Semantic Event Detection and Classification in Cricket Video Sequence

M. H. Kolekar, K. Palaniappan
Department of Computer Science, University of Missouri, Columbia, MO, USA
mkolekar@gmail.com, palaniappank@missouri.edu

S. Sengupta
Dept. of Electronics and ECE, Indian Institute of Technology, Kharagpur, WB, India
ssg@ece.iitkgp.ernet.in

Abstract

In this paper, we present a novel hierarchical framework and effective algorithms for cricket event detection and classification. The proposed scheme performs a top-down video event detection and classification using hierarchical tree which avoids shot detection and clustering. In the hierarchy, at level-1, we use audio features, to extract excitement clips from the cricket video. At level-2, we classify excitement clips into real-time and replay segments. At level-3, we classify these segments into field view and non-field view based on dominant grass color ratio. At level-4a, we classify field view into pitch-view, long-view, and boundary view using motion-mask. At level-4b, we classify non-field view into close-up and crowd using edge density feature. At level-5a, we classify close-ups into the three frequently occurring classes batsman, bowler/fielder, umpire using jersey color feature. At level-5b, we classify crowd segment into the two frequently occurring classes spectator and players' gathering using color feature. We show promising results, with correctly classified cricket events, enabling structural and temporal analysis, such as highlight extraction, and video skimming.

1 Introduction

As digital video becomes more pervasive, efficient way of mining the information from the video becomes an increasingly important. Video itself contains huge amount of data and complexity that makes the analysis very difficult. However, there are certain similarities among certain types of video, which can be cues to solve the problem. For example, a news video can be considered as a sequence of video segments which starts with an anchor person followed by story unit; a sports video as a repetition of play and breaks. As an important type of TV programs, sports video has been widely analyzed due to tremendous commercial potentials [6], [8].

With remarkable development in multimedia systems, many sports applications came into birth. The huge amount of data that is produced by digitizing sports videos demands a process of data filtration and reduction. The large number of sports TV broadcasts also creates a need among sports fans to have ability of seeing interesting parts of all these broadcasts, instead of watching all of them in their entirety. These needs were addressed by the applications such as video summarization and highlight event extraction.

Sports video analysis has received much attention in the area of digital video processing. Existing approaches can be broadly classified as genre-specific or genre-independent. Due to dramatically distinct broadcast styles for different sports genres, much of the prior art concerns genre specific approaches. Researchers have targeted the individual sports game such as soccer (football) [11], [1] tennis [14], cricket [7], basketball [12], volleyball [4], etc. These works show that genre specific approaches typically yield successful results within the targeted domain. In comparison with the genre-specific research work, less work is observed for genre-independent studies [5], [4]. For a specific sports event detection task, it is not feasible to expect a general solution that will work successfully across all genres of sports video.

Cricket is the most popular sport in the world after soccer. Cricket is played globally across 17 countries including India, Australia, England, Pakistan, Srilanka, South-Africa, New-Zealand, Bangladesh and West Indies. However, less research work [7], [10] has been reported on cricket in comparison with sports like soccer, tennis, basketball, baseball. The reasons are possibly the increased complexity of the game and especially the long duration for which highly efficient video pruning is required.

Most of the research in sports video processing [3], [9] assumes a temporal decomposition of video into its

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.102

structural units such as clip, scenes, shots and frames similar to other video domain including television and films. A group of sequential frames often based on single set of fixed or smoothly varying camera parameters (i.e. close-up, medium or long shots, dolly, pan, zoom, etc) form shot. A collection of related shots form scene. A series of related scenes form a sequence. A part of the sequence is called as clip. A video is composed of different story units such as shots, scenes, clips, and sequences arranged according to some logical structure defined by the screen play. In our work, we extract the clips and after analysis assign a descriptive label to each clip and refer the clip as event.

The main contributions of the paper are as follows: 1) We propose novel hierarchical framework for cricket video event detection and classification, 2) Our proposed field-view classification based on motion-mask is new and gives very good classification accuracy, 3) We propose novel close-up detection algorithm based on edge density feature. We proposed novel domain specific algorithm based on jersey color features for close-up classification and crowd classification. The rest of the paper is organized as follows. Section-2 presents proposed hierarchical classification tree. Section-3 presents experimental results. Section-4 concludes the paper with direction for future work.

2 Hierarchical classification

In [11] the authors integrate multiple features to classify video sequences. Although the integration of multiple features improves the classification accuracy, it leads to other problems such as proper selection of features, proper fusion and synchronization of right modalities, critical choice of the weighting factor for the features and computational burden. To cope with these problems, we propose a novel hierarchical classification framework for the cricket videos as shown in Figure 1, which has the following advantages: (1) The approach avoids shot detection and clustering that are the necessary steps in most of video classification schemes, so that the classification performance is improved. (2) The approach uses top-down four-level hierarchical method so that each level can use simple features to classify the videos. (3) This improves the computation speed, since the number of frames to be processed will remarkably reduce level by level.

2.1 Level-1:Excitement Detection

We have observed that spectator's cheer and commentator's speech become louder, during the exciting events. Based on this observation, we have used two popular audio content analysis techniques- short-time audio energy and zero crossing rate (ZCR) for extracting excitement clip. We are considering the short-time as the number of audio samples corresponding to one video frames. A particular video frame is considered as an excitement frame if the product of its audio excitement and ZCR exceeds a certain threshold. After computing short-time audio energy $E(n)$ and ZCR $Z(n)$, We propose following steps for excitement clip detection.

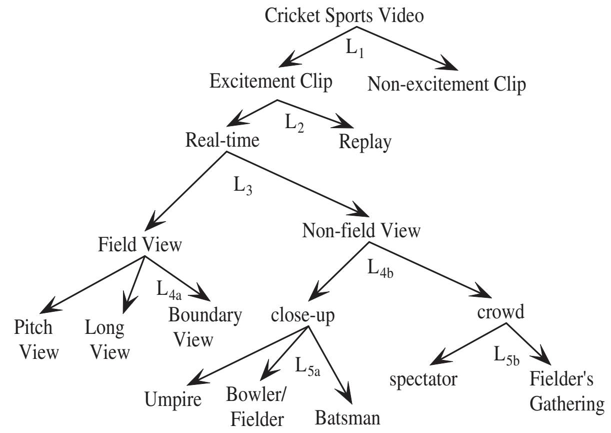

Figure 1. Tree Diagram of Hierarchical Framework

Algorithm-1:

1: Short-time audio energy

It is defined as

$$E(n) = \frac{1}{V} \sum_{m=0}^{V-1} [x(m)w(n-m)]^2 \tag{1}$$

where,

$$w(m) = \begin{cases} 1 & \text{if} \quad 0 \le m \le V-1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$x(m)$ is the discrete time audio signal, V is the number of audio samples corresponding to one video frame.

2: Short-time zero-crossing rate

In discrete-time signals, a zero crossing is said to occur if successive samples have different signs. The short-time average zero-crossing rate $Z(n)$, as defined below, gives rough estimates of spectral properties of audio signals.

$$Z(n) = \frac{1}{2} \sum_{m=0}^{V-1} |sgn[x(m)] - sgn[x(m-1)]| w(n-m) \tag{3}$$

where,

$$sgn[x(m)] = \begin{cases} 1 & x(m) \ge 0 \\ -1 & x(m) < 0 \end{cases} \tag{4}$$

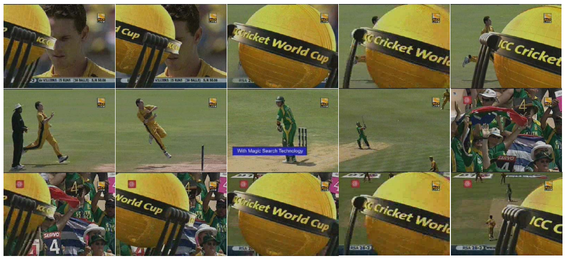

Figure 2. Representative frames of the replay segment. Row-1: Representative frames of flying graphics(#899 to #914), Row 2: Representative frames of replay (#915 to #1381), Row-3: Representative frames of flying graphics (#1382 to #1397)

where, and $w(m)$ is a rectangular window. It is observed that commentary corresponding to exciting moments give rise to high ZCR.

3: Averaging through sliding window

To distinguish genuine audio excitement from audio noise, we have used a sliding window. However, it helps for early detection of the events as well.

$$E_1(n) = \frac{1}{L}\sum_{l=0}^{L-1} E(n+l) \ \ and \ \ Z_1(n) = \frac{1}{L}\sum_{l=0}^{L-1} Z(n+l) \tag{5}$$

where, L is the length of sliding window.

The normalized values are as follows:

$$E_2(n) = \frac{E_1(n)}{\max_{1\le i\le N} E_1(i)} \ \ and \ \ Z_2(n) = \frac{Z_1(n)}{\max_{1\le i\le N} Z_1(i)} \tag{6}$$

where, N is the total number of video frames.

4: Excitement frame detection

The product $P(n)$ is given as

$$P(n) = E_2(n) * Z_2(n) \tag{7}$$

Based on the the product term $P(n)$, a video frame n will be finally labeled as $\psi(n) \in [0,1]$ as defined below:

$$\psi(n) = \begin{cases} 1 & (excitement) \ \ P(n) \ge \mu_p \\ 0 & (non-excitement) \ \ otherwise \end{cases} \tag{8}$$

where, μ_p is the mean of $P(n)$.

5: Excitement clip detection

To distinguish genuine audio excitement from audio noise, we select the excitement clips of duration greater than 10 seconds.

2.2 Level-2: Replay Detection

As shown in figure 2 we observed that replays are generally broadcasted with flying graphics (logo-transition) indicating the start and end of the replay. The flying graphics generally last for 10 to 20 frames. Replay segment is sandwiched by two logo-transitions. Since a replay shows many different viewpoints and thus contains many shots in a relatively short period, shot frequency in a replay segment is significantly higher than the average shot frequency in the excitement clip. Based on these observations, we propose the following algorithm for replay detection.

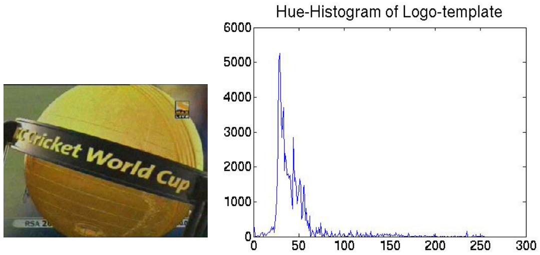

Figure 3. (a) Logo template (b) Hue-Histogram of Logo-template

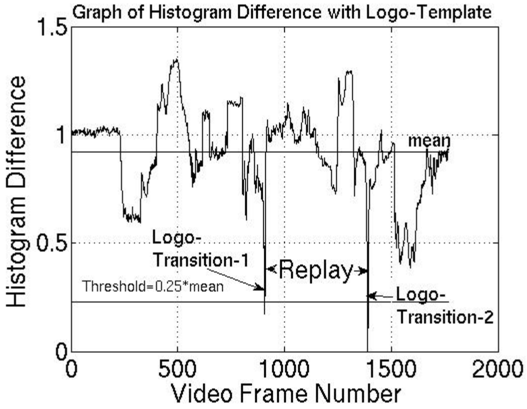

Figure 4. Graph of Hue-histogram difference vs video frame number for the video containing replay segment shown in figure 2

Algorithm-2:

1: Convert the input RGB frame into HSV image format and plot 256-bins Hue-histogram.

2: Compute hue-histogram of the logo-template as shown in figure 3.

2: Let M by N be the size of the frame. Compute Hue-Histogram Difference (HHD) between frame n and the logo-template using the following formula:

$$HHD(n) = \frac{1}{M*N}\sum_{b=1}^{B} |I_n(b) - L(b)| \tag{9}$$

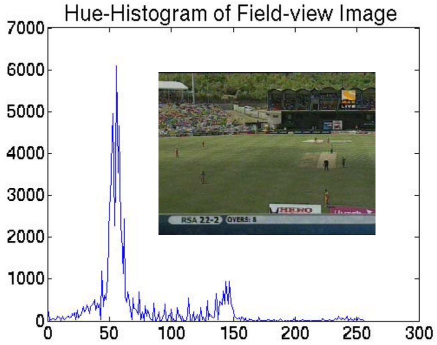

Figure 5. Hue-histogram of field view image

where B=total number of bins.
3: Plot the graph of HHD vs frame number as shown in Figure 4. Select the frames which has $HHD < threshold$. In our case we have selected $threshold = 0.25 * mean$. These frames will be logo-transition frames.
4: Select the segment between two successive logo-transitions and compute shot frequency f_{sr}.
5: Compute average shot frequency $\overline{f_{sc}}$ for the excitement clip.
6: The selected segment is classified as replay segment using following condition.

```
if ($f_{sr} > \overline{f_{sc}}$),
then selected segment belongs to replay class
else selected segment belongs to real-time class
```

2.3 Level-3: Field View Detection

At level-3, we are using grass-pixel ratio similar to [13] to classify the real-time clips into field view and non-field view. In our experimental set-up, we consider 60 field view images in hsv format for training. We plot 256-bin histogram of the hue component of these images. We pick up the peaks of hue histogram of these images. As shown in Figure 5, we observed peak at bin $k = 56$ and value of the peak is 6092 for the particular image of size 240 × 320. By testing all 60 images, we observed that the green color peak occurs between bin $k = 52$ to $k = 62$. The peak of the histogram gives number of the pixels of the grass in the image. We call this number as x_g. From this, we compute the dominant grass pixel ratio ($DGPR$) as x_g/x, where x is the total number of pixels in the frame. We observed $DGPR$ values vary from 0.16 to 0.24 for the field view. For non-field view image shown in Figure 6, we observed peak belongs to the bins other than $k = 52$ to $k = 62$ and $DGPR$ value is very small.

Algorithm-3:
1: Convert the input RGB frame into HSV image format and plot 256-bins Hue-histogram.
2: Compute $DGPR$ by observing the peak between bins $k = 52$ to $k = 62$.
3: The image can be classified as field view or non-field view by using following condition:

```
if ($DGPR > 0.07$),
then frame belongs to class field view
else frame belongs to class non-field view
```

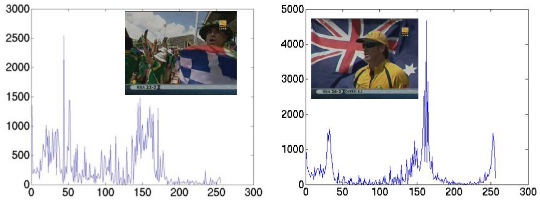

Figure 6. (a) Hue-Histogram of non-field view image (crowd), (b)Hue-Histogram of non-field view image (close-up)

2.4 Level-4a: Field View Classification

Under the constant illumination model, the optic-flow equation [2] of a spatiotemporal image volume $\mathbf{I}(\mathbf{x})$ centered at location $\mathbf{x} = [x, y, t]$ is given by Eq. 10 where, $\mathbf{v}(\mathbf{x}) = [v_x, v_y, v_t]$ is the optic-flow vector at $\mathbf{x}$,

$$\begin{aligned} \frac{d\mathbf{I}(\mathbf{x})}{dt} &= \frac{\partial \mathbf{I}(\mathbf{x})}{\partial x} v_x + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial y} v_y + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial t} v_t \\ &= \nabla \mathbf{I}^T(\mathbf{x})\, \mathbf{v}(\mathbf{x}) = 0 \end{aligned} \tag{10}$$

and $\mathbf{v}(\mathbf{x})$ is estimated by minimizing Eq. 10 over a local 3D image patch $\mathbf{\Omega}(\mathbf{x}, \mathbf{y})$, centered at $\mathbf{x}$.
In order to reliably detect only the moving structures *without* performing expensive eigenvalue decompositions, the concept of the *flux tensor* is proposed [2]. Flux tensor is the temporal variations of the optical flow field within the local 3D spatiotemporal volume.
Computing the second derivative of Eq. 10 with respect to t, Eq. 11 is obtained where, $\mathbf{a}(\mathbf{x}) = [a_x, a_y, a_t]$ is the acceleration of the image brightness located at $\mathbf{x}$.

$$\begin{aligned} \frac{\partial}{\partial t}\left(\frac{d\mathbf{I}(\mathbf{x})}{dt}\right) &= \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial x \partial t} v_x + \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial y \partial t} v_y + \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial t^2} v_t \\ &+ \frac{\partial \mathbf{I}(\mathbf{x})}{\partial x} a_x + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial y} a_y + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial t} a_t \end{aligned} \tag{11}$$

which can be written in vector notation as,

$$\frac{\partial}{\partial t}(\nabla \mathbf{I}^T(x)\mathbf{v}(\mathbf{x})) = \frac{\partial \nabla \mathbf{I}^T(\mathbf{x})}{\partial t}\, \mathbf{v}(\mathbf{x}) + \nabla \mathbf{I}^T(\mathbf{x})\, \mathbf{a}(\mathbf{x}) \tag{12}$$

Using the same approach for deriving the classic 3D structure, minimizing Eq. 11 assuming a constant velocity model and subject to the normalization constraint $||\mathbf{v}(\mathbf{x})|| = 1$ leads to Eq. 13,

$$e_{ls}^{F}(\mathbf{x}) = \int_{\mathbf{\Omega}(\mathbf{x},\mathbf{y})} \left(\frac{\partial(\nabla \mathbf{I}^{T}(\mathbf{y})}{\partial t}\, \mathbf{v}(\mathbf{x}) \right)^{2} W(\mathbf{x},\mathbf{y})\, d\mathbf{y} + \lambda \left(1 - \mathbf{v}(\mathbf{x})^{T} \mathbf{v}(\mathbf{x}) \right) \quad (13)$$

Assuming a constant velocity model in the neighborhood $\mathbf{\Omega}(\mathbf{x},\mathbf{y})$, results in the acceleration experienced by the brightness pattern in the neighborhood $\mathbf{\Omega}(\mathbf{x},\mathbf{y})$ to be zero at every pixel. The 3D flux tensor $\mathbf{J_F}$ using Eq. 13 can be written as

$$\mathbf{J_F}(\mathbf{x},\mathbf{W}) = \int_{\mathbf{\Omega}} W(\mathbf{x},\mathbf{y}) \frac{\partial}{\partial t} \nabla \mathbf{I}(\mathbf{x}) \cdot \frac{\partial}{\partial t} \nabla \mathbf{I^T}(\mathbf{x}) d\mathbf{y} \quad (14)$$

and in expanded matrix form as Eq. 15.

$$\mathbf{J_F} = \begin{bmatrix} \int_{\mathbf{\Omega}} \left\{ \frac{\partial^2 \mathbf{I}}{\partial x \partial t} \right\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial x \partial t} \frac{\partial^2 \mathbf{I}}{\partial y \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial x \partial t} \frac{\partial^2 \mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial y \partial t} \frac{\partial^2 \mathbf{I}}{\partial x \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \left\{ \frac{\partial^2 \mathbf{I}}{\partial y \partial t} \right\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial y \partial t} \frac{\partial^2 \mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial t^2} \frac{\partial^2 \mathbf{I}}{\partial x \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial t^2} \frac{\partial^2 \mathbf{I}}{\partial y \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \left\{ \frac{\partial^2 \mathbf{I}}{\partial t^2} \right\}^2 d\mathbf{y} \end{bmatrix} \quad (15)$$

As seen from Eq. 15, the elements of the flux tensor incorporate information about temporal gradient changes which leads to efficient discrimination between stationary and moving image features. Thus the trace of the flux tensor matrix which can be compactly written and computed as,

$$\mathbf{trace}(\mathbf{J_F}) = \int_{\mathbf{\Omega}} ||\frac{\partial}{\partial t} \nabla \mathbf{I}||^2 d\mathbf{y} \quad (16)$$

and can be directly used to classify moving and non-moving regions without the need for expensive eigenvalue decompositions. Motion-mask is obtained by thresholding and post-processing averaged flux tensor trace. Post-processing include morphological operations to join fragmented objects and to fill holes.

In field view, players and crowd are moving objects and field is non-moving object. Hence, we used motion-mask to classify the frames of the field view as long view, straight view and corner view. Our approach is summarized as follows:

Algorithm-4a:

1: Generate motion-mask for the input field-view frame as shown in the second column of the Figure 7.

2: Apply connected component technique to remove noisy objects from the image as shown in the third column of figure 7.

3: In the connected component image, background color is the color of object 'field'. Divide the frame into three regions 11, 12, and 2 as shown in the figure 7(a).

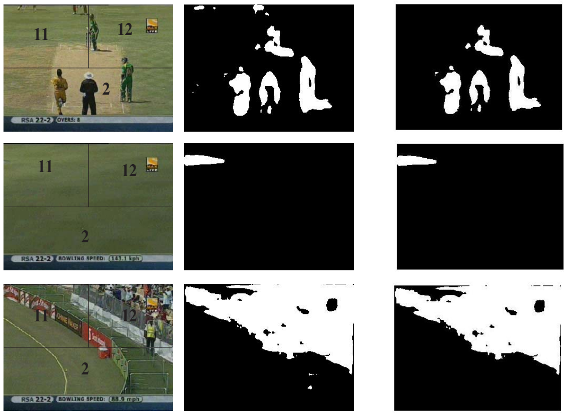

Figure 7. Row-1 shows pitch view: (a) Image (b) motion-mask (c) connected component image, Row-2 shows long view: (d) Image (e) motion-mask (f) connected component image, Row-3 shows boundary view: (g) Image (h) motion-mask (i) connected component image

4: Let FP_2, FP_{11}, FP_{12} be the percentage of field pixels in the region 2, 11, 12 of the connected component image respectively. Let T_1, T_2, T_3 be the thresholds. The field-view frame is classified into long view, corner view, and straight view using following condition:

```
if $(FP_2 > T_1) \bigwedge ((FP_{11} + FP_{12}) > T_2)$,
then frame belongs to class long-view
else if $|FP_{11} - FP_{12}| > T_3$
frame belongs to class boundary-view
else
frame belongs to class pitch-view
```

2.5 Level-4b: Close-up detection

At this level, we have to classify the frames of non-field views. We observed that non-field view generally contains only close-up and crowd frames. Hence we are classifying the non-field views broadly into close-up and crowd classes. We have proposed the feature based on the percentage of edge pixels (EP) to classify the frame into crowd or close-up, since the edginess is more for crowd frame as shown in figure 8. Our approach is summarized as follows:

Algorithm-4b:

1:Convert input RGB image into YC_bC_r image format.

2: Apply *Canny* operator to detect the edge pixels.

3: Count the percentage of edge pixels (EP) in the image.

4: Classify the image using following condition:

```
if $(EP > T_4)$,
```

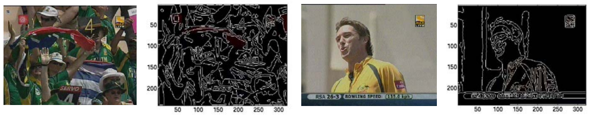

Figure 8. (a) Crowd Image, (b) Edge detection results of image (a), (c) Close-up Image, (d) Edge Detection results of image (c)

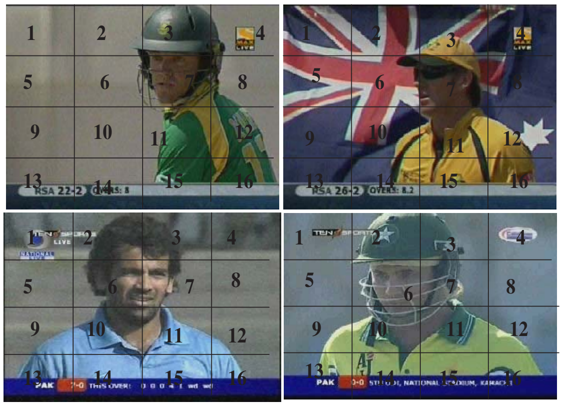

Figure 9. Close-up frames frequently observed in broadcasted cricket video. Row-1: (a)Batsman, (b) Fielder, Row-2: (c) Fielder, (d) Batsman

then *frame belongs to class crowd*
else *frame belongs to class close-up*

2.6 Level-5a:Close up Classification

In this level, we will classify the close-up images into batsman, bowler/fielder, umpire. We will first find out the location of the face of the person in the close-up using skin or hair color information. In most of the close-up images, the face position will occur in the block 6,7,10,11, as shown in Figure 9. Depending on the face location, we will select the block for checking the jersey color of the player. Our approach is summarized as follows:

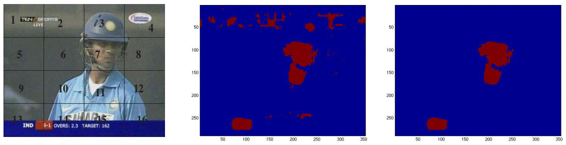

Figure 10. (a) Image of batsman (b) Image showing skin detection, (c) Connected component image

Table 1. Condition for block selection for checking jersey color of the player

x	Condition-x	Result-x (block number for jersey color checking)
1	$(S_6, S_7, S_{10}, S_{11}) > T_5$	14, 15
2	$(S_6, S_{10}) > T_5$	14
3	$(S_7, S_{11}) > T_5$	15
4	$(S_6, S_7) > T_5$	10, 11
5	$(S_{10}, S_{11}) > T_5$	14, 15
6	$S_{11} > T_5$	15
7	$S_{10} > T_5$	14
8	$S_6 > T_5$	10
9	$S_7 > T_5$	11

Algorithm-5a:

1: Convert input RGB image into YC_bC_r image format. Use the following condition for detecting skin pixels.

if $(105 < Y < 117) \wedge (110 < C_b < 113) \wedge (C_r > 128)$,
then *pixel belongs to skin color*
else *pixel does not belong to skin color*

2: Apply connected component technique to remove noisy skin detected objects from the image as shown in Figure 10.

3: Divide the image into 16 blocks as shown in figure 10 (a), and compute the percentage of skin color pixels in each block.

4: Let S_6, S_7, S_{10}, S_{11} are the skin percentages of block number $6, 7, 10, 11$ respectively. Select threshold T_5 for considering the block as skin block.

5: If no skin color found in the block $6, 7, 10, 11$, use threshold T_5 for checking hair color (black).

6: Check the condition-x of the Table 1 using skin or hair color block information. If condition-x is satisfied, result-x will give the block number for checking jersey color of the player.

7: Compute 256-bins Hue-histogram $Histo_b$ of the selected block as per condition-x.

8: Compute the average 256-bins Hue-histogram for all the known classes C.

9: Compute the Euclidean distance of the block b of frame n from the class k using following formula.

$$d_k(b) = \sqrt{\sum_{i=1}^{256}[histo_k(i) - histo_b(i)]^2} \quad for\ k = 1, 2, .., C \tag{17}$$

10: Select the value of k for which d_k has lowest value. Assign that value of k as a class-label to the particular frame n.

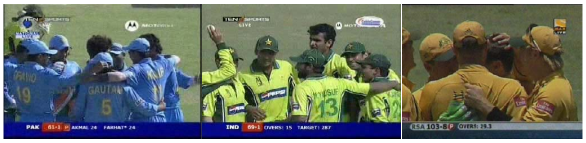

Figure 11. Players gathering of various teams (a) Players gathering of India, (b) Players gathering of Pakistan, (c) Players gathering of Australia

2.7 Level-5b:Crowd Classification

At this level, we classify the crowd using jersey color information into the following two classes: Fielder's Gathering and Spectator. Since fielders gather on the field after exciting event, in most of the fielders gathering frames will have field as a background as shown in figure 11. Hence, if we set the dominant field color (green) bins to zero, classification performance will be improved. Our approach is summarized as follows:

Algorithm-5b:
1:Convert input RGB image n into HSV format.
2: Compute 256-bin Hue-histogram for the input image n.
3: Dominant green color occur in the bins 56 to 64. So if we set these bins to zero, the error due to field color will be removed. Let $histo_n$ be the histogram of image n after setting the bins of green color to zero.
4: For all known classes, compute average 256-bins hue-histogram with setting dominant green color bins (i. e. 56 to 64) to zero.
5: Compute histogram distance of the hue-histogram of input image n from the hue-histogram of the class k using following formula.

$$d_k(n) = \sqrt{\sum_{i=1}^{256}[histo_k(i) - histo_n(i)]^2} \quad for\ k = 1, 2, ., C \tag{18}$$

6: Find the value of k for which the distance $d_k(n)$ is minimum. Assign label of that class k to the frame n.

2.8 Event

We have defined the events as scenes in the video with some semantic meaning (i.e. labels from a semantic hierarchy) attached to it based on the leaf nodes shown in Figure 1. Events are extracted as the leaf nodes of the level-2 to 5 of hierarchical tree. The events are replay, batsman, bowler/fielder, spectator, fielder's gathering, pitch-view, long-view, boundary-view.

Table 2. Cricket Videos used for testing

ID	Name of the Match	A Vs B	Date
V$_1$	Videocon Cup	India vs Australia	23/8/2004
V$_2$	Hutch Cup	India vs Pakistan	18/2/2006
V$_3$	Hutch Cup	India vs Pakistan	16/2/2006
V$_4$	World Cup	Australia vs South Africa	25/4/2007

Table 3. Performance of Level-1 of hierarchical Classifier

ID	Total Duration	Extracted Clips Duration	$N_c/N_m/N_f$	Recall (%)	Precision (%)
V_1	90 min	26 min	41/5/8	89.13	83.67
V_2	364 min	197 min	139/24/18	85.28	88.54
V_3	256 min	92 min	127/21/18	85.81	87.59
V_4	272 min	108 min	133/19/20	87.50	86.93

3 Experimental Results

We define the threshold vector as $T = [T_1, T_2, T_3, T_4, T_5]$. We experimentally found that $T = [65, 65, 10, 8, 60]$ gives better results. We are extracting excitement clips at level-1 and from level-2 to level-5, we are analyzing the clips to extract the event sequence. We present clip-based performance for classifiers of level-2 to level-5. The length of the clips decreases as the level of hierarchy increases, because at each level, we divide the clips into sub-clips. For measuring the performance of classifiers at each level, we use following parameters:

$$Recall = \frac{N_c}{N_c + N_m} \quad and \quad Precision = \frac{N_c}{N_c + N_f}$$

Where, N_c, N_m, N_f represents the number of clips correctly detected, missed and false positive, respectively.

The overall performance of the classifier at level-1 is shown in table 3. In case of poor broadcasting quality and noisy audio, performance of audio-based excitement clip extraction decreases. The performance of the classifiers of level-2 to 5 are presented in table 4. At level-2, we observed 84.21 % recall and 90.81 % precision for replay detection. Replays generally occur at the end of the excitement clips. If audio is low during the last logo-transition of the replay, it will not be extracted as a part of the excitement clip, and hence there will be possibility of missing replays. For field view detection, we observed 94.88 % recall and 95.16 % precision. For field view classification, we used motion mask based approach. Since we use few previous frame for generating motion-mask, we observed some

Table 4. Performance of Classifiers from level-2 to level-5

Level No.	Class	Total clips	$N_c/N_m/N_f$	Recall (%)	Precision (%)
2	Replay	399	336/63/34	84.21	90.81
	Real-time	390	344/46/29	88.21	92.23
3	Field view	352	334/18/17	94.88	95.16
	Non-field view	372	355/17/18	95.43	95.17
4a	Pitch view	296	260/36/28	87.84	90.28
	Long view	301	271/30/21	90.03	92.81
	Boundary view	164	147/17/34	89.63	81.22
4b	Close-up	261	236/25/15	90.42	94.02
	Crowd	196	181/15/25	92.35	87.86
5a	Batsman	193	161/32/24	83.42	87.03
	Fielder	171	147/24/25	85.96	85.46
	Umpire	39	33/6/7	84.62	82.50
5b	Players Gathering	53	43/10/12	81.13	78.18
	Spectator	158	146/12/10	92.41	93.59

miss-classification near the shot boundaries.

Since more number of edges are observed for crowd class, we use edge pixel density as a feature. We observed more than 90.42 % recall and 94.02 % precision for close-up detection. For close-up classification, we observed the average 84.66 % and 84.99 % recall and precision respectively. For crowd classification, we used similar technique which we used for close-up classification. We observed the average 86.77 % and 85.89 % recall and precision respectively.

4 Conclusion

In this paper, video semantic analysis is formulated based on low-level image features and high-level knowledge for cricket video sequences. The sports domain semantic knowledge encoded in the hierarchical classification not only reduces the cost of processing data drastically, but also significantly increases the classifier accuracy. The hierarchical framework enables the use of simple features and organizes the set of features in a semantically meaningful way. The proposed hierarchical semantic framework for event classification can be readily generalized to other sports domains as well as other types of video. Our future work includes higher-level semantic concept extraction based on the classified events for highlight generation, indexing and retrieval.

References

[1] M. Baillie and J. M. Jose. Audio-based event detection for sports video. *in Lecture Notes on Computer Science*, 2728, 2003.

[2] F. Bunyak, K. Palaniappan, S. Nath, and G. Seetharaman. Flux Tensor Constrained Geodesic Active Contours with Sensor Fusion for Persistent Object Tracking. *in Journal of Multimedia*, 2(4), 2007.

[3] V. Chasanis, A. Likas, and N. Galatsanos. Scene detection in videos using shot clustering and symbolic sequence segmentation. *in IEEE Workshop on Multimedia Signal Processing*, 2007.

[4] L. Duan, M. Xu, Q. Tian, C. Xu, and J. Jin. A unified framework for semantic shot classification in sports video. *in IEEE Transactions on Multimedia*, 7(6), 2005.

[5] A. Hanjalic. Generic approach to highlight extraction from a sport video. *in IEEE Int. Conf. on Image Processing*, 1, 2003.

[6] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros, and I. Sezan. Browsing sports video: trends in sports-related indexing and retrieval work. *in IEEE Signal Processing Magazine*, 23(2), 2006.

[7] M. H. Kolekar and S. Sengupta. Event-importance based customized and automatic cricket highlight generation. *in IEEE Int. Conf. on Multimedia and Expo*, 2006.

[8] Y. Li, J. Smith, T. Zhang, and S. Chang. Multimedia database management systems. *in Elsevier Journal of Visual Communication and Image Representation*, 2004.

[9] C. Ngo, T. Pong, and H. Zhang. On clustering and retrieval of video shots through temporal slices analysis. *in IEEE Transactions on Multimedia*, 4(4), 2002.

[10] K. P. Sankar, S. Pandey, and C. V. Jawahar. Text driven temporal segmentation of cricket video. *in Lecture Notes in Computer Science*, 4338, 2006.

[11] J. Wang, E. Chng, C. Xu, H. Lu, and Q. Tian. Generation of personalized music sports video using multimodal cues. *in IEEE Transaction on Multimedia*, 9(3), 2007.

[12] C. Xu, J. Wang, H. Lu, and Y. Zhang. A novel framework for semantic annotation and personalized retrieval of sports video. *in IEEE Transactions on Multimedia*, 10(3), 2008.

[13] P. Xu, L. Xie, S. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and system for segmentation and structure analysis in soccer video. *in IEEE Int. Conf. on Multimedia and Expo*, 2001.

[14] G. Zhu, Q. Huang, C. Xu, L. Xing, W. Gao, and H. Yao. Human behavior analysis for highlight ranking in broadcast racket sports video. *in IEEE Transactions on Multimedia*, 9(6), 2007.