Content Based Analysis for Video from Snooker Broadcasts.

Conference Paper · January 2002

Source: DBLP

CITATIONS
7

READS
121

3 authors, including:

Hugh Denman
Google Inc.
14 PUBLICATIONS **179** CITATIONS
SEE PROFILE

Anil Kokaram
Trinity College Dublin
130 PUBLICATIONS **1,119** CITATIONS
SEE PROFILE

Content Based analysis for video from Snooker Broadcasts*

H. Denman, N. Rea and A. Kokaram

Electronic and Electrical Engineering Department,
University of Dublin, Trinity College, Dublin, Ireland.
hdenman@cantab.net, {oriabhan, anil.kokaram}@tcd.ie

Abstract. This paper presents three new tools appropriate for content analysis of sports, applied to footage from snooker broadcasts in particular. The first tool is a new feature for parsing a sequence based on geometry without the need for deriving 3D information. The second tool allows events to be detected where an event is characterised by an object leaving the scene at a particular location. The final tool is a mechanism for summarising motion in a shot for use in a content based summary. As a matter of course, the paper considers a number of enabling techniques such as the removal of irrelevant objects and object tracking using a particle filter. The paper shows that by exploiting context, a convincing summary can be made for snooker footage.

1 Introduction

The problem of content-level processing for multimedia has much exercised the research community in recent years [1]. Video, in particular, is the focus of much research; the high volume of content makes content-level video processing difficult. Content-based retrieval for video involves developing search-engines capable of running queries against a corpus of video footage.

Common to most content-level multimedia applications is a feature extraction stage, in which the video footage is annotated with derived information or metadata. For example, simple summary generation is often based on the extraction of the first frame of each scene; in this instance scene cuts are the feature to be extracted. Automated feature extraction has been implemented in numerous research and commercial systems. Some systems augment the stored corpus with extracted features such as global colour, texture and motion [2], and dominant camera motion [3]. Systems such as QBIC, Virage, VisualGREP rely on distance measures against these features to enable query by example.

However, any domain-agnostic system relying on automated feature extraction will be limited in the kinds of query that it can support. Specifically, semantic level queries against the 'meaning' of the video are not possible at present; supporting such queries in a domain-agnostic system is an AI-complete problem,

* Work sponsored by Enterprise Ireland Project MUSE-DTV (**M**achine **U**nderstanding of **S**ports **E**vents for **D**igital **T**elevision)

as the machine must understand the video presented. To enable high-level queries against the corpus, it is necessary to restrict the domain addressed [4]. Previously successful domain-specific systems include the work of D.D. Saur *et al* in basketball footage [5], and the tennis classification system developed by Jain [6]. This latter was capable of classifying footage of tennis shots into passing shots, volleys, etc.

This paper presents three new tools applied to televised snooker coverage. The focus of this work is the creation of a semantically meaningful summary of the game. This implies identifying meaningful table views and ball movement events. We illustrate that unlike previous work in sports retrieval [6], the content analysis engine *does not need to extract the 3D scene geometry* to parse the video effectively. In particular, the first tool is a new geometric feature for parsing this kind of video sequence. The second tool deals with event detection, while the last considers a new kind of motion representation for building single images to represent entire shots. All these tools can be generalised to other sports, and are directly applicable as described to pool and billiards as well as snooker.

In the next section, the initial parsing of the footage according to scene geometry is presented and assessed. There then follows a discussion about event detection, which implies the need for disambiguation between player and ball in the case of snooker. Then finally a discussion about motion representation leads into a consideration of the Motion History Image for representing information about an entire shot in a single picture.

2 Parsing sport video using implicit geometry

Sports such as tennis, snooker, badminton, and cricket all occur within predefined playing limits. Most of the video footage from these events contains well delineated field lines in the views which contain the most information about the play - for example, the court lines in tennis, and the edge of the table in snooker. It is sensible then that the video should be parsed according to the geometry of the camera view. This immediately exploits the context of these kinds of events and would conceivably be a more powerful approach than the generic use of histogram based shot cut detection [1]. This is because the geometry would not only allow the scene cut to be identified but also the camera view and hence the importance of that shot for summary purposes. For instance, in both tennis and snooker, shots of the crowd and of the players can be considered less important than shots containing game events, and summarised simplistically, or discarded entirely.

Previous work has considered the use of 3D scene geometry [6] to generate a correspondence between image features and *real* court markings. This information could be used also for identifying the camera view, hence allowing the video to be parsed. This can be a complicated exercise, and in fact a much simpler idea yields the same information. What is of interest is the relative geometry of the lines within each image; it is not important to know how that geometry relates to the *real world*, only how it relates to other geometries in

the footage. We know that edge information in sports footage well represents most of the geometry of the court. Therefore summarising the geometry of that edge information *in the scene view* will yield a useful feature for parsing. The Hough transform of an image containing edge information yields concentrated peaks representing significant straight lines. The nature of this Hough *surface* will therefore follow changes in the edge information. Summarising the Hough transform should therefore yield the feature we seek, and we propose to use the second order central moment as follows:

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p (j - y_c)^q f(i, j) \tag{1}$$

where i, j are the co-ordinates in Hough space, p, q the moment order and x_c, y_c the co-ordinates of the origin in Hough space and f represents the Hough transform. Here we use the ρ, θ form of the Hough Transform with $\theta = [0 : 180°]$; $\rho = [0 : 720]$ for Standard Definition PAL frames.

In each sport, the edge information can be extracted with different emphases. For tennis, the edges of the court lines are important, and we have had some success at using generic edge detection here [7]. For snooker, the playing area is a green table. The main view that is important for content summary is the full table view, shown in figure 5. By segmenting this green area, and thus locating the edge of the table, we have access to the geometry for summarisation. A segmentation of the green area of each image frame is created using a threshold on the two colour planes as follows

$$\begin{gathered} b(i, j) = \{(G(i, j) - R(i, j)) > T\} \wedge \\ \{(G(i, j) - B(i, j)) > T\} \end{gathered} \tag{2}$$

where $b(i, j)$ is set to 1 at sites where both colour differences exceed T and set to 0 otherwise. The difference signals measure the *greenness* of the images.

Using $T = 25$, figure 1 shows typical behaviour; the middle image shows the segmentation result from the frame shown on the left. The rightmost image shows the Hough transform of the contour of the detected table. It is the relative location of the Hough peaks that evolves with change in shot. Figure 2 shows the same processing applied to a different shot with a different view, and the change in orientation can clearly be seen.

Figure 3 shows the evolution of the 2nd order moment for each frame calculated using equation 1, for a sequence of 3500 frames. The colour bar along the bottom of the plot indicates the different shot geometries appearing in the footage. The different camera views, and thus *semantic* events, are clearly delineated as plateaus with significantly different values of 2nd order moment. Furthermore, crowd scenes and close ups of players show significantly lower plateaus in the moment signal since they contain significantly less straight line geometry. Wipes, fades and dissolves can all be detected as impulsive transitions in the moment signal, and shot cuts show up clearly as step edges.

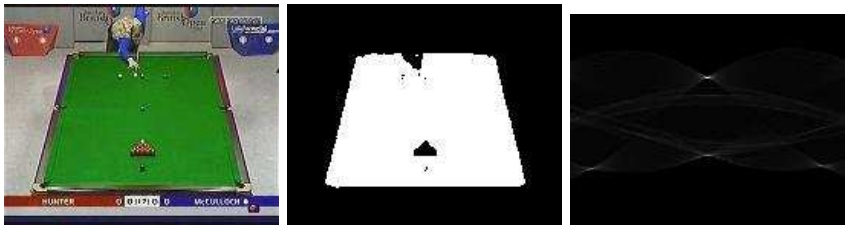


Fig. 1. Characteristic geometry of a full table shot. Left to right: the shot, the green area, and the Hough transform.

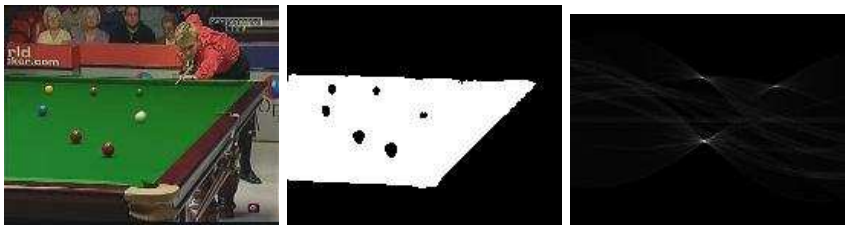


Fig. 2. Characteristic geometry of a partial table shot. Left to right: the shot, the green area, and the Hough transform.

A HMM is used to determine the correspondence between 2nd order moment values and camera views. This is not the primary focus of this paper, but is outlined in the appendix.

The efficacy of this approach has been evaluated using the Receiver Operating Characteristic (ROC), where the independent variable is the green value threshold. It was found that the 2nd order moment of the Hough transform allows detection of relevant table shots with 98.19% precision and 98.93% recall, at the optimal value for T (25). Figure 4 shows the ROCs for the 2nd moment table detection, with and without median filtering the 2nd moment feature.

3 Accessing specific geometry

The previous section has shown how a generic method can be used for parsing based on geometry. Having located and identified shots using that idea, it is interesting to note that context can again be exploited to calibrate the court view, again *without* 3D information. For snooker, this idea can be used to localise pot locations and hole spots; for tennis, the court boundaries and position of the net can be discovered. Here we present results for snooker.

In the present application of snooker, we are principally interested in the "full-table" view (figure 1) as far as semantic summarisation is concerned. Here, the Hough transform is again used, applied to those shots that have been char-

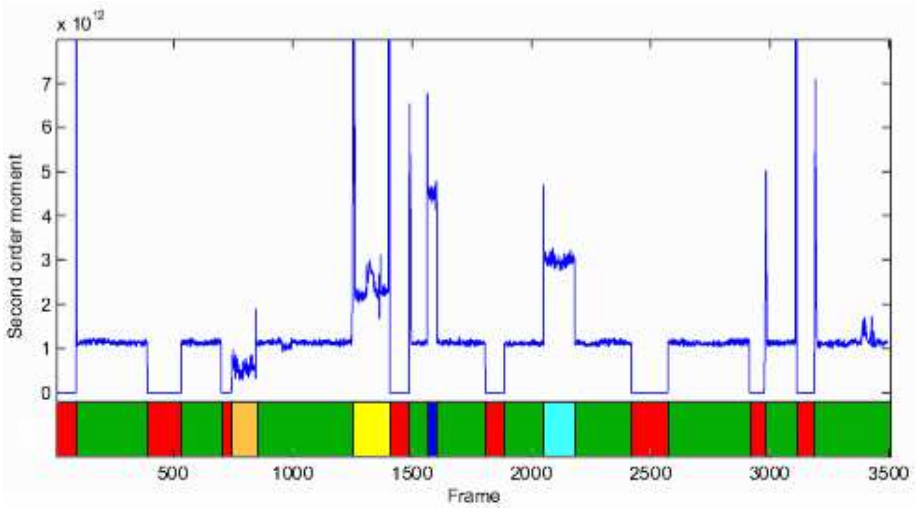


Fig. 3. Illustrating the effectiveness of the new Moment feature. The lowest second order moment values (red) signify crowd/player shots. Green indicates a full table shot (Moment $\approx 0.1 \times 10^{13}$). Editing effects such as wipes and dissolves yield impulses in the Moment. The yellow segment (frame 1300) corresponds to a shot from the black (bottom right) pocket, orange (frame 800) corresponds to a close up of the player leaning over the table and the blue (frame 1580) and light blue (frame 280) are shots over the yellow (top left) and green (top right) pockets respectively.

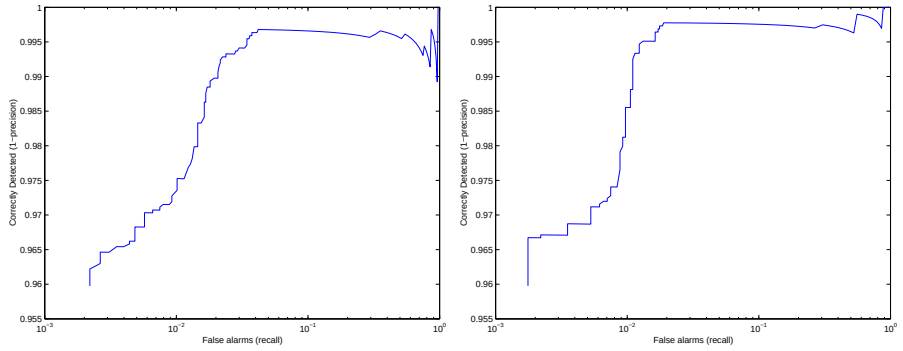


Fig. 4. ROCs for the 2nd moment table detection. At left, ROC without median filtering of 2nd moment values. On the right, ROC with median filter eliminates spikes due to erroneous table identification. Note the smoother curve. The curve is contained mainly in the top left corner indicating good performance.

acterised as full-table footage. The transform here is restricted to values of θ in the ranges [3..25], [89..90], and [155..177], corresponding to the regions of parameter space where the table edges are likely to be found; restricting θ in this way increases computation speed considerably and also allows fine granularity in parameter space. Lines in the configuration corresponding to a snooker table create a characteristic pattern in the transform space, as shown in figure 5): one peak each in the ranges $\theta \in [3..25]$, and $\theta \in [155..177]$, and two in the range $\theta \in [89..90]$. These four peaks correspond to the four edges of the table.

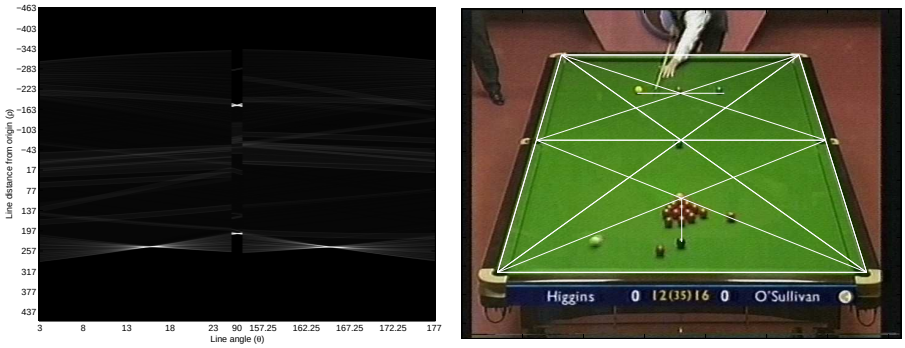


Fig. 5. Extracting geometry. Left to right: Restricted parameter Hough Transform of the full table shot; Table geometry, spots and pockets recovered.

Once the edges of the table have been detected, the table geometry can be inferred, i.e. the locations of the ball spots and the pockets. As the table is

distorted by a perspective projection resulting from the camera angle in full-table footage, attempting the inverse perspective transform is an obvious candidate. However, this is not practical: the inverse perspective transform is an ill-posed computation, especially when only one view is available. Sudhir, Jain, and Lee described in [6] a method that uses basis vectors, spanning the playing area, to find points of interest in the playing area.

Here we introduce the novel approach of using the fact that the intersection of the diagonals of a trapezoid are invariant under perspective distortion. Having located the table edges, therefore, we can calibrate the table surface *within the view geometry* without actually extracting the 3D geometry of the scene. Once the corner pockets have been found, the table dimensions assist in locating the other points of interest.

Because the diagonals of a trapezoid intersect at the midpoint of the trapezoid, irrespective of perspective transformation; the table can be repeatedly subdivided vertically to find any vertical position. For example, the black spot is 320 mm up from the bottom edge of the table, whose total length is 3500mm. 320/3500 expressed in binary is approximately 0.0001011101, so $320 = 3500 * (2^{-4} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-10})$. Each of these negative powers of two can be found by division along the vertical dimension of the table. In practice, the position of the black ball is adequately approximated with 0.00011, or 3/32 along the vertical. Repeated subdivision along the vertical yields the positions 1/8 and 1/16 of the way up the table; subdivision between these two points yields the black spot approximation, 3/32 of the way up the table. The perspective projection does not affect horizontal proportion, so to localise horizontal position it is sufficient to use the table dimensions directly. Most ball spots are exactly half-way between the table edges. Figure 5 shows the full table view with all relevant spots and pockets recovered.

4 Event detection

To effectively detect meaningful events in a sports media stream, context must again be exploited. In snooker, the principal event is ball potting, which corresponds to the disappearance of a ball within a region of interest. Our approach here relies on motion detection, which is complicated by the motion of the player when leaning over the table. A useful precursor to event detection, therefore, is object suppression or removal, which we consider below specifically for player removal in snooker. We then consider object disappearance as a means for event detection, which is a more generally applicable technique.

4.1 Player Masking

Our approach to player masking relies on a robust segmentation based on thresholding, followed by the detection of large, non-table regions which touch the edge of the table. These regions are assumed to be the player.

The initial segmentation consists of the conjunction of three thresholding operations: $B < dark\_threshold$, $B > bright\_threshold$, and $(Green - Red) > green\_threshold$), where B is the intensity component of the table region and $Green$ and Red are the colour channels of the table region. The resulting map is then dilated to ensure that nearby regions are connected. Any region in this dilated map that touch the edge of the table is considered to be the player. The mask corresponding to this region is stored for use in further processing.

The principal feature distinguishing a player region from other non-table objects is its proximity to the edge of a table, thus the method is robust for a wide range of threshold values. In the results described in this paper, the values for the thresholds are 50 for the $dark\_threshold$, 160 for the $bright\_threshold$, and 25 for the $green\_threshold$; these have been found effective for footage captured from a variety of sources. It is not necessary for the algorithm to be exhaustively accurate, as subsequent processing stages are designed for robustness to minor inaccuracies in the player mask. We consider the accidental masking of a ball to be be a more costly error than failure to entirely mask the player, and this algorithm errs largely according to this bias.

As a demonstration of successful player masking, the area corresponding to the player can be filled with an artificial texture similar to that of the table. A region of the table with a low high-frequency component is found by filtering the table with a first-order differential filter. It is assumed that this region contains only empty table and is free of balls or holes. This region is then randomly sampled and the samples used to fill the player region. Successful masking is shown in figure 6.

Fig. 6. Player masking: the region where the player occludes the table has been detected and filled with a generated table-like texture.

4.2 Event Detection: Ball Potting

Here we consider event detection based on object disappearance. This can be used, for example, in snooker and pool, to detect ball potting, and in golf, to

detect a successful putt. We describe here a new approach for detecting the disappearance of objects from a scene, based on motion-detection monitoring of region(s) of interest, as applied to to snooker.

Two regions around each pocket are monitored (the pockets have been located as described previously); a small region and a larger, encompassing region. The small regions are 1/15 of the table width on a side, and the large regions 1/8 of the table width on a side; these sizes have chosen to be large enough such that fast moving balls will appear within the regions while in motion. We can detect when a ball enters a region being monitored, using a motion-detection process based on frame differences. The use of two areas for each hole allows simple monitoring of proximity of the ball to the hole over time, distinguishing between 'close to the hole' and 'very close to the hole'. The coarse temporal resolution afforded by two proximity levels enables event detection.

The necessity of using two regions for event detection becomes clear on consideration of the following possible scenarios. If a ball enters the large, surrounding area, then enters the smaller area near the hole, and then leaves both areas simultaneously, it has been potted. If it leaves the small area first, and then the large, it has bounced back from the cushion and has not been potted. If it enters the small area but does not leave, it has stopped moving in close proximity to the hole. These latter two possibilities are 'near miss' events, of interest in themselves in summary generation. If only one region were used, the ball would enter and disappear, and where the ball was moving at high speed, it would not be possible to determine whether it had disappeared into the hole or merely traversed the region.

We detect ingress of the ball into a monitored area by taking a snapshot of each area at the start of each clip, and comparing the areas in each successive frame to the corresponding snapshot using simple differencing. Motion due to the player occluding a monitored area is suppressed via player masking, and non-player regions with an absolute difference greater than a motion detection threshold are considered to be balls. The motion threshold currently in use is 30, which has worked consistently across a variety of footage. Plotting the number of motion pixels against time, for each of the large and small surrounding regions, yields traces with clear peaks corresponding to the ball entering and leaving the regions.

The event scenarios described above are used directly in the analysis of these peaks. The peak for the large region will invariably begin before the peak for the small region. If both peaks return to zero in the same frame, the ball has been potted. If the small peak returns to zero before the large peak, a near miss has occurred.

Where the player masking algorithm fails, spurious motion traces can be introduced, resulting in false peaks. The peak detection algorithm considers only peaks with a height of between 20 and 80 pixels: these values correspond to the motion sizes typically generated by balls. Furthermore, peaks of only one frame's duration ignored, as are returns to zero of only one frame's duration (an example

of this latter is shown in figure 8). These modifications give high recall, with some sacrifice of precision.

The event detection algorithm was applied to a 16-minute game of snooker, consisting of 24250 frames and including 14 pots. All pots were selected correctly, along with 5 false alarms, for 100% recall and 74% precision. The false alarms were caused by failures in the player masking algorithm. The same footage contained four near misses, of which three were found, with no false alarms: thus 75% recall and 100% precision for near miss detection. The missed 'near miss' event was again due to a malfunction of the player masking algorithm.

Figure 7 shows a trace in which a ball is potted, while a near miss is shown in figure 8. We are working to incorporate detection of the colour of the potted ball within this event detection framework. Event detection by this method can be integrated with the MHI summary image, described below.

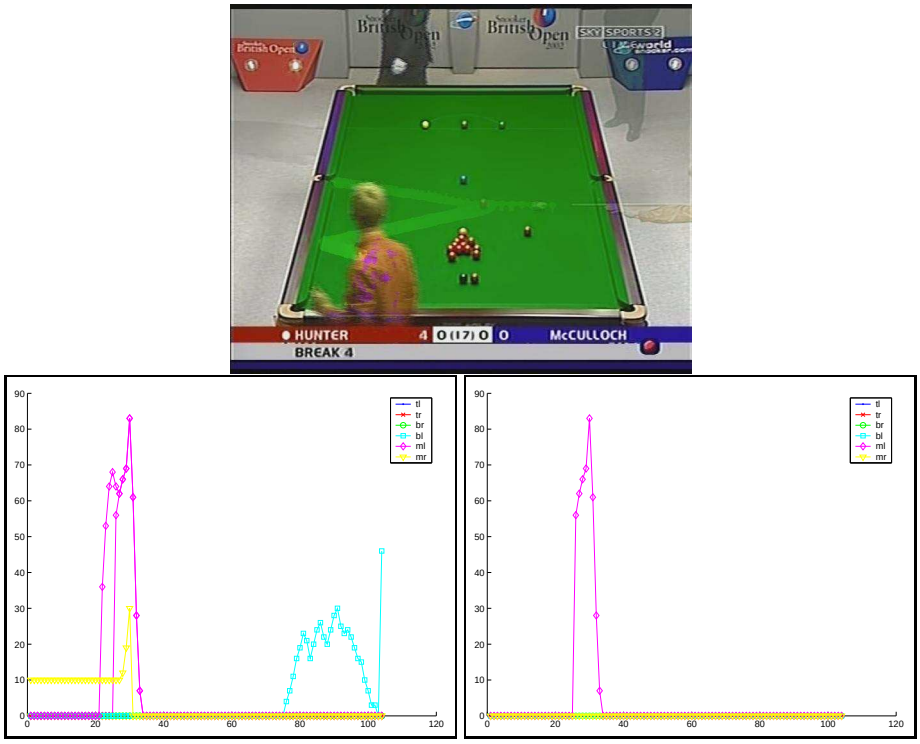


Fig. 7. Traces of the blue ball being potted into the left center pocket; the traces on the left are of the large regions, and those on the right represent the small regions. Traces for all six holes are shown; the ones of interest here are for the middle left (ml). The ball enters the large region in frame 20, and enters the small region in frame 25. It leaves both simultaneously in frame 34, indicating that the ball has been potted.

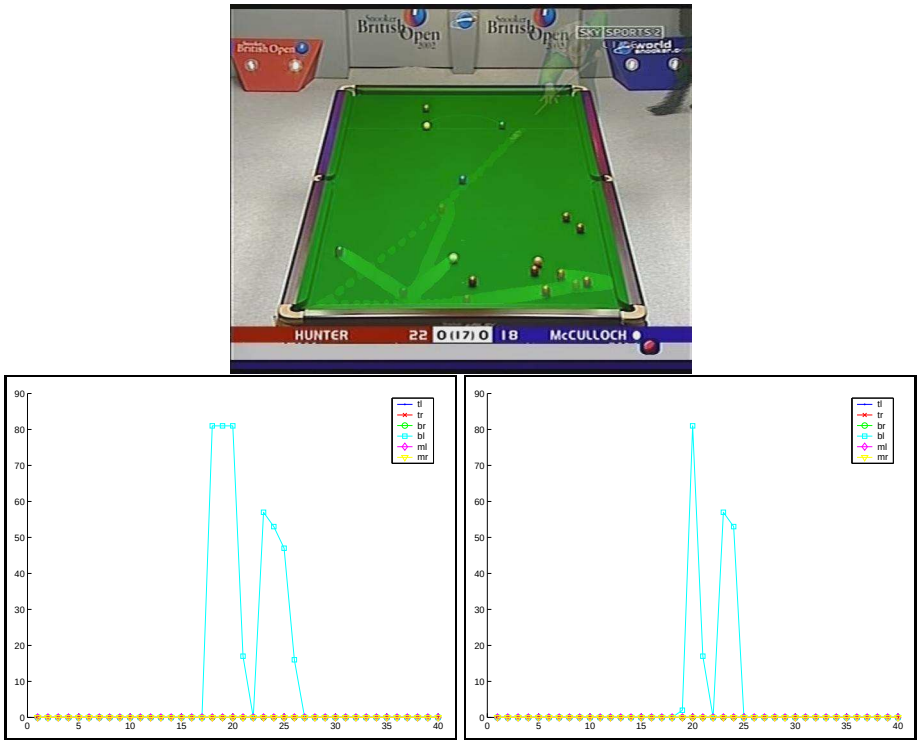


Fig. 8. Trace of a near miss, in which a red is almost potted to the bottom left pocket. The ball enters the large area in frame 17 and the small area in frame 18. It leaves the small area in frame 25, and then the large area in frame 27. A near miss is flagged. The brief return to zero is due to a the player masking algorithm, and is ignored.

5 Summarising Shot Activity

Having identified camera views or shots, summarisation is a typical next step. The semantic content of much sport video is principally conveyed by the motion in the footage. In presenting a summary of a sport to a user, it is imperative to consider how this information can be conveyed in as succinct a manner as possible. Assuming that a summary of an event can be presented as browsable key frames, each of which summarises the information in a shot, or in each *content rich* shot, then the problem is how to condense the useful information in a shot into a single image representation. We consider here implicit and explicit content summaries of motion. For implicit summaries a mechanism is required that summarises all useful motion in a shot and represents this in some intuitive manner. An explicit summary would include the specific trajectory of an important semantic object across the shot. In the example of snooker, the latter summary manifests as the exact path of a ball across the table and we explore the use of

a particle filter for tracking below. The former idea of implicit summaries is a new tool for sports summarisation and is discussed next.

5.1 Motion history images as summary frames

The Motion History Image (MHI) is a representation of the change due to motion over time in each image, first used by Bobick and Davis [8] for activity recognition. What is interesting is that these "motion templates" can be used for representing content itself.

The MHI is generated by

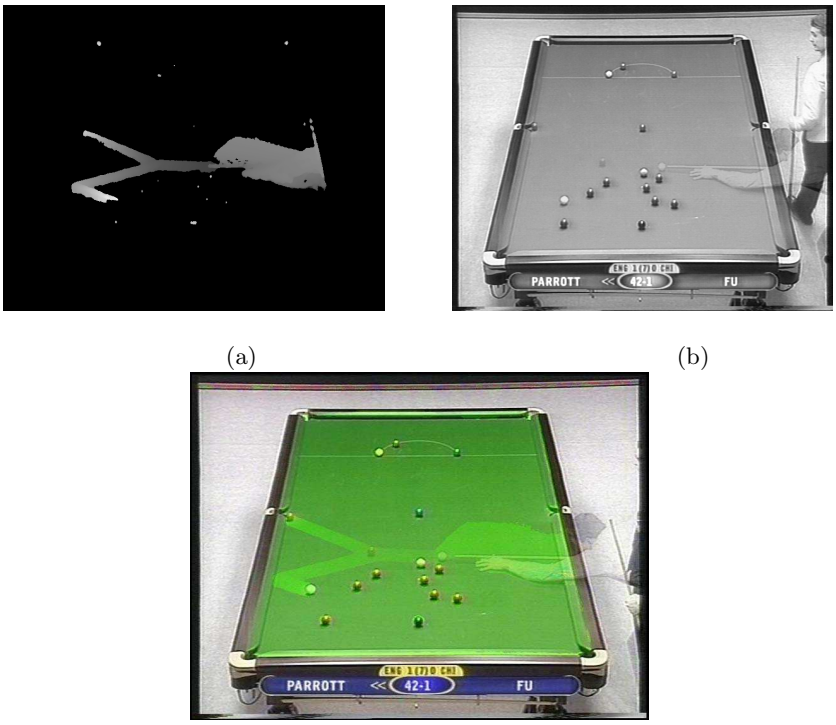
$$MHI(x,y,t) = \begin{cases} max\_motion & \text{if D(x,y,t)=1} \\ max(0, MHI(x,y,t-1)-1) & \text{otherwise} \end{cases} \quad (3)$$

where $D(x,y,t)$ is a binary image indicating where motion has been detected and max_motion is a large number which is decremented at each time step and scaled at the end of the motion to encompass the range 32-255. Thus MHI brightness corresponds to recent motion. A sample MHI is shown in figure 9 (a). Shot summaries can then be generated by averaging the first and last frames of the shot (figure 9 (b)) and overlaying the MHI on this composite, as shown in figure 9 (c).

However, generating a useful MHI for content representation must rely on some contextual information, since motion clutter can easily obscure the essence of a summary. In snooker (and many other court sports), the ball motion is important. This is detected here using simple frame differencing. The intensity field of successive frames are blurred with a Gaussian filter, and subtracted from each other. Motion regions are taken to be those regions where the absolute frame difference is greater than 30, whose size exceeds 10 pixels. The MHI generation is only switched on over frames in which the ball is moving. However, the motion of the player will also be flagged by this operation. In addition, for content representation, the start and end position of the player is sufficient to represent that motion. For these two reasons, the player motion is suppressed in the MHI generation using the player masking algorithm described previously. The importance of player masking for MHI summary generation is illustrated in figure 10.

MHI summaries are also generated of clips that show a close-up of a region of the table or a corner; these clips are detected using the 2nd order Hough feature classification described above. A drawback of our current implementation is that global motion in the sequence can be another source of motion clutter. We are working to incorporate global motion estimation and compensation of the MHI into this process to address this defect.

This tool is most effective in generating a summary of the entire game; for example, a 16 minute game of snooker was reduced to a 98 image summary, suitable for transmission to a mobile phone or other low-bandwidth device. An selection of images from this summary is shown in figure 13.



(a) (b)

(c)

Fig. 9. (a) The Motion History Image of a snooker clip. (b) First and last frames superimposed. (c) MHI overlayed on composite image.

(a) (b)

Fig. 10. Motion History Image summary generation. Player masking has not been used in figure (a), resulting in significant motion clutter. Figure (b) illustrates how suppressing the motion due to the player greatly improves the quality of the summary image.

5.2 Explicit motion extraction

The MHI is not suitable for machine analysis of exact object motion because it contains no explicit information regarding how many balls are moving, which balls are moving, whether balls have collided, the position of the balls at each moment in time, etc. In addition, it is conceivable that users may wish to query footage on the basis of trajectories. Thus, explicit trajectories are useful features. Explicit tracking is achieved here through a colour-based particle filter, based on that described by Blake and Isard in [9], and similar to that of Nummiaro *et al* [10].

The tracker is initialised at the MHI generation stage. As described above, the initial motion due to balls can be detected, independently of motion generated by the player with the use of player masking. This initial motion detection mask corresponds to the balls that are moving; the brightest region in this motion mask is then assumed to be the white ball. In our tests for 16 minutes of a snooker game, we were able to locate the initial position of the white ball with 100% accuracy using this method.

A target model of the ball's colour distribution is created in the first frame of the clip from this initial location. A HSB space colour histogram, of a circular region specified by the relative size of the ball in relation to the table size, is calculated. As a snooker table can be affected by luminance gradients due to non-uniform lighting conditions, the brightness component of the colour space was quantized to 32 bins. The colour histogram was therefore represented using (256x256x32 bins).

A collision between balls or a collision between a ball and the bottom cushion of the table may temporarily block the ball being tracked from view. Therefore, partial occlusion of the ball must be addressed, as described in [10]. Hence, pixels that are further away from the center of the object are assigned a lesser weight when calculating the colour distribution of the target and candidate models. This is also useful for avoiding the incorporation of the colour properties of the table into the ball model. The weighting function is given by

$$w(r) = 1 - \left(\frac{r_i}{\frac{5}{4} \underset{i}{max}(r_i)} \right) \tag{4}$$

where r_i is the L2 distance from the center of the ball to each pixel in the circular region.

The ball object is typically 100 pels in size. In order to facilitate an increase in resolution by selecting such a small object relative to the size of the image, and also accounting for changes in colour due to motion, which will corrupt the nature of the measured pdfs, the colour distribution needs to be extended for both the target and candidate models. This is achieved using a Parzen window [11] where σ is taken as the variance of the noise from the footage. The Parzen window is not applied to the brightness component as it has be quantized to 32 bins. The range of values is sufficiently small so as not to require a broader behaviour from that actually measured. The colour space is represented by $\Psi = \{H, S\}$.

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} \tag{5}$$

The colour distribution, $p = \{p^u\}_{u=0...m-1}$ of the object region, R, is therefore

$$p^u = c \sum_{x_i \in R} w\left(||\mathbf{x_c} - \mathbf{x_i}||\right) \phi\left(j - \Psi\left(\mathbf{x_i}\right)\right) \tag{6}$$

where c is a normalisation factor, $\mathbf{x_c}$ is the location of the center of the ball and $j = [0...m-1]$.

A Bhattacharyya distance measure is used to calculate the similarity between candidate histograms and the target and is used for weighting the sample set, $S = \{(s^n, \pi^n) | n = 1...N\}$, to locate the mean position of the ball. The likelihood attached to each sample is computed according to

$$\pi^n = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(1-\rho[p(s^n),q])}{2\sigma^2}} \tag{7}$$

where,

$$\rho\left[p, q\right] = \sum_{i=1}^{m} \sqrt{p^i q^i} \tag{8}$$

where p is the target model, q is the candidate and m is the number of bins.

As is typical, it is necessary to distinguish between correct tracking in the next frame and loss of 'lock'. This can be detected by using a threshold on the sum of the sample likelihoods, L, of the particle cloud. If $L > L_t$ a correct lock is assumed and the ball is deemed to have been found in this frame. The sample set is then dispersed according to a second order auto-regressive motion model with a stochastic component as shown below.

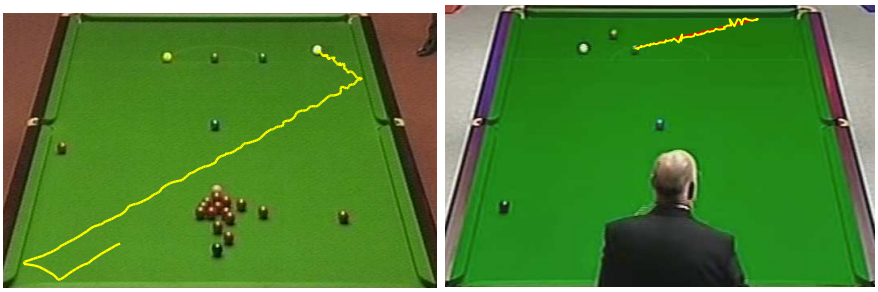
$$x_{t+1} = x_t + \left[\alpha\left(x_t - x_{t-1}\right)\right] + \left[\left(1 - \alpha\right)\left(x_{t-1} - x_{t-2}\right)\right] + \epsilon \tag{9}$$

where $\alpha = 0.7$, and $\epsilon \sim \mathcal{N}(0, 6)$. The same process is used for both horizontal and vertical directions.

If $L < L_t$, tracking is assumed to be lost and the sample with highest likelihood is used as a seed for the search on the next iteration. On every second iteration the sample distribution variance is incremented and on each alternate iteration the search space is decreased to the initial distribution, $\mathcal{N}(0, 6)$, in an attempt to locate the object. Here we use $L_t = 0.04$.

It is in the nature of the condensation algorithm that the modelled hue, saturation and brightness of the ball being tracked will lock on to a false object if the ball cannot be found—for example, if the ball has been potted. When this happens, a considerable change in one of these parameters will be observed. In the implementation considered here, a 50% change in the cumulative likelihood between the current and previous sample set is taken to indicate that the ball has been potted.

To assess the performance of the tracker, 10 balls (one of the white, red, black, blue, brown, pink or green balls) were individually tracked at different points in the game and in various locations on the table, in two of which were balls being potted. In total, this represented analysis over 300 frames, lasting approximately a second for each trajectory. Two performance measures were used. The first was the distance of the normal from the sampled points to the true trajectory of the ball. The second was the the angle between the true trajectory and the least squares fit to the data. The ground truth was found by manually locating the start and end locations of the ball across the trajectory. It was found that the mean angle between the true trajectory and the least squares fit was 0.7905° and the mean perpendicular distance from the samples to the true trajectory was 0.7305 pixels. Our tracker is therefore accurate to a sub-pixel level, and certainly good enough for quantitative summary information. Two examples of the tracking performance are shown in figure 11.



(a) (b)

Fig. 11. The particle filter for motion tracking, applied in (a) to the white ball and in (b) to the green ball. In (b), the true trajectory is shown in red.

As a preliminary example of the kind of semantic information that can be provided by explicit object tracking, see figure 12. Here, the bounces of the white ball can clearly be seen in the position and speed graphs, and the position of the ball at the time of each bounce is known. Because each bounce is close to the table edge, there is a high probability that the white ball has not hit another ball in this clip; had it hit another ball, a bounce and speed change would have been recorded at a position away from the edge. Furthermore, the speed decreases to zero in the clip, so it can be assumed that this clip represents the complete snooker shot. Thus, there is a higher than normal probability of this shot being a foul (as it is a foul in snooker not to strike a coloured ball with each shot). We are working to investigate further the reliability of these ideas.

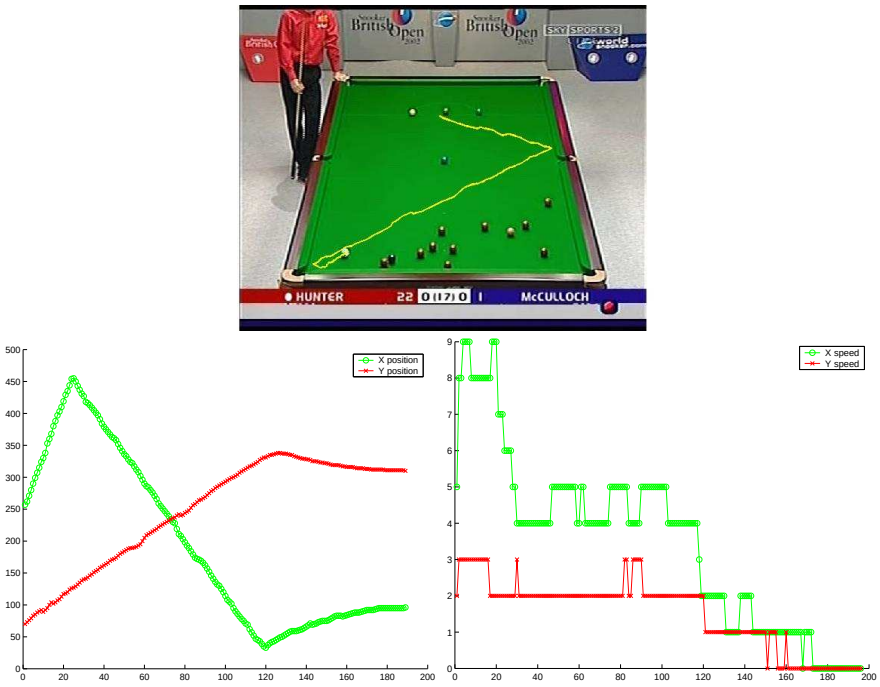


Fig. 12. An example of a foul, in which the white ball does not strike a colour. The two graphs are of white ball position and speed. Observe that bounces are clearly visible in the graphs.

6 Final Comments

Figure 13 shows an example portion of a complete summary of a game using the various tools above. The key frames are MHI frames, and in addition, where a pot has been detected the target hole has been highlighted. For one of the shots, a superimposed particle filter trajectory is shown. It is useful to note that manually generated summaries of sports in general simply do not exist in this form. Sports summaries are typically shortened video footage, and textual annotation can take 10 hours for 1 hour of summary generation [12]. Our future work revolves around attempting to assess the effectiveness of the kind of summaries that we have created. This is a difficult task.

The individual computational load of the tools presented here is not high. For instance, both our edge information generation and second order moment generation are of the order of $N_1 \times N_2$ operations per frame (where N_1, N_2 are the vertical and horizontal resolutions of the frame). Furthermore, as the techniques are designed for batch use, running entirely unattended, we have not yet undertaken performance optimised implementation of any of these tools; many of the tools are implemented in Matlab. Aside from player masking, the tools, run in cascade on a single 1 GHZ PIII, take about 3 hours to summarise 16 minutes of footage. Player masking introduces significant computational load; this can take up to 7 seconds per frame, principally because of its use of morphological operations such as dilation. We are investigating techniques to reduce the computational complexity of this stage.

The paper has introduced three new ideas for sports summarisation. The most important of these is that 3D information is not necessary to summarise footage according to geometry. Using in-image analysis the moment feature, hough transform analysis, and use of a-priori knowledge about diagonal intersection, allows a great deal of content rich information to be extracted from the footage. Our results show that we can successfully detect and cluster each camera view, detect balls being potted and create ball tracks. We have also presented a new concept in the generation of content rich summary key frames, through the use of the MHI. We expect that the moment feature and the use of the MHI in particular are applicable to a wide range of court sport events and we are currently working on specific adaptations of each tool to other domains.

References

1. Alberto Del Bimbo: Visual Information Retrieval. Morgan Kaufmann (1999)
2. Y. Deng, D. Mukherjee, B. S. Manjunath: Netra-V: toward an object-based video representation. SPIE International Conference on Storage and Retrieval for Image and Video Databases VI, Vol. 3312 (1997)
3. W. Xiong, J C.-M. Lee: Automatic dominant camera motion annotation for video retrieval. SPIE International Conference on Storage and Retrieval for Image and Video Databases VI, Vol. 3312 (1997)
4. Di Zhong and Shih-Fu Chang: Structure Analysis of Sports Video Using Domain Models. IEEE ICME (2001)

Fig. 13. MHI summary of 2421 frames of a snooker game. 779 frames of the total were considered relevant by the summary generation system, and are explicitly represented here. Event detection has been applied and ball pot events automatically highlighted in summary images 4 and 9 (a red circle is overlaid on the target hole). In image 4, the motion track of the white ball has also been presented. Summary image 7 illustrates the problem of global motion (here zoom) leading to motion clutter.

5. D.D. Saur, Y.-P. Tan, S.R. Kulkarni, and P.j. Ramadge: Automated analysis and annotation of basketball video. SPIE Storage and Retrieval for Still Image and Video Databases V, Vol.3022, pages 176-187. (1997).
6. G. Sudhir and John C. M. Lee and Anil K. Jain: Automatic Classification of Tennis Video for High-level Content-based Retrieval. International Workshop on Content-Based Access of Image and Video Databases (CAIVD'98) (1998)
7. R. Dahyot, N. Rea and A. C. Kokaram: Sport Video Shot Segmentation and Classification Visual Communications and Image Processing 2003, 8-11 July 2003, Univ. of Italian Switzerland (USI), Lugano, Switzerland. (2003)
8. A. Bobick and J. Davis: Real-time Recognition of Activity Using Temporal Templates. IEEE Workshop on Applications of Computer Vision, December 1996, pp. 39-42. (1996)
9. M. Isard, A. Blake: CONDENSATION – conditional density propagation for visual tracking. Int. J. Computer Vision, Vol. 29 (1998)
10. K. Nummiaro, E. Koller-Meier, L. Van Gool: An Adaptive Color-Based Particle Filter. Image and Vision Computing, Vol. 21, Issue 1, p.p 99-110 (2003)
11. C.M. Bishop: Neural Networks for Pattern Recognition. Oxford University Press, 1995.
12. W.J. Christmas, J. Kittler, D. Koubaroulis, B. Levienaise-Obadia, and K. Messer: Generation Of Semantic Cues For Sports Video Annotation. Oulu International Workshop on Image Retrieval, 2001, Finland (2001)
13. I. Cohen, A. Garg, T.S. Huang: Emotion Recognition from Facial Expressions using Multilevel HMM. Neural Information Processing Systems. (2000)
14. M. Petkovic, W. Jonker: Content-Based Video Retrieval by Integrating Spatio-Temporal and Stochastic Recognition of Events. IEEE Workshop on Detection and Recognition of Events in Video. (2001)
15. L. R. Rabiner and B. H. Juang: An introduction to hidden Markov models. IEEE ASSP Mag., pp 4–16, Jun. 1986

Appendix

As shown in Figure 3, the evolution of the feature vectors is closely related to the view in the image. If the temporal behaviour of the feature vector for each camera view can be modelled, it is then possible to categorise each shot. The Hidden Markov Model (HMM) allows a rich variety of temporal behaviours to be modelled. This approach has been used to good effect in cognition based systems [13, 14]. A two state ergodic HMM was found empirically to sufficiently model the feature vector for each of the camera view models. In order to generate an alphabet for the discrete HMM the features were quantised using K-means clustering algorithm.

Knowing the number of states and discrete codebook entries, a model λ, can be defined for each of the competing camera views. A succinct definition of a HMM is given by the following parameters, where K is the number of classes, N is the number of states and M is the number of unique observation symbols per state.

$$
\begin{aligned}
&A = \{a_{ij}\} = P(q_{t+1} = S_j | q_t = S_i), \ \ 1 \leq i, j \leq N \\
&B = \{b_j(O_t)\} = P(O_t = v_k | q_t = S_j), \ \ 1 \leq j \leq N, 1 \leq k \leq M \\
&\pi = \{\pi_i\} = P(q_1 = S_i), \ \ 1 \leq i \leq N \\
&\lambda_K = (A, B, \pi)
\end{aligned}
\tag{10}
$$

The parameters are defined as follows : A is a state transition probability matrix, B is an observation probability matrix or confusion matrix, in the discrete case, and π is a vector of initial state probabilities. An extensive tutorial on HMMs is available from Rabiner [15].

The parameter reestimation process was conducted by training the system with the ground truth. Half of the feature vectors from each camera view were used to train each model. The Baum-Welch algorithm was then used to find the maximum likelihood model parameters that best fit the training data. The models for the footage were then tested against the entire sequence. The shot cuts were detected offline using a combination of the gradient of the second order moment of the Hough transform and the sum of histogram differences of the luminance component of the image sequence. Given this observation sequence the probability $P(O|\lambda)$ can be calculated. Each camera view can then be classified by finding the model that results in the greatest likelihood of occurring according to:

$$
C = \arg \max_{1 \leq x \leq K} [P(O|\lambda_x)] \tag{11}
$$

The whole process is summarized in figure 14.

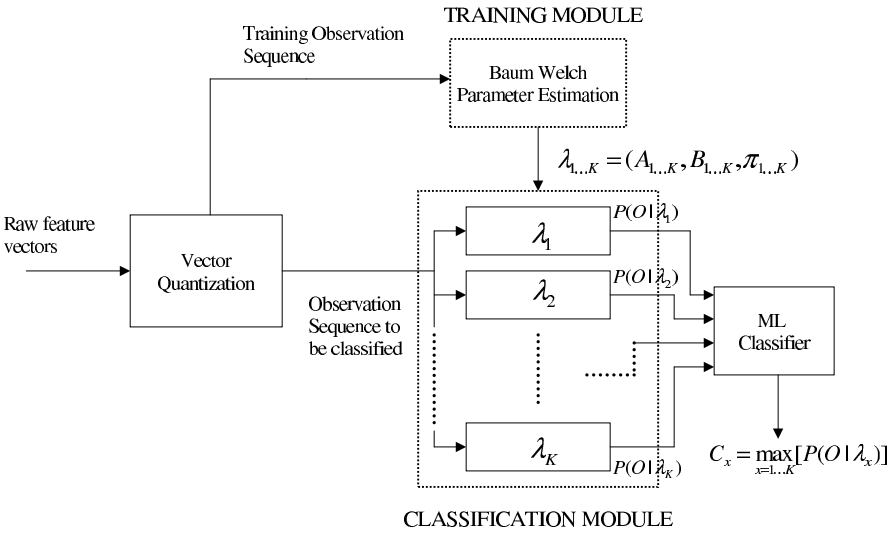


Fig. 14. ML-classifier.