for example, the time tag may refer to the specific time of scoring a goal in a football game.

With reference to the input sports video 102, A/V features such as shot boundary, semantic shot class, replay, motion, and audio keywords are extracted at step 116, for video event detection. Details of A/V feature extraction will be described later.

Simultaneously with step 116, the digital clock overlaid in the sports video 102 is detected at step 118, and the game time is recognized at 120. Details of digital clock detection and game time recognition will be described later.

At step 122, the text events and video events are synchronized to obtain the starting and ending boundaries of a final video event product based on the input text 104 and sports video 102. Details of the aligning at step 122 will be described later.

The example embodiment advantageously uses text describing a sports video event to detect the semantics of the event. Text analysis greatly increases the video event detection performance due to the precise information available in the text description based on the web-casting text 104. Furthermore, text analysis can extract semantics such as the player/team involved in the sports video event.

In the sports domain, textual information of a sports event is available as caption text, Automatic Speech Recognition (ASR), Closed Caption (CC), game logs, and Text Web Broadcasting (TWB). Compared with caption text and ASR, which are heavily affected by video quality, or CC and game logs which are not easily obtainable, information on sports events is more freely available from TWB e.g. in the form of live text commentaries or match reports that are published on the Internet. In addition, TWB possesses very detailed information about the event such as the players involved in the sports event and the approximated time of the sports event.

Figure 2 shows an example text event entry 202 of a TWB sports commentary 200 on the Internet, which contains detailed information of a soccer match. The text event entry 202

indicates that a goal has been scored. The goal scorer is Fabio Cannavaro and the time of scoring is 62 minutes and 28 seconds into the game. The updated scoreboard is 2 goals to Liverpool and 1 goal to Juventus.

Moreover, TWB is a broadly used text broadcasting means for sports games and can be easily accessed (in HTML format) from many sports web sites such as ESPN, BBC, etc. In some websites, the accuracy of the time-stamp in TWB can be as good as about 30 seconds off from the actual event happening. Hence, the example embodiment uses TWB as the source of textual information.

Returning to Figure 1, in the example embodiment, at the step of keyword definition 108 in Figure 1, each type of event features one or several unique nouns. For example, "Yellow card" and "Red card" for a "card" event in a soccer game. These nouns are defined as keywords and are used to detect keywords from TWB to identify a relevant event. To improve accuracy of event detection performance, the following conditions are preferably satisfied in the the example embodiment.

Firstly, keywords might have different appearances. For example, "g-o-a-l", "goooooaaaal", etc are forms of expressing the word "goal". Hence, in the example embodiment, during keyword search, stemming, phonic and fuzzy technologies are employed to determine whether the actual keyword has appeared in the TWB.

Secondly, when keywords are shown in phrases, there may be different meanings. For example, the meaning of "goal kick" is far from "goal". Hence phrases containing keywords can have different meanings. Both the keyword and the accompanying words occurring immediately before or after the keyword are searched and deciphered in the example embodiment to verify that the phrase containing the keyword does not change the meaning of the keyword.

Thirdly, the presence of a keyword in the TWB does not guarantee the occurrence of the related event described by the keyword. The occurrence of the keyword may be a false alarm of the related event. In the example embodiment, both the keyword and the accompanying verbs, adverbs, adjectives or the like are searched and deciphered to verify that the occurrence is not a false alarm.

Figure 3 illustrates a table 300 containing keyword definitions for soccer events in the example embodiment.

One of the events is goal 302, which indicates scoring a goal. Keywords and phrases associated with the event "goal 302" are "g-o-a-l", "scores", "goal", "equalize", "kick" or similar variations.

A second event is card 304, which indicates a player getting a yellow card or red card. Keywords and phrases associated with the event "card 304" 320 are "yellow card", "red card", "yellowcard", "redcard", "yellow-card", "red-card" or similar variations.

A third event is foul 306, which indicates a player committing a foul. Keywords and phrases associated with the event "foul 306" are "commits foul", "foul", "booked", "ruled", "yellow" or similar variations.

A fourth event is offside 308, which indicates the referee calling for offside. Keywords and phrases associated with the event "offside 308" are "flag offside", "adjudge offside" "rule offside", "offside", "off side", "off-side" or similar variations.

A fifth event is save 310, which indicates the saving of the ball from being scored or from danger to score. Keywords and phrases associated with the event "save 310" are "make same", "produce save", "bring save", "dash save", "pull off save" or similar variations.

A sixth event is injury 312, which indicates a player is injured in the game. Keywords associated with the event "injury 312" is "injury", "hurt", "pain" or similar variations. The phrase "injury time" and its variations are not included as defined keywords for the event "injury 312" in the example embodiment as it relates to additional time to continue with the game after game ending time.

A seventh event is free-kick 314, which indicates that a free-kick is taking place. Keywords associated with the event "free-kick 314" are "take free-kick", "save free-kick", "concede free-kick", "deliver free-kick", "fire free-kick", "curl free-kick", "free-kick", "free kick", "freekick" or similar variations.

An eighth event is substitution 316, which indicates the substitution of a player. Keywords and phrases associated with the event "substitution 316" are "substitution", "replacement of player" or similar variations.

The soccer events are selected in the example embodiment because these events are important to a soccer match and are difficult to be detected by traditional A/V analysis techniques. However, it is appreciated that different keywords and events can be defined as appropriate. Once proper keywords are defined, a text event corresponding to one of the soccer events can be detected by finding sentences that contain the relevant keywords. Text event detection is achieved at step 110 of Figure 1.

Through text events, semantics such as the players involved in a particular text event may be identified and extracted. Event semantic extraction occurs at step 112 of Figure 1. In the example embodiment, during the initialization phase, a player/team database made up of the team name and players' names is built by analyzing the start-up line information from the TWB of the live soccer game for the purpose of extracting the players involved in an event. During semantic extraction, when a text event is recognized through the matching of keywords, every word in the text event are matched with the name entries in the player/team database to extract all the names of players and teams relevant to the text event.

Broadcast soccer game video is used in the example embodiment for A/V analysis (116) because broadcast soccer game video contains rich post-production information that is favourable for video summary, and furthermore, the post-production information provides robust A/V features. However, it is appreciated that other types of videos can also be used. Examples of the A/V features used in the example embodiment are listed in table 400 shown in Figure 4. The associated A/V feature description, type of analysis and method of extracting the features are described below.

One A/V feature for visual analysis is Shot boundary detection (F_1) 402. In broadcast video, the shot or duration of an event can be regarded as a basic analysis unit. An initial shot boundary detection (SBD) may be performed using software such as M2-Edit Pro™ or the like. The detection result is then manually refined by deleting any obvious and

unnecessary boundaries to combine the neighboring shots. The obtained shot boundary feature is denoted with identification (ID) F_1 , which is a sequence of video frame numbers. Each number indexes a boundary between two successive shots.

A second A/V feature for visual analysis is Semantic shot classification (F_2) 404. The shots transition in soccer video reveals the state of the game, hence the semantic shot classification provides a robust feature for soccer video analysis. The algorithm for shot classification is well known in the art that can be performed in a variety of ways. For example, a unified framework for semantic shot classification in sports video as described by Lingyu Duan, et. al, Storage and Retrieval for Media Database, SPIE 2003, may be used. The shots are classified into 5 semantic classes: far view, in-field medium view, in-field close-up view, out-field medium view and out-field close-up view. The generated shot classification feature is denoted as ID F_2 . F_2 is a sequence with each element indicating the shot class label of the corresponding video frame. Figure 5 illustrates examples of far-view shots 502, 504 that are a “zoom out” shots of the soccer field and close-up shots 506, 508 that are a “zoom in” shots of the soccer field in the broadcast soccer game video.

In Figure 4, a third A/V feature for visual analysis is Replay detection (F_3) 406. In a typical broadcast soccer game the director would normally launch a replay for interesting events. Hence, replay detection greatly facilitates the soccer video analysis. For example, replays can be detected using flying-logo template matching techniques in Red (R), Green (G) and Blue (B) channels. The replay/non-replay state of each frame is denoted by value 1 and 0 respectively and is collected as a sequence in the feature denoted as ID F_3 .

A fourth A/V feature for visual analysis is Camera motion (F_4) 408. As the camera typically follows the movement of the player, the camera motion provides a useful cue to represent the activity of the game. In the example embodiment, six camera motion features are extracted for each frame, specifically “average motion magnitude”, “motion entropy”, “dominant motion direction”, “camera pan factor”, “camera tilt factor” and

“camera zoom factor”. The generated camera motion feature is denoted as ID F_4 , which is an R^6 vector sequence.

A fifth A/V feature for audio analysis is Audio keyword (F_5) 410. There are some significant game-specific sounds that have strong relationships to the action of players, referees, commentators and audience in sports videos. Hence the creation of suitable audio keywords can help the high-level semantic analysis. Machine learning (e.g. Support Vector Machine) is used to train audio features (e.g. Mel Frequency Cepstral Coefficients and Liner Prediction Coefficient Cepstral features) to generate audio keywords in sports video. For example, three audio keywords can be created for soccer audio, namely “whistle”, “acclaim” and “noise”. The generated audio keyword is denoted as ID F_5 . F_5 is a sequence with each element indicating the audio keyword label of the corresponding frame.

Video clock detection and game time recognition will now be described. These actions correspond with steps 118 and 120 respectively in Figure 1. With reference to Figure 6, a digital clock 600 is overlaid in the soccer game video of the example embodiment. Typically, the digital clock 600 is placed together with other text contents such as abbreviated player team names 602, 604 and current score 606, 608. A traditional video text recognition method that can be used to extract the game time from the video is the Optical Character Recognition (OCR) method. In the OCR method, the digit characters 612 on the digital clock 600 are recognized in the same way as the OCR method is used to recognize alphabetical text characters. However, the performance of the OCR method in detecting digit characters 612 in can be limited due to the typically small size of the digit characters 612 and low resolution of the soccer game video. The speed of the recognition process is also typically slow because of the complexity of the algorithm used in the OCR method, furthermore, the OCR method involves training the algorithm to recognize characters.

In the example embodiment, a special characteristic of the digital clock 600 in the soccer game video is used to detect the digit characters 612. Utilizing the special characteristic, the digit characters 612 of the digital clock 600 can be located without first having to recognize all the characters found in static regions of the soccer game video. The special

characteristic of the digit characters 612 is the periodic pattern change of the digit character image due to the natural transition cycle of time. Using this special characteristic, the positions of digit characters 612 within a static region are located reliably and accurately with less computing power, and hence it is suitable for use in real time applications.

With reference to Figure 7, at step 702, static regions overlaid in the soccer game video are detected. The digital clock 600 will be in one of these static regions. By locating these static regions, the searching region for the digital clock 600 is narrowed. The static regions in the video are detected by calculating the edges for each consecutive image frame decoded from the video and evaluating the temporal changes of the edge frames. The regions with minimal temporal changes are the static regions. After locating the static regions, each consecutive image frame of the video corresponding to the static regions is put through binarization at step 704.

At step 706, Connected Component Analysis is utilized to segment the binarized video image to get a series of connected Regions Of Interest (ROI), which may potentially contain the digit characters 612.

At step 708, the candidate ROIs are further filtered and reduced based on domain knowledge such as according to size, height and width ratio, distance between multiple ROIs and their vertical position so that noise and other irrelevant symbols in the video image are not included. The digit characters 612 of the digital clock 600 will be contained in some of the remaining candidate ROIs.

At step 710, a periodic pattern change detection procedure is applied to the candidate ROIs to locate and identify the digit characters. In the example embodiment, the SECOND digit character 614 (Figure 6) is selected to be the first character to be located as its period is the shortest and therefore can be located the fastest.

The periodic pattern change detection procedure involves observing pattern changes by applying a similarity measure $S(n)$ on a candidate ROI in two consecutive video frames. $S(n)$ is given by equation (1).

$$S(n) = \sum_{(x,y) \in R} f_{n-1}(x, y) \otimes f_n(x, y) \quad (1)$$

where n is the frame sequence number, $f(x, y)$ is the pixel value at position (x, y) , R is the segmented region of the digit character. x and y are position values along two orthogonal axes of the video frame. The symbol $\otimes$ means that $a \otimes b$ is 0 if a and b are the same value, otherwise it is 1.

When the frames are binarized, the formulae of $S(n)$ becomes equation (2):

$$S(n) = \sum_{(x,y) \in R} B_{n-1}(x, y) \otimes B_n(x, y) \quad (2)$$

where $B(x, y)$ is the binarized image pixel value at position (x, y) and $B(x, y)$ is given by equation (3):

$$B_n(x, y) = \begin{cases} 1, & \text{if } f_n(x, y) \geq \text{threshold} \\ 0, & \text{if } f_n(x, y) < \text{threshold} \end{cases} \quad (3)$$

Pattern changes due to changes in character numerals according to time (if any) of the candidate ROI can be observed in a graph plot of Temporal Neighbouring Pattern Similarity (TNPS) Sequence, that is, $S(n)$ versus video frame sequence number, n . If the pattern change is regular and periodic, the candidate ROI under test is concluded to be a digit character of the digital clock. If the pattern change happens once every second, the digit character is a SECOND digit character 614 (Figure 6). If the pattern change happens once every ten seconds, the digit character is a TEN-SECONDS digit character 618 (Figure 6). If the pattern change happens every one minute, the digit character is a MINUTE digit character 616 (Figure 6). If the pattern change happens every ten minutes, the digit character is a TEN-MINUTES digit character 610 (Figure 6).

Figure 8 shows a graph of the TNPS Sequence for the SECOND digit character 614 (Figure 6), with $S(n)$ 802 versus the video frame sequence number, n 804. A periodic drop-down 806 with a significant lower $S(n)$ value can be seen occurring every second. The periodic drop-down 806 indicates that a pattern change is taking place due to the transition of the SECOND digit character 614 (Figure 6) from one numeral to another at

each second. In the example embodiment, the SECOND digit character 614 (Figure 6) based on a pattern change in the SECOND digit character 614 (Figure 6) is detected by calculating the value of $S(n)$ for each consecutive video frame. Where the $S(n)$ is at a minimum value, it can be concluded that a pattern change has occurred to the SECOND digit character 614 (Figure 6).

An algorithm is utilized in the example embodiment to calculate the minimum value of $S(n)$ and determine if the candidate ROI is a clock digit character. As the number of video frames of the video occurring in one second in various video system formats are fixed, the algorithm uses this fact to determine whether the consecutive calculated minimum values of $S(n)$ of a series of video frames of a video have equal period. For instance, every second is about 25 or 29.97 frames of video image depending on whether the video system format is Phase Alternating Line (PAL) or National Television System Committee (NTSC) respectively. Assuming PAL system, if M consecutive minimum values of $S(n)$ are about 25 video frames apart, which is equivalent of one second, the first conclusion is that the digital clock 600 (Figure 6) is present and the second conclusion is that the candidate ROI under test is the SECOND Digit character 614 (Figure 6).

It is typical in a soccer game that the digital clock may disappear occasionally and the position of the digital clock may appear at another location in the video. Figure 9 illustrates M cycles of the TNPS sequence of the SECOND digit character 614 (Figure 6) when the digital clock starts to appear in the video. The frame positions of the changes at each second of the SECOND digit character 614 (Figure 6) in graph 900 are indicated as N_0 902, N_1 904, ..., N_M 906. Before the N_0 902 position, no periodic minimum value can be observed, indicating that the digital clock has not appeared yet.

With reference to Figure 10, the TNPS sequence of the SECOND digit character 614 (Figure 6) has smaller variation except at the time when the digit character changes. The small variations are obvious when comparing the TNPS sequence in the region without a digital clock as shown in graphs 1000 and 1002 of Figure 10A to the TNPS sequence in the region with the SECOND digit character 614 (Figure 6) as shown in graphs 1004 and 1006 of Figure 10B.

In the example embodiment, the algorithm includes a further step of verifying that the periodic pattern changes or periodic minimum values $S(n)$ belong to the SECOND digit character 614 (Figure 6) and not other unimportant pattern changes by calculating the variance of $S(n)$ at all frame positions but excluding the frame positions with minimum values. For instance, with reference to Figure 9, a value E is calculated based on the variance of $S(n)$ excluding frame positions at N_0 902, N_1 904, ..., N_M 906, E is given by:

$$E = \sum_{\substack{n=N_0+1 \\ n \neq N_1, \dots, N_M-1}}^{N_M-1} (S(n) - \bar{S})^2 \quad (4)$$

where

$$\bar{S} = \frac{1}{N_M - N_0 - M} \sum_{\substack{n=N_0+1 \\ n \neq N_1, \dots, N_M-1}}^{N_M-1} S(n) \quad (5)$$

To test whether the tested candidate ROI truly contains the SECOND digit character 614 (Figure 6), E values of the tested candidate ROI are compared with a threshold value. The maximum E value of a set of E values calculated based on sample clock digit characters is set as the threshold value. If the calculated variance E value of a tested candidate ROI is smaller than the threshold value, it means that the image pattern does not change much except at the periodic positions N_0 902, N_1 904, ..., N_M 906, hence, the tested candidate ROI can be considered to contain the SECOND digit character 614 (Figure 6). For other random changing patterns that are "false" signals but may happen to have similar periodic minimum values, the value of E is likely to be beyond the threshold value.

Returning to Figure 7, after the SECOND digit character 614 (Figure 6) is located and verified, all the other clock digit characters can be located at step 712. The other digit characters are located according to their relative position to the SECOND digit character 614 (Figure 6).

After all the digit character positions are located at step 712, the next step is to derive the clock or game time from the digit characters of the digital clock in the video. This can be achieved using two different methods in different example embodiments.

The first method determines the game time by detecting pattern changes of the digit characters and recognizing the clock digit characters in the clock. The second method determines the game time by observing the clock digit characters and inferring the time for the rest of the game through the natural transition cycle of time without involving recognizing the numeric characters in the digital clock.

The first method to derive the clock or game time from the digital clock will now be described.

Continuing from step 712, sample patterns of digit characters of the digital clock are collected at step 718. The natural transition pattern of the digit characters in the digital clock is such that whenever the TEN-SECOND digit character changes, the SECOND digit character 614 (Figure 6) will start counting from "0". Hence, when a pattern change of the TEN-SECOND digit character 618 (Figure 6) is detected as the video is playing, the pattern of the SECOND digit character 614 (Figure 6) is understood to turn "0" followed by "1", "2", ... and so on. Each numeric character, "0" to "9", of the SECOND digit character are then collected at each transition as sample patterns of digit characters of the digital clock. The pattern change of the TEN-SECOND digit character 618 (Figure 6) is detected by monitoring one cycle of its TNPS based on calculations according to equation (2). The pattern change position is detected at the frame sequence number when the TNPS reaches a minimum value. In this way, the pattern change is detected more reliably without a heuristic threshold. In the example embodiment, which uses PAL system, 25 samples of each numeric character "0" to "9" in the 10 seconds time period are gathered at the frame rate of 25 and all of which are used as templates for a later step of recognizing digit characters. The number of templates can be reduced by combining similar samples as one sample to minimize calculation time.

At step 720, the numeric value of the clock digit characters are recognized based on the templates collected. After the templates for digit characters "0" to "9" are collected at step 718, every frame of decoded images of the video are matched against the templates. The matching score of a numeric character i is calculated by equation (6).

$$S(i) = \text{Max}_j \left\{ \sum_{(x,y) \in I} T_{ij}(x,y) \otimes D(x,y) \right\}, \quad i = 0, 1, \dots, 9, 10 \quad (6)$$

where $T_{ij}(x, y)$ is the binarized image pixel value in position (x, y) for the j^{th} template of numeric character i . $D(x, y)$ is the binarized image pixel value in position (x, y) for the digit character to be recognized. I is the ROI of the digit character. When $i = 10$, $T_{10j}(x, y)$ is the template for a flat candidate ROI without any character.

To recognize a digit character in the clock, the matching scores for the 25 (frame rate) frames are calculated. The possible 25 recognition results may either comprise the same numeric character or two consecutive numeric characters. In particular, in the first case when the numeric character is the same, the result is x for all the 25 frames, where x is a numeric character from "0" to "9" or a space. In the second case when there are two consecutive numeric characters, the result for a first series of consecutive frames is x and the result for the subsequent series of consecutive frames is $x+1$.

Figure 11 illustrates the two cases through a graph 1100 plot of result x 1110 versus frame sequence n 1108. According to the graph 1100, if the change point m , is such that $m=n+1$ or $m=n+25$, then the recognition result should be consistent for frame $n+1$ to $n+25$. This corresponds with the first case where the numeric character is the same. For the second case where there are two consecutive numeric characters, the result is x 1102 before point m 1106 and the result is $x+1$ 1104 after point m 1106.

For the first case, the result x does not change during the 25 frames and the possible result will be x when $F(x)$ is at a maximum. $F(x)$ is the sum of matching scores for all the 25 frames and is given by equation (7).

$$F(x) = \sum_{k=0}^{24} S_k(x), \quad x = 0, 1, 2, \dots, 10 \quad (7)$$

For the second case, the result changes from x to $x+1$ during the 25 frames, therefore there are 25 possible sums of matching score $G_m(x)$, which is given by:

$$G_m(x) = \sum_{k=0}^{m-1} S_k(x) + \sum_{k=m}^{24} S_k(x+1), \quad m = 0, 1, \dots, 24 \quad (8)$$

Considering all possibilities, the corresponding x of the maximum value in the set of values $\{F(x), G_0(x), G_1(x), \dots, G_{24}(x)\}$, where $x = 0, 1, 2, \dots, 10$ is the final recognition result of the single digit character that is shown in the digital clock 600 at the particular tested video frame. When applying step 720 to consecutive video frames, a sequence of final recognition results corresponding with the transitions of time $X(n)$, where $n=0, 1, \dots, 9$ of each digit character in the digital clock can be derived.

Next, at step 722, the sequence of final recognition results derived at step 720 is verified against the natural or correct transition sequence of the clock digit characters to confirm a correct final recognition result. Since the clock digit character transitions naturally follow a specific pattern, the accuracy of the sequence of final recognition results can be verified by checking whether the specific transition pattern is being followed or not.

For instance, the natural transition sequence for the SECOND digit character 614 (Figure 6) is given by:

$$D(n) = (D_0 + n) \% 10, \quad n = 0, 1, \dots, 9 \quad (9)$$

where $0 \leq D_0 \leq 9$, $D(0) = D_0$ and D_0 is the SECOND digit number of the actual game time at the instance where the recognition result sequence $X(n)$ is derived at step 720. If the recognition result sequence $X(n)$ is the same as $D(n)$, the recognition result $X(n)$ is considered to be correct. To determine whether $X(n)$ is running in the same sequence as $D(n)$, equation (10) is used. Equation (10) is given by:

$$\sum_{n=0}^9 (X(n) - D(n))^2 = 0 \quad (10)$$

By substituting equation (9) into equation (10), equation (11) is derived:

$$\sum_{n=0}^9 (X(n) - (D_0 + n) \% 10)^2 = 0 \quad (11)$$

The most possible correct result sequence for the SECOND digit character 614 (Figure 6) occurs when the $D(n)$ sequence is best matched with the sequence of final recognition results

$X(n)$. When a D_0 value is found to satisfy equation (11), the corresponding recognition result $X(n)$ is considered the final recognition result sequence. Accordingly, the recognition result for all the other digit characters can be derived utilising the verification process described in step 722.

After the correct result sequence is derived, the lapsed game time can be determined using equation (12). In the example embodiment, if a clock time is recognized with the format $mn:st$ after applying the matching process on each digit character, the time lapsed from "00:00", which is the start of the game, in seconds is given by:

$$T = (m \times 10 + n) \times 60 + s \times 10 + t \quad (12)$$

For the i^{th} recognition result $m(i)n(i):s(i)t(i)$, equations (13) to (16) are used.

$$t(i) = ((T(i-1) + 1) \% 60) \% 10 \quad (13)$$

$$s(i) = ((T(i-1) + 1) \% 60) \div 10 \quad (14)$$

$$n(i) = ((T(i-1) + 1) \div 60) \% 10 \quad (15)$$

$$m(i) = ((T(i-1) + 1) \div 60) \div 10 \quad (16)$$

where $T(i-1)$ is the time for the last result.

In the example embodiment, a criterion for correct clock time recognition is set, that is, if all the digits recognized do not tally with the values calculated in equations (13) to (16) for consecutive 10 seconds, the recognition results are regarded as wrong and steps 702 to 720 are performed again. This is because the clock may have disappeared or the position of the clock has shifted in the video. By repeating all the steps, new sample patterns will be collected and the recognition step after that is based on new templates. This verification process ensures that the recognition result is accurate and reliable.

It is appreciated that for different video content, the transition sequence of the clock time may be different. Therefore, the equation defining $D(n)$ depends on the specific video content. The $D(n)$ given by equation (9) above is for a soccer game where the clock time is increasing. In, for example, a basketball game, the clock time is decreasing,

hence it is appreciated that a different equation for $D(n)$ should be used. The approach adopted above is hereinafter denoted as digit transition sequence verification.

In another approach (hereinafter referred to as time transition sequence verification), the clock time can be used for verification instead of using digit transition verification. In this case, only the time transits are of concern and how the digits transit is of no concern. The time transition sequence in this case is given by,

$$T(i) = T(i-1) + 1 \quad (17)$$

where $T(i) = (m(i) \times 10 + n(i)) \times 60 + s(i) \times 10 + t(i)$. In this way, the calculation is simpler but accuracy is lower as shown in Figure 12.

If equation (17) is satisfied for consecutive N seconds of clock time recognition, the N results are considered reliable and the correct base time can be identified based on the format $mn:st$ that is recognized. After that, future final recognition result $X(n)$ can be verified against the identified correct base time to verify whether the final recognition result $X(n)$ is correct.

Figure 12 is a table 1200 illustrating the results of using the first method on six different soccer game videos each lasting about 45 minutes. The results show that there are no false recognition of clock digit characters. "gn" represents game name; "noc" represents number of digit characters; "drr" represents digit recognition rate, which is given by $\text{drr} = 100 \times (\text{number of digits recognized correctly}) / (\text{total number of digits})$; "tv" represents "drr with time transition sequence verification" and "dv" represents "drr with digit transition sequence verification". It is clear that "dv" is the more accurate approach compared to "tv".

The second method to derive the clock or game time of the digital clock 600 will now be described.

With reference to Figure 1, continuing from step 712, the second method involves, at step 714, detecting the video frame position at the first instance where the MINUTE digit character 616 (Figure 6) changes pattern after the detection of the presence of the digital clock 600 (Figure 6) from the start of the video.

Figure 13 shows a graph plot of $S(n)$ 1302 versus video frame, n 1304, for the TNPS sequence of the MINUTE digit character 616 (Figure 6) in a period no more than one minute after the start of the video at frame $n = 0$. The MINUTE digit character 616 (Figure 6) changes pattern at the minimum value of $S(n)$. As this is the first time the MINUTE digit character 616 (Figure 6) changes, the change is from "0" to "1". Due to the natural transition of time, when the MINUTE digit character 616 (Figure 6) changes, the SECOND digit character 614 (Figure 6), and the TEN-SECOND digit character 618 (Figure 6) must be "0", therefore the clock time in the video frame corresponding with the minimum value of $S(n)$ is inferred as "01:00". From time "01:00" onwards, the video frame sequence number, N_1 , corresponding to time "01:00" is synchronized with the clock time. Hence, where the clock time is required at any point in the video, the clock time can be calculated via the video frame sequence number N corresponding with that point. The clock time is given by:

$$t = (N - N_1)/F + 60 \quad (18)$$

where F is the frame rate, N is the video frame number corresponding to the clock time t and the unit of time t is in seconds.

It is appreciated that the TEN-SECOND digit character 618 (Figure 6) may be monitored instead of the MINUTE digit character 616 (Figure 6), in a period no more than 10 seconds from the start. In this case, the clock will show "00:10" when a pattern change is detected, and the clock time can be synchronized sooner than using the MINUTE digit character 616 (Figure 6). In the example embodiment, the MINUTE digit character 616 (Figure 6) is chosen to be monitored as the digital clock 600 (Figure 6) typically appears in the video within a minute from the start of the video.

Figure 14 is a table 1400 illustrating the results of using the second method on different soccer game videos. The results show that there is no false recognition of clock time provided the clock digit characters are initially segmented correctly, the clock appears in the video before it reaches one minute and the SECOND digit character 614 (Figure 6) changes every one second.

A method for aligning the text events and video events referred to previously in step 122 of Figure 1, will now be described in detail.

In some websites, for instance, sports.sina.com, the inaccuracy of the time-stamp in TWB can be as great as 3-4 minutes off the time of the actual video event happening. In order to tackle this problem, a robust algorithm is preferably utilized to align the text analysis result with the A/V features without using accurate time information and to detect the time period or boundary of the occurrence of the event in the video.

Figure 15 shows a typical situation where the time stamp 1510 of a text event from a TWB text stream 1506 does not correspond with the start of the broadcast video event 1502. As such, text analysis based on TWB only suggests a time duration or search range 1508 during which the identical event might happen in the video (indicated in Figure 15 as A/V stream 1504). The search range 1508 is defined based on the time stamp 1510 in the video, which is a fixed time of, for example, 5 minutes of backward search limit 1512 and 3 minutes of forward search limit 1514 from the time stamp 1510. The defined search range is such that it is long enough to cover the actual video event 1502. This decision may be made based on experience with the length of the video event. For instance, a typical "red card" event is expected to last 5 to 10 minutes long.

In the example embodiment, accurate event detection as well as event boundary detection is achieved by using a machine learning (e.g. Hidden Markov Model (HMM)) approach. When the event 1502 in the A/V stream 1504 is detected within the search range 1508 determined by text analysis on the text stream 1506, the A/V and text analysis results are aligned using a HMM based alignment method. The HMM based alignment method will now be described.

With reference to Figure 16A, the HMM based alignment method begins by building a search model with a "noise-event-noise" structure 1600 in A/V stream. Hence three HMM models are built, one for the event (*EventHMM*) 1602, one for the beginning noise (*NoiseHMM1*) 1604 and one for the ending noise (*NoiseHMM2*) 1606. The features used for HMMs are subsets of A/V features, F_2 404, F_3 406, F_4 408 and F_5 410 (in Figure 4). Different events use different subset features. The three HMMs are concatenated

according to the grammar defined by the “noise-event-noise” structure 1600 in Figure 16A.

Figure 16B shows a search model with grammar that allows self-jump between noise HMM models 1108. A self-jump structure improves the performance of the HMM based alignment method.

The three HMM models, that is event (*EventHMM*) 1602, noise (*NoiseHMM1*) 1604 and noise (*NoiseHMM2*) 1606, may be used as three separate probability classifiers. In the example embodiment, a sliding window is applied on the search range 1508 in Figure 15 to obtain three segments where the portions before, under and after the sliding window are called “beginning noise candidate”, “event candidate” and “ending noise candidate”, respectively. The three segments are sent to the three HMM Models respectively. The three probability scores obtained from each HMM Model are then weightily combined. By assigning different weight values to “event candidate” according to the shot count in the segment, the event duration can then be taken into consideration. Sliding the window over the whole search range 1508 in Figure 15, all possible partitions of the search range 1508 can be evaluated, and the one that yields the highest combined score will give the recognized event boundaries. This idea is explained as follows:

Let $S = \{s_1, s_2, \dots, s_N\}$ denote the search range 1508 defined by text analysis, and let X be the sequence of features from S , and Θ_{n1} , Θ_e and Θ_{n2} to denote the parameters of the beginning noise HMM Model 1604 in Figure 16, event HMM 1602 in Figure 16 and ending noise HMM 1606 in Figure 16 respectively.

To find the optimum partition of S into “beginning noise candidate” S_{n1} , “event candidate” S_e and “ending noise candidate” S_{n2} , all possible partitions are exhaustively evaluated. For example, given a trial evaluating S_e with M shots, the sequence of features from S_{n1} , S_e and S_{n2} are denoted as X_{n1} , X_e and X_{n2} respectively. The logarithm likelihood $p(X)$ with respect to the three HMM Models expands as

$$\begin{aligned} p(X) &= p(X | \Theta_{n1}, \Theta_e, \Theta_{n2}) \\ &= p(X_{n1} | \Theta_{n1}) + p(X_e | \Theta_e) + p(X_{n2} | \Theta_{n2}) \end{aligned} \quad (19)$$

In practice, the three probability scores are weightily combined to consider the shot count possessed by the "event candidate" X_e segment. As the noise segments X_{n1} and X_{n2} do not present distinguishing temporal patterns, higher weight is given to $p(X_e | \Theta_e)$. Hence instead of Equation (19), Equation (20) given by,

$$p(X) = w_n p(X_{n1} | \Theta_{n1}) + w_e G(M) p(X_e | \Theta_e) + w_n p(X_{n2} | \Theta_{n2}) \quad (20)$$

is used to sum the three logarithm likelihoods. The weights associated with the noise segments w_n and w_e are found empirically, and the weight for the event segment, $G(M)$, varies according to M . Specifically, $G(M)$ is modeled by Gaussian Model given by,

$$G(M) = \frac{1}{\sigma_e \sqrt{2\pi}} e^{-(M - \bar{M})^2 / 2\sigma_e^2} \quad (21)$$

where $\bar{M}$ and σ_e represent the mean and covariance of the shot count. The values $\bar{M}$ and σ_e are found during training and have different values for different events.

It is observed that the desired event boundary is aligned with shot boundary, hence there only exists limited number of possible partitions that can be exhaustively evaluated quickly. The one that yields the highest $p(X)$ is the optimum partition and the "event candidate" boundary in this optimum partition is the detected event boundary.

It is noted that so far the three HMM Models have been combined using a scheme 1700 represented by Figure 17A. Similar to Figure 16B, the scheme 1700 in Figure 17A is changed to the scheme 1702 in Figure 17B to allow self-jump between the noise HMM Models. A self-jump structure improves the performance of the HMM based alignment method.

Figure 18 provides an example of the probability scores with respect to all possible partitions of S for a "card" event (304 in Figure 3). The suggested search range 1508 in

Figure 15 by text analysis is from video frame number 141241 to 144991. The x axis 1802 and y axis 1804 represent the start and end frame respectively of S_e , and the z axis 1806 is the normalized probability $p(X)$. The optimum partition 1808 is highlighted, and the recognized event boundary is from video frame number 143366 to 143915.

The algorithms, processes, procedures and methods described can be implemented in a computer system 1900, schematically shown in Figure 19. The procedures may be implemented as software, such as a computer program being executed within the computer system (which can be a palmtop, mobile phone, desktop computer, laptop or the like) 1900, and instructing the computer system 1900 to conduct the method of the example embodiment.

The computer system 1900 comprises a computer module 1902, input modules such as a keyboard 1904 and mouse 1906 and a plurality of output devices such as a display 1908, and printer 1910.

The computer module 1902 is connected to a computer network 1912 via a suitable transceiver device 1914, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN).

The computer module 1902 in the example includes a processor 1918, a Random Access Memory (RAM) 1920 and a Read Only Memory (ROM) 1922. The computer module 1902 also includes a number of Input/Output (I/O) interfaces, for example I/O interface 1924 to the display 1908 (or where the display is located at a remote location), and I/O interface 1926 to the keyboard 1904.

The components of the computer module 1902 typically communicate via an interconnected bus 1928 and in a manner known to the person skilled in the relevant art.

The application program is typically supplied to the user of the computer system 1900 encoded on a data storage medium such as a CD-ROM or flash memory device and read utilizing a corresponding data storage medium drive of a data storage device 1930. The application program is read and controlled in its execution

by the processor 1918. Intermediate storage of program data maybe accomplished using RAM 1920.

Example embodiments of the present invention enable automatic event detection in live videos by using multi-modal (audio, video, text) analysis techniques. Such multi-modal (audio, video, text) analysis techniques advantageously include text analysis and event detection from web-casting text, automatic video clock detection and game time recognition from the broadcast sports video, and text and video alignment to automatically detect events and the event time boundaries in videos. Specifically, the use of text information greatly improves the event detection performance compared with traditional methods that uses A/V features only, and the text analysis also enable extraction of semantics of the detected events. Example embodiments of the present also employ efficient and accurate methods to automatically detect the video clock and recognize the game time in broadcast sports video and uses a robust algorithm to align the detected text event with the video to identify the event boundaries.

Example implementations of the present invention will now be described with reference to Figures 20 to 38.

Video footage processing, particularly automatic video processing, requires some knowledge of the content of the footage. For example, in order to generate a video summary of events within the footage, the original footage needs some form of annotation of the footage. In this way a personalised video summary may be generated that only includes events that meet one or more criterion.

Figure 20 exemplifies an example implementation of a method for classifying events within video footage. Video footage 2000 may be stored or received live including three different streams: structured text broadcast (STB), video and audio. In step 2002, one or more features and/or temporal information are extracted from at least said structured text broadcast stream. In step 2004, the footage is temporally annotated with the features and/or temporal information. In step 2006, temporally adjacent annotated features and/or temporal information are analysed to determine information about one or more events within said footage.

An example application is annotating sports video. In sports video, typical annotations may include the time of an event, the player or team involved in the event and the nature or type of event. The venue of the event may also be used as an annotation. For the following example implementations, football or soccer will be used as one example, although it will be appreciated that other example implementations are not so restricted and may cover annotated video generally.

A user of sports video will typically have a preference for given players or teams and/or a particular nature or type of event. Accordingly once annotated, events that meet the preferences may be easily selected from the annotated footage to generate a personalised video summary. The summary may include video, audio and/or STB streams.

Figure 21 shows an example implementation of a method for personalised video generation from stored video footage, where the footage has been annotated. In step 2102, the preferences are set for which events to include. In step 2104, events that have annotations that satisfy the set preferences are selected from the stored video footage. In step 2106, the summary is generated from the selected events.

The methods shown in Figures 20 and 21 may be employed independently or in combination. Typically they may be combined and employed in a system as shown in Figure 22.

Figure 22 shows an example implementation of a system for indexing video and generating personalised video. Video footage 2200 is received at the input 2201. The content data may be stored or processed immediately. Each stream of the video footage is separated and provided correspondingly to a video processor 2202, an audio processor 2204 and an STB processor 2206, to annotate the video footage. Each processor may interface with temporary data storage 2208 (for example, Random Access Memory (RAM)) and permanent data store 2210 (for example, a hard disk), which includes algorithms and/or further data to assist the classification of each event. The annotated footage is then stored in a database in the permanent data store 2210.

User preferences 2203 are also received at the input 2201. Video generation processor 2212 receives the preferences and scans the database for events with annotations that satisfy the preferences. The summary video is provided at the output 2214, or may be stored in the permanent data store 2210 for later retrieval.

Each processor may take the form of a separate programmed digital signal processor (DSP) or may be combined into a single processor or computer.

In an example implementation, the content data is received (step 2000 in Figure 20) as shown in Figure 23, as an STB stream 2300, a video stream 2302 and an audio stream 2304. The data may be received and processed in real time or may be stored for offline analysis. The STB stream may be created separately from the video/audio streams or from a different source, but may easily be integrated with the video and audio streams for processing.

In order to facilitate annotation, a framework is necessary. In an example implementation, each of the streams of the footage is analysed and "keywords" are extracted (step 2002 in Figure 20) based on both spatial and temporal features in each of the streams. These features are mainly low-level features of the three media contents. For the video stream, the features may include colour and intensities, histograms, motion parameters of key frames and video shots. For the audio stream, the features may include Mel frequency Cepstral Coefficients (MFCC), Zero Crossing Rate (ZCR), linear prediction coefficient (LPC), short time energy (ST), spectral power (SP). For the STB stream, the features may include extracted terms and their distributions.

The video features, for example, have two axes: temporal and spatial, the former refers to its variations along time, the latter refers to its variations along spatial dimension, like horizontal and vertical positions.

For example, the STB stream 2300 is subjected to STB analysis 2310 including parsing the text to extract key event information such as who, what where and when. Then one or more "play keywords" 2316 (PKW) are extracted from the STB stream. The keywords are defined depending on the type of footage and the requirements of annotation.

The video stream 2302 is subjected to video analysis 2306 including video structural parsing into play, replay and commercial video segments. Then one or more "video keywords" 2312 (VKW) are extracted from the video stream and/or object detection is carried out.

The audio stream 2304 is subjected to audio analysis 2308 includes audio low-level analysis. Then one or more "audio keywords" 2314 (AKW) are extracted from the audio stream.

Once the keywords are extracted, the keywords may be aligned in time for each stream 2318. Player, Team and Event detection and association 2319 takes place using the keywords. Here events refer to actions that are taking place during sports games. For instance, events for soccer game include goal, free kick, corner kick, red-card, yellow-card, etc.; events for tennis game include serve, deuce, etc. Each replay may then be classified 2320, for example by identifying who features in each event, when each event occurred, what happened and where each event occurred. The semantically annotated video footage may then be stored in a database 2322.

STB Analysis

STB allows easier parsing of information that is less computationally intensive and more effective compared to parsing transcriptions of commentary. Normal commentary may have long sentences, may be unstructured and may involve opinions and/or informal language. All of this combines to make it very difficult to reliably extract meaningful information about events from the commentary. Prior art Natural Language Parsing (NLP) techniques have been used to parse such transcribed commentary, but this has proven highly computationally intensive and only provides limited accuracy and effectiveness.

An example of an STB stream is Sports Web casting Text (SWT). Sports game annotators manually create SWT in real-time, and the SWT stream is broadcast on the Internet. SWT is structured text that describes all the actions of sports game with relatively low delay. This allows extraction of information such as the time of an event, the player or team involved in the event and the nature or type of event. Typically SWT

provides information on action and associated players/teams approximately every minute during a live game.

SWT follows an established structure with regular time stamps. Figure 24 shows the structure of an example SWT stream 2401. Each sentence is typically short and the language simple, typically relating to action taking place in the footage. This allows the information to be parsed more easily and more reliably than in the prior art. The SWT stream consists of a sequence of action description tokens (ADT) 2400. Current commercially available SWT typically delivers 1 to 3 ADT(s) per minute depending on the activity levels each minute.

The PKW extracted from the SWT may be used to identify events and may be used to classify each event.

In order to analyse the SWT and generate the PKW over the whole footage (2316 in Figure 23), the game introduction 2410 is first parsed to obtain general information and then each of the ADTs 2400 are parsed to get temporal information relating to events within the footage. Examples of parsing include processing of stop words, stems and synonyms on the SWT stream.

The PKW may consist of a static and dynamic component. In Figure 25, the static part 2500 is extracted and stored in the Sport Keywords Database (SKDB) 2502, including a set of sports event and teams. The dynamic part 2504, including players' names and events, is extracted over the length of the game and also stored in the SKDB 2502.

The dynamic component includes parsing over each ADT unit. Each ADT is parsed into the following four items: Game-Time-Stamp 2506; Player/Team-ID 2508; Event-ID 2510; and Score-value 2512. That will be followed by an extraction performed on the PKW over a window of a fixed length, to extract the true sports event type and the associated player. Parsed ADTs within a time window (ADTw) are processed to extract player keywords and associated event keywords. For soccer or football an example window of 2 minutes may be used, since typically each soccer or football event has a longer duration than 1 minute.

As shown in Figure 26, the static part 2600 of the play keyword e.g. the name of the game, venue, teams, players from each team, and referees, etc, may be extracted at the beginning of the commentary. The dynamic component 2602 of the play keyword may be extracted over the duration of the game.

In sporting footage events may be inter-dependent instead of being considered as isolated events. As seen in Figure 27 an event foul 2700 may cause an event of free-kick 2702, which in turn may result in a goal 2704. Knowledge of such inter-relations may assist in segmenting events with accurate temporal boundaries for video summary or query. This process is called context sports event parsing.

Video Analysis

Depending on the level of event granularity or temporal resolution required the VKW may be used to further refine the indexed location and the indexed boundaries in the footage used to represent the event. For example the event may be detected using just the PKW, resulting in an event window of about 1 minute. If the event is first identified using the PKW, the VKW may be used to refine the event window to a much shorter period. For example using the VKW, the event may be refined to the replay (already chosen by the human production director) of the event within the footage.

The VKW may also be used in synchronising the event boundaries within video stream and the STB stream.

Video analysis (2312 in Figure 23) may involve video shot parsing as shown in Figure 28 and/or VKW extraction and object detection as shown in Figure 29. The video shot parsing and/or VKW extraction and object detection may be used to refine the indexing of the events in the footage.

Video shot parsing involves parsing the footage into types of video segments (VS). Figure 28 shows extraction into commercial segments 2800, replay segments 2802; play video segments (PVS) 2804 and break video segments (BVS) 2806. The commercial segments 2800 are detected using a commercial detection algorithm 2808. The replay segments 2802 are detected using a replay detection algorithm 2810. The PVS 2804 and

BVS 2806 are detected using a play-break detection algorithm. It is not necessary for all algorithms to be used. For example, if only replays are required to be extracted, then only the replay algorithm is required. However the system may be employed more generally to extract any type of video segments from the footage.

An example of a commercial detection algorithm is disclosed in United States Patent 6,100,941. TV commercials are detected based on whether a black frame has occurred. Other parameters are used to refine the process including the average cut frame distance, cut rate, changes in the average cut frame distance, the absence of a logo, a commercial signature detection, brand name detection, a series of black frames preceding a high cut rate, similar frames located within a specified period of time before a frame being analyzed and character detection.

An example of a replay detection algorithm is disclosed in a paper by L.Y. Duan, M. Xu, Q. Tian, CS Xu, entitled "Mean shift based video segment representation and applications to replay detection", published in ICIP2004, Singapore. Replay segments are detected from sports video, based on mean-shift based video segmentation where both spatial and temporal features are clustered to characterize video segments. For example colours and motions may be utilized for clustering. Subsequently parameters of these clusters can be used to detect replays robustly because of special characteristics of the replay logos.

An example of a play-break detection algorithm is disclosed in a paper by L. Xie, S.-F. Chang, A. Divakaran and H. Sun, entitled "Structure Analysis of Soccer or football Video with Hidden Markov Models", published in Proc. International Conference on Acoustic, Speech and Signal Processing, (ICASSP-2002), Orlando, FL, USA, May 13-17, 2002. A HMM based method may be used to detect Play Video Segments (PVS) 2804 and Break Video Segments (BVS) 2806. Dominant colour ratio and motion intensity are used in a HMM models to model two states. Each state of the game has a stochastic structure that is modelled with a set of hidden Markov models. Finally, standard dynamic programming techniques are used to obtain the maximum likelihood segmentation of the game into the two states.

As shown in Figure 29 after TV commercial segments have been removed, sports video segments, including play/break/replay segments will be processed to extract VKWs, the structure of which depend on the type of sports game. The rules for the VKW extraction for different sports types are stored in the knowledge database 2912.

There are at least three types of video keywords. A first type has a length of one video shot. A second type is a sub-video shot which is less than one video shot. Finally a third type is a super-video shot that covers more than one video shot.

An example of a sub-video shot would be where one video shot can be rather long, including several rounds of camera panning which covers both defence and offence for a team, in for example basketball or football. In these situations it's better to segment these long video shots into sub-shots so that each sub-shot describes either a defence or an offence.

Similarly, a super-video shot relates to where more than one video shot can better describe a given sports event. For instance in tennis video, each serve starts with a medium view of the player who is preparing for a serve. The medium view is then followed by a court view. Therefore the medium view can be combined with the following court view to one semantic unit: a single video keyword to represent the whole event of ball serving.

The process of determining VKW types is now described. In step 2900, intra video shot features (colour, motion, shot length, etc.) are analyzed. In step 2902, middle level feature detections are performed to detect sports field region, camera and object motions. In step 2904, a determination is made as to whether sub-shot based video keywords should be considered. Sub-shot video keywords can be identified and refined through step 2900, step 2902 and step 2904. Similarly super-shot video keywords are identified in step 2906 so that one semantic unit can be formed to include several video shots.

In step 2908, a video keyword classifier parses the input video shot/sub-shot/super-shot into a set of predefined VKWs. Many supervised classifiers can be used, such as neural networks (NN), supporting vector machine (SVM).

In step 2910, various types of object detection can be used to further annotate these video keywords, including soccer ball or football, goalmouth, and other important land marks. This allows higher precision in synchronising events between the streams.

An example of object detection is ball detection. As shown in Figure 35, in typical footage the soccer ball or football may be highly distorted due to many reasons, including high speed moving of the balls, and cameras' view changes, and occlusions of players, etc. Two methods may be used in combination to detect the ball trajectory to avoid distortion problems. Firstly, ball candidates are detected 3500 by eliminating non-ball shape objects. Secondly, the ball trajectory 3502 is estimated in the temporal domain. In this way, any gaps or video shots missing the ball, which may be caused by occlusion or ball being too small, can be compensate for.

A further example of object detection is goalmouth location. The process is shown in Figures 36 in which step 3600 encompasses the sports field being detected by isolating the dominant green regions. In step 3602, a Hough Transform-based line detection is performed on the sports field area. In step 3604, coarse level play field orientation detection is performed. In step 3606, vertical goalposts are isolated and in step 3608, horizontal goal-bar are isolated, by colour-based region (pole)-growing. In step 3610, post-processing is used to detect the localized goalmouth from the input video.

Audio Analysis

Similarly to VKW, the AKW may be used to further refine the indexed location and the indexed boundaries in the footage used to represent the event. The AKW may also be used in synchronising the event boundaries within audio stream and the STB stream.

Figure 30 shows the process of AKW extraction (2314 in Figure 23) from the audio stream. The AKW is defined as a segment of audio where we can observe the presence of several classes of sounds with special meaning to semantic analysis of sports events. For instance, the excited or plain voice pitches of the commentator's speech or the sounds of the audience, etc may be indicative of an event. It is very useful to detect these special sounds robustly to associate with varying sports events.

Some example AKWs are listed below. AKWs may either be generic or sports specific.

Generic Audio Keywords

- Plain Commentator Speech
- Excited Commentator Speech
- Plain Audience Sounds
- Excited Audience Sounds

Domain-specific Audio Keywords

- Whistling in Basketball and Soccer or Football
- Hitting Ball in Tennis

Low level features 3000 that may be used for AKW extraction include Mel frequency Cepstral Coefficients (MFCC), Zero Crossing Rate (ZCR), linear prediction coefficient (LPC), short time energy (ST), spectral power (SP), and Cepstral coefficients (CC), etc. The audio data is sampled from the audio stream at a 44.1 KHz sample rate, stereo channels and 16 bits per sample.

The MFCC features may be computed from the FFT power coefficients of the audio data. A triangular band pass filter bank filters the power coefficients. The filter bank consists of $K=19$ triangular filters. They have a constant mel-frequency interval, and cover the frequency range of 0Hz – 20050 Hz. The Zero crossing rate may be used for analysis of narrowband signals, although most audio signals may include both narrowband and broadband components. Zero crossings may also be used to distinguish between applause and commentating.

Supervised classifiers 3002 can be used for AKW extraction such as multi-class support vector machine (SVM), decision tree and hidden Markov model (HMM). Samples of the pre-determined AKW samples are prepared first, classifiers can be trained over the training samples, and then they can be tested over testing data for performance evaluation.

Time alignment

Cross-media alignment (2318 in Figure 23) using time-stamps embedded in the sports video/audio and STB streams may be required, as the timing of each stream may not be synchronised. Alternatively a machine learning method such as HMM may be used to make such corrections, which is useful to correct any delays the STB texts. The use of HMM is as described above with reference to Figures 15 to 18.

Player, Team and Event detection

It may be useful, depending on the application, to detect events within the footage, and annotate the footage with this additional information.

In a first example, events are detected (step 2004 in Figure 20 and 2319 in Figure 23) by analyzing the STB stream. The PKW extracted from the STB stream is used to detect events. The association between the PKW and an event is based on knowledge-based rules. In Figure 23, rules are stored in the knowledge database 2324. For example PKW such as goal or foul in the SWT provides a time stamp for an event. Then the boundaries of the event are detected using the VKW and AKW and the streams synchronised.

The player and team involved in each event are determined based on an analysis of the surrounding PKW.

In a second example, events are identified based on the video stream. As one possible case, in the first example, the visual analysis previously described is used to detect each of the replays inserted by the human production director. Each of the replays is then annotated, and stored in a database. Various methods may be used to analyse the video stream and associate with events. For example with machine learning methods such as neural networks, supporting vector machines, and hidden markov models, may be used to detect events in this configuration.

As seen in Figure 37, the footage is stored in the database once fully annotated. Three parsed streams are stored including STB stream 3710, video stream 3720 and audio stream 3730. As we can see that PKW 3712 from the STB are time stamped at each

minute 3740 while VKW 3722 and AKW 3732 are indexed at seconds and milliseconds intervals. Three streams can be fused together for various applications such as event detection, classification, and personalized summary. Based on the required granularity of particular applications, one, two or three streams can be used for generation of summary video. They can be used either in sequence or in parallel.

Replay Classification

It may also be useful, depending on the application, to detect and classify replays (2320 in Figure 23) within the footage, and annotate the footage with this additional information.

Replay detection and classification is described in detail in other sections. Thus the indexing and classification of replays simply forms another level of semantic annotation of the footage once stored in the database.

Generation of personalised video summary

According to a first example implementation, Figure 31 shows procedures of personalized video summary based on large content collections with multiple games of many tournaments. In step 3100, users give their preferences on their desired video summary, possibly including players, teams or specific sports events, and possible other usage constraints like the total length of the output video, etc. In step 3102, a set of PKW input can be identified based on users' input. In step 3104, the annotated sports video content database is searched using the set of PKW input to locate corresponding game video segments. In step 3106, the selected segments are refined based on a preferred length or other preferences. In step 3108, the video summary is generated.

For instance, a video summary of all the goals by the football star David Beckham's can be created by identifying all games for this year, and then identifying all replays associated with David Beckham and selecting those replays that involve a goal.

Figure 38 shows the above created summary can be refined by using VKW 3820 and AKW 3830 where boundaries of video/audio segments can be adjusted based on boundaries of VKW 3820 and AKW 3830 instead of relying on PKW 3810 only which has

a granularity of one minute. Machine learning algorithms such as HMM models, neural networks or supporting vector machines, can perform the boundary adjustment.

According to a second example implementation, Figure 32 shows how a video summary might be generated from an annotated sports video using a text summary of the game. A typical text summary of a sports game consists of around 100 words, including names of teams, outcome of the game, and highlights of the game.

Firstly the text summary 3200 is parsed to produce a sequence of important action items 3202 identified with key players, actions and teams, and other possible additional information such as time of the actions, name and location of the sports games. This generates the preferences (2100 in Figure 21) for the event selection.

SWT parsing produces sequences of time-stamped PKWs that describe actions taking place in the sports game. The event boundaries are refined and aligned with the video stream and audio stream, and the annotated video is stored in a database 3206.

The preferences from the text summary are then used to select 3204 which events to include (step 2102 in Figure 21) by searching the database 3206. Sports highlight/events candidates are organized based on a set of pre-defined keywords for given sports; for example, sports highlights for soccer or football include goal, free kick, corner kick, etc. All these sports keywords are used in both text summary and text broadcasting script.

The selection of events may be further refined 3208, depending on preferred length of summary or other preferences.

Finally, the video summary 3210 is generated (2104 in Figure 21) based on the video shots and audio corresponding to the time-stamped action items selected above.

According to a third example implementation, a learning process may be used for detecting and classification of replays, and summary generation. Video replays are widely used in sports game broadcasting to show highlights occurred in various sessions

of the broadcasting. For typical soccer or football games there are 40 – 60 replays generated by human production directors for each game.

There are three types of replays generated at three different stages of broadcasting. Instant replay video segments RVS_{instant} appear during regular sessions such as first half and second half of games. Break replay video segments RVS_{break} and post-game replay video segments RVS_{post} appear during the break sessions between two half play sessions, and the post-game sessions. On average, there are 30 – 60 RVS_{instant} for each soccer or football game while numbers of RVS_{break} , RVS_{post} are much smaller because only the most interesting actions or highlights can be selected for showing during the break and post-game sessions.

Figure 33 shows a method of detecting and classifying replays. In step 3300, video analysis is used to detect a replay logo within the footage, to detect each event. In step 3302, the type of replay identified such as RVS_{instant} , RVS_{break} and RVS_{post} . In step 3304, each replay is then classified into a pre-defined set of event categories 3304 such as such as goal, goal-saved, goal-missed, foul etc using analysis of the STB stream. Once an RVS_{instant} is detected the PKW is analysed in the preceding time period (0.5 ~ 1.0 minutes) to identify the category.

For a soccer or football game, where the total number of replays are denoted $N\text{-}RVS_{\text{instant}}$, $N\text{-}RVS_{\text{break}}$, $N\text{-}RVS_{\text{post}}$, then $N\text{-}RVS_{\text{break}}$, and $N\text{-}RVS_{\text{post}}$ are much smaller than $N\text{-}RVS_{\text{instant}}$. Since human production directors carefully select $N\text{-}RVS_{\text{break}}$ and $N\text{-}RVS_{\text{post}}$ from $N\text{-}RVS_{\text{instant}}$, the selection process done by human directors can be learned. The learning process may involve a machine learning methods such as neural networks, decision trees or supporting vector machines such so that different weightings or priorities can be given to different types of $N\text{-}RVS_{\text{instant}}$, even together with consideration of users' preference to create more precise video replays for users.

Figure 34 shows an example learning process. The video and web-casting text data is collected for multiple games. For each game all RVS_{instant} 3400 and RVS_{break} RVS_{post} for 3402 are identified. Then each replay break is categorised 3404 by visual and audio analysis with manual corrections if needed. Then machine learning is used to calculate the weighting factors 3406 for different types of replay, j , from the two collections. These

weighing factors then reflect how human production directors use replays when they create RVS_{break} and RVS_{post}.

Based on the detected and classified RVS_{instant} as well as the learned weighting factors in terms of their importance, a selection can be made of the RVS_{instant} to generate the personalised video summaries automatically.

Many modifications and other embodiments can be made to the system and its method by those skilled in the art having the understanding of the above described disclosure together with the drawings. Therefore, it is to be understood that the device and its utility is not to be limited to the above description contained herein only, and that possible modifications are to be included in the claims of the disclosure.

THE CLAIMS

1. A method of event detection in a video stream, comprising the steps of:
detecting a text event in a casting text stream external to the video; and
identifying a corresponding video event in the video stream based on the detected text event.
2. The method as claimed in claim 1, wherein the step of identifying of the corresponding video event comprises time aligning the casting text stream with the video stream.
3. The method as claimed in claim 2, wherein the step of time aligning comprises:
detecting a time stamp in the casting text stream associated with the text event;
detecting a digital clock region in frames of the video stream;
extracting time information from the digital clock region; and
aligning the text event with video event based on the time stamp and the extracted time information.
4. The method as claimed in claim 3, wherein the step of extracting the time information from the digital clock region comprises:
utilising periodic pattern change detection to identify digit regions within the digital clock region;
collecting numerical pattern templates from different frames based on the pattern change of another clock digit, each pattern template representing a digit character displayed in the digit region; and
recognising the digit characters displayed in the digital clock region utilising the pattern templates.
5. The method as claimed in claim 3, wherein the step of extracting the time information from the digital clock region comprises:
utilising periodic pattern change detection to identify digit regions within the digital clock region;
identifying a first change in a pattern of one of the digit regions; and

inferring the clock time based on the identified first change in the pattern of the one digit region.

6. The method as claimed in any one of claims 3 to 5, wherein the identifying of the corresponding video event comprises defining a search range for identifying the video event in the video stream aligned with the casting text stream, the search range being around the temporal location of the time stamp in the aligned casting text stream and video stream.

7. The method as claimed in claim 6, further comprising utilising a model with a "noise-event-noise" structure for identifying the video event within the search range.

8. The method as claimed in claim 7, wherein the "noise-event-noise" structure comprises a self-jump structure.

9. A system for event detection in a video stream, the system comprising:
a detector for detecting a text event in a casting text stream external to the video;
and
an identification unit for identifying a corresponding video event in the video stream based on the detected text event.

10. The system as claimed in claim 9, wherein the identification unit identifies the corresponding video event in the video stream based on the detected text event by time aligning the casting text stream with the video stream.

11. The system as claimed in claim 10, wherein the identification unit comprises:
a detection unit for detecting a time stamp in the casting text stream associated with the text event and detecting a digital clock region in frames of the video stream;
an extractor for extracting time information from the digital clock region; and
a time aligning unit for aligning the text event with video event based on the time stamp and the extracted time information.

12. The system as claimed in claim 11, wherein the extractor unit extracts time information from the digital clock region utilising periodic pattern change detection to identify digit regions within the digital clock region; collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and recognising the digit characters displayed in the digital clock region utilising the pattern templates.
13. The system as claimed in claim 11, wherein the extractor unit extracts time information from the digital clock region utilising periodic pattern change detection to identify digit regions within the digital clock region; identifying a first change in a pattern of one of the digit regions; and extracting the time information based the identified first change in the pattern of the one digit region.
14. The system as claimed in any one of claims 11 to 13, wherein the identification unit identifies the corresponding video event in the video stream based on the detected text event by defining a search range for identifying the video event in the video stream aligned with the casting text stream, the search range being around the temporal location of the time stamp in the aligned casting text stream and video stream.
15. The system as claimed in claim 14, wherein the identification unit utilises a model with a "noise-event-noise" structure for identifying the video event within the search range.
16. The system as claimed in claim 15, wherein the identification unit utilises a model with a "noise-event-noise" structure comprising of a self-jump structure.
17. A method of extracting time information from a video stream, the method comprising:
detecting a digital clock region in frames of the video stream; and
utilising periodic pattern change detection to identify digit regions within the digital clock region.
18. The method as claimed in claim 17, comprising
detecting the digital clock region in the frames of the video stream; and

utilising the periodic pattern change detection to identify the digit regions within the digital clock region;

collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and

recognising the digit characters displayed in the digital clock region utilising the pattern templates.

19. The method as claimed in claim 17, comprising:

detecting the digital clock region in the frames of the video stream;

utilising the periodic pattern change detection to identify the digit regions within the digital clock region;

identifying a first change in a pattern of one of the digit regions; and

extracting the time information based the identified first change in the pattern of the one digit region.

20. A system of extracting time information from a video stream, the system comprising a detection unit for detecting a digital clock region in frames of the video stream by utilising periodic pattern change detection to identify digit regions within the digital clock region.

21. The system as claimed in claim 20, wherein the detection unit detects the digital clock region in frames of the video stream by

utilising the periodic pattern change detection to identify the digit regions within the digital clock region;

collecting pattern templates for the digit regions from different frames, each pattern template representing a digit character displayed in the digit region; and

recognising the digit characters displayed in the digital clock region utilising the pattern templates.

22. The system as claimed in claim 20, the detection unit detects the digital clock region in frames of the video stream by

utilising the periodic pattern change detection to identify the digit regions within the digital clock region;

identifying a first change in a pattern of one of the digit regions; and

extracting the time information based the identified first change in the pattern of the one digit region.

23. A data storage medium having stored thereon code means for instructing a computer system to execute a method of event detection in a video stream, comprising the steps of:

- detecting a text event in a casting text stream external to the video; and

- identifying a corresponding video event in the video stream based on the detected text event.

24. A data storage medium having stored thereon code means for instructing a computer system to execute a method of extracting time information from a video stream, the method comprising:

- detecting a digital clock region in frames of the video stream;

- utilising periodic pattern change detection to identify digit regions within the digital clock region.