

Semantic Analysis of a Cricket Broadcast Video

Rashish Tandon (Y6377)
CS 397 Project Report
Supervisor : Prof.Amitabha Mukherjee
{rashish , amit}@iitk.ac.in

April 20, 2009

1 Abstract

Most approaches to the semantic analysis of sports videos involve the use of auxiliary cues to detect events[4]. Detecting events of semantic importance based on video alone is a difficult task.

In this project, we attempt to semantically characterize a cricket broadcast video based on video alone. Initially, we perform shot boundary detection and shot classification based on multi-scale spatio temporal analysis of colour and optical flow features. This allows the representation of every frame by a feature vector over which a classifier can then be built. We can then query the video for shallow queries like “How many fours were hit by Hayden ?”, by aligning it with textual commentary[3].

Finally, we examine an optical flow based feature set construction to identify the semantic characteristics of the events (viz. balls) detected using the above technique. This may enable us to answer queries involving finer semantic features for eg. ”How many balls were hit by Hayden on the On Side ?” with greater accuracy and without the aid of any external cues like commentary.

2 Introduction

‘Semantics’ of a video involves the development of a model to identify, extract and represent ‘semantically’ relevant events. Events for sports video analysis are extracted from three primary channels : *video, audio and text*[5]. Each channel furnishes certain cues that correlate with the occurrence of events. As part of the project,we stick to the extraction of cues based on the video alone

For the purpose of semantic analysis, some amount of video processing needs to be done to generate a framework where the semantics of the video may be examined. The steps involved are listed in Figure 2 .Moreover, in episodic games like cricket where certain semantically relevant events (viz. balls) are repeated, it may be beneficial to be able to identify switches between cameras as well as to classify shots once their boundaries have been detected, as even the shots have a tendency to show a recurring pattern due to the inherent episodic nature of the game.

A common approach to scene change detection exploits changes in a Colour Histogram at shot boundaries to detect changes in scenes[1]. Such an approach assumes that the content of a video changes across shot boundaries and tries to characterize this change in terms of the Colour Histogram. This approach would fail in cases when a camera is switched but the the content of the scene remains the same.

We examine a multi-scale spatio temporal analysis of colour and flow features for Shot Boundary Detection and Shot Classification. Once every shot has been classified, a Bayesian Probability Analysis is done to detect the shots that represent the start of the ball. Finally, for every ball, we look at their semantic characterization through a generation of features based on optical flow. This feature vector is then used to classify the balls on the basis of the runs scored, area in which the ball is hit and the type of batting stroke played. The results of the experiments are presented.

Before we present the algorithms, we list a few common terminologies in video analysis :

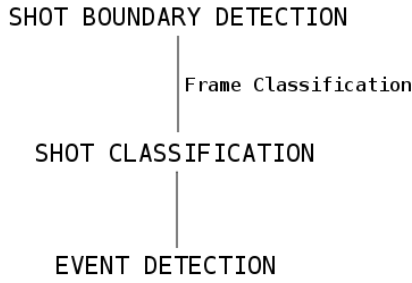


Figure 1: Video Processing done for Semantic Characterization

- **Frame** : The smallest entity in a video. A video is a sequence of frames
- **Shot** : A sequence of frames taken from the same camera
- **View** : A semantic entity. Frames with similar content may be deemed to have the same view. A shot may also be said to belong to a particular view

3 Shot Boundary Detection

There are 3 types of Shot Boundaries :

- Cut
- Fade
- Action Replay Marker

We only talk about Cut and Fade detection in this section. For details on detecting the ARM (Action Replay marker), please refer to the thesis by Dipen Rughwani[3].

A Cut involves a sudden/abrupt change in the view while a Fade involves a gradual change in the view. A colour histogram is used to characterize the view. Thus, it is expected that a Cut would involve a sudden change in the Colour Histogram, while the change in Fade would be gradual, yet more than the change when no transition is involved. Figure 3 shows that this is indeed so

However, it may happen that two images with very different content may have similar histograms, due to similar overall colour content. To overcome this difficulty, local histogram differences are also taken into consideration. This helps in retaining spatial information about the image. The image is divided into blocks and the histograms for each of these block are also represented in the feature vector used for classifying a frame as a Cut/Non-Cut and Fade/Non-Fade.

The following distance metric is used for computing the difference between 2 histograms

$$D(H_1, H_2) = \sum_{c=1}^n \frac{(H_1(c) - H_2(c))^2}{\max(H_1(c), H_2(c))} \text{ where,}$$

H_1 and H_2 are Histograms and n is the number of bins

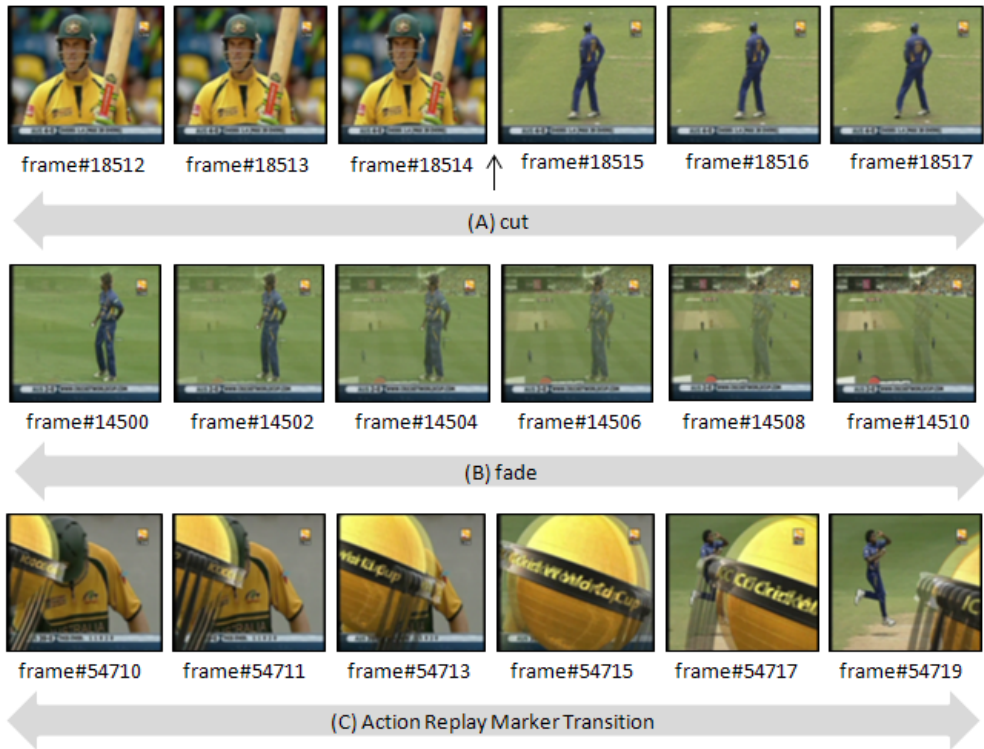


Figure 2: Examples Of Cut and Fade[3]

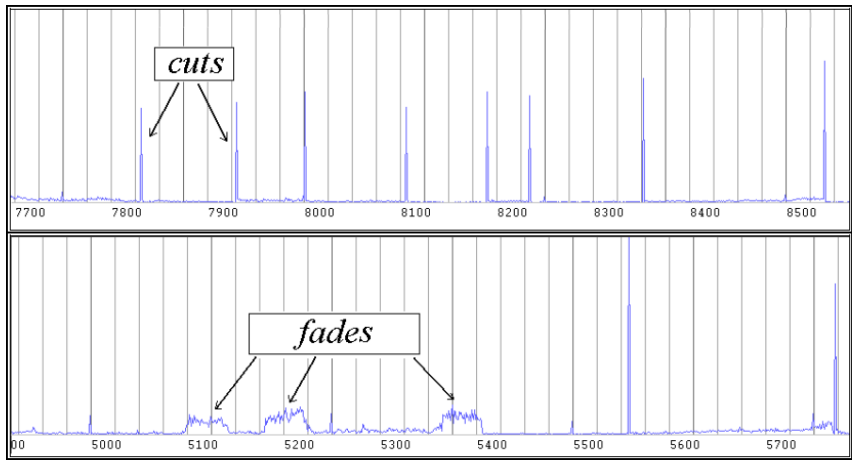


Figure 3: Changes in Colour Histogram for Cut and Fade[3]

Now, the feature vector for a frame f comprises of Global and Local Histogram differences computed between f and its previous 30 frames. So for a frame $f-k$ (where $k \leq 30$), there are 2 attributes in the feature vector. One corresponding to the Global Histogram difference and the other corresponding to the Local Histogram difference. Thus, the total feature vector length would be 60.

Another attribute added to the feature vector is the average magnitude of the optical flow. This is done to prevent misclassification of frames as fade due to very fast camera panning. Due to a fast camera panning while following the trajectory of the ball, the content of the frames may change rapidly in comparison to the usual rate, leading to higher histogram differences and hence, misclassification as a fade. Using optical flow helps address this issue, as is clear in the accuracy of Fade detection using this feature vector.

Using the feature vector of length 60,

Accuracy for Cut Detection = 90.59%
Accuracy for Fade Detection = 74.16%

Using the feature vector of length 61,

Accuracy for Cut Detection = 93.97%
Accuracy for Fade Detection = 82.92%

4 Shot Classification

Now that the video has been split into chunks of shots, the subsequent step is to classify these shots into categories. This is done by classifying each of the constituent frames of a shot into categories and then using a voting scheme to classify the shot. The frame is classified into one of the following categories (as shown in Figure 4)

- Batsman View
- Fielder View
- Ground View
- Pitch View
- Crowd View



Figure 4: Examples of the five classes for classification[3]

For the purpose of classification, a feature vector based on multi-scale spatio temporal colour and flow features is constructed. Every frame is examined at multiple levels. At the highest level (Level 0), the entire frame is examined for determining the colour and flow features. Thus, the colour and flow features at this level would be represented through the average (R,G,B) values and the average (dx,dy) respectively (Note that the flow features for every frame are computed using Proesman's Optical Flow).

Every Level branches into 4 lower levels, by splitting the part of the frame being examined at a particular level into 4 quadrants. Thus, at Level 1, we would have 4 blocks to be examined obtained by splitting the frame into 4 quadrants. The number of blocks at each Level k is 4^k . The colour and flow features for a particular block at Level k are computed by considering the colour and flow values for only the pixels in the block. Representation of a block in the feature vector requires 5 attributes (3 for colour and 2 for flow). Thus, the total attributes needed to incorporate Level k

into our feature vector is $5 * 4^k$.

The primary advantage of using multi-scale colour and flow features is that they retain the spatial information of a frame. We combine this multi-scale approach with a temporal one, by incorporating multi-scale features of preceding and succeeding frames for the frame being considered. To do this, we define μ_C^{fk} as the feature vector comprising of Colour features of frame f upto Level k, & μ_C^{fk} as the feature vector comprising of Colour and Flow features of frame f upto Level k.

Then, the following feature vectors have been experimented with for the purpose of classifying frames[3]

- Spatial Pyramid Of Colour(SC) - $\mu_C^{f_2}$
- Spatial Pyramid Of Colour and Flow(SCF) - $\mu_{CF}^{f_2}$
- Spatio-Temporal Pyramid Of Colour(StC) - $\mu_C^{f-2_0} \mu_C^{f-1_1} \mu_C^{f_2} \mu_C^{f+1_1} \mu_C^{f+2_0}$
- Spatio-Temporal Pyramid Of Colour and Flow(StCF) - $\mu_{CF}^{f-2_0} \mu_{CF}^{f-1_1} \mu_{CF}^{f_2} \mu_{CF}^{f+1_1} \mu_{CF}^{f+2_0}$

The results of Frame Classification are as follows[3]

Feature Vector	SC	SCF	StC	StCF
Accuracy	93.48%	93.46%	93.99%	93.81%

It is observed from the results that the flow features do not have much relevance for this type of a semantic differentiation. However, as we shall see in subsequent sections, they most definitely play a key role in finer grained semantic distinctions like knowing where the ball was hit or how it was hit.

Once the Frames have been classified, the Shot is classified into one of the categories mentioned or the *Other* category based on a voting scheme of Absolute Dominance / Relative Dominance. For a detailed approach on the voting scheme as well as an analysis of the results, please refer to the Thesis by Dipen Rughwani[3].

5 Ball Detection

The final step before semantic information may be extracted from the video is the identification of ‘semantically’ relevant events, in our case the starting of the ball. In cricket, a ball is the unit of play, and the knowledge of when it starts is of paramount importance for semantic analysis. A ball is presented in the video through a variety of camera shots. However, it is observed that certain camera shots occur more frequently in comparison to others. Being a game of recurring episodes of *balls*, there is a tendency for certain camera shots to recur. Also, some shots are likely to be followed/preceded by only certain other kinds of shots. As an example, a Pitch Shot is more likely to be preceded by a Batsman Shot/Ground Shot than others. It is hardly ever preceded by another Pitch Shot. Another important observation is that the Bowler’s runup just before the ball delivery is always shown as a Pitch Shot. Hence, the start of a ball is defined as

The Pitch Shot showing the bowler’s runup just before ball delivery

We call this shot a Strokeplay Shot.

Figure 5 shows the likelihood of shots of a particular type to be followed by shots of other types.

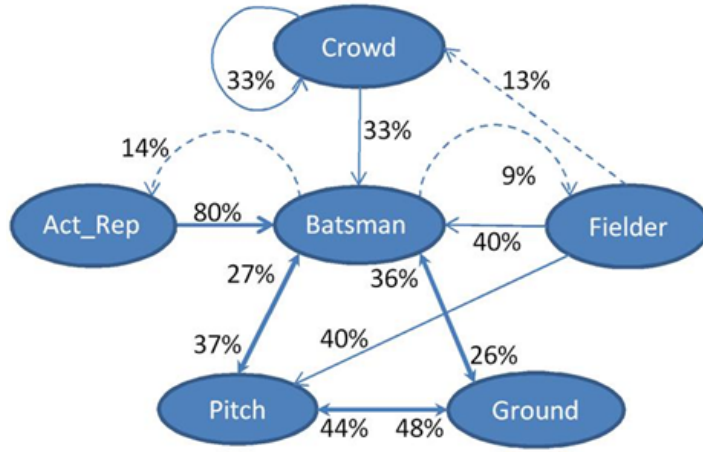


Figure 5: Transition Markov Probability Diagram[3]

Based on these observations, a Bayesian probability analysis is done to identify the strokeplay shot[3] based on the following parameters

- Class - Class to which the current shot belongs. Must be Pitch Shot to be a Strokeplay Shot
- Length - Number of Frames in the Current Shot. An upper threshold of 74 frames has been experimentally determined for this parameter
- Interval - Number of Frames from the Previous Ball Boundary. A lower threshold of 364 frames has been experimentally determined for this parameter
- Previous-Class - Class of the Shot preceding the current shot. The likelihood for the current shot being a Strokeplay shot is computed and a lower limit of 0.15 has been set experimentally for this parameter

A Bayesian Probability Analysis is an effective approach and yields an accuracy of close to 100% for Strokeplay Shot identification.

6 Semantic Information Extraction

Once the starting of our every ball has been identified, our aim is to generate a feature set that allows for the following categories of Semantic distinctions

- Runs Scored - Classify the ball into the classes 0,1,2,3,4,5, or 6
- Where the Ball was Hit - Classify the ball into the classes On-Side, Off-Side, or Centre
- Type of Stroke Played - Classify the ball into one 8 classes of Strokes viz. Leg Glance, Hook, On Drive, Straight Drive, Off Drive, Cover Drive, Cut or Late Cut. See Figure 6

A number of feature vectors based on multi-scale temporal flow features are constructed by considering the following attributes (which would intuitively suggest the nature of the stroke etc.)

- L - Length of the Strokeplay Shot + the Shot that succeeds it
- V_k - Multi-Scale Flow velocities upto Level k at the point of contact with the ball
- M - Maximum Displacement Vector from the initial Pitch View shot



Figure 6: Areas on the Field corresponding to different strokes

The shots under consideration for extracting semantic information are the Strokeplay Shot and the shot that follows it. The shot after the strokeplay shot usually follows the trajectory of the ball till it arrives at a fielder. Then, either the shot changes or the same shot continues showing the fielder returning the ball. Sometimes, a different camera angle is used to show the fielder returning the ball. On other occasions, the batsmen may be shown running between the wickets. However, it is very difficult to predict the exact nature of the shot that follows it, and for this reason, we restrict our analysis to only 2 shots.

The length of these 2 shots is a relevant semantic attribute as it is observed to be small for both very low (0 or 1) and very high (4 or 6) number of runs as compared to moderate number of runs (2 or 3). The flow velocities at the point of hitting the ball are relevant as intuitively, one may say that the initial power of strokeplay would determine the nature of the ball. Maximum Displacement vector would represent the extent to which the ball has travelled in the field and whether or not it has been intercepted by a fielder. Hence, this attribute is relevant too.

Before the feature vector may be generated, we need to identify the frame for the point of contact with the ball. This is described in the following section

6.1 Point Of Contact Detection

It is observed that the camera motion involves a sudden jerk around the point when the ball is hit, as the camera tries to catch up with the ball while following its trajectory. This leads to a sharp peak in the optical flow (see Figure 7 & Figure 8). It is this peak that is detected and deemed as the point of hitting the ball.

The peak in optical flow is computed by composition with discrete Hermite functions of varying standard deviation ($\sigma = 0.5, 1, 1.5, 2$). It is observed that the Hermite function with standard deviation of 0.5 performs the best for hitting point detection, which implies that the point of hitting has quite a sharp peak indeed.

$$\begin{aligned}
 g_x^f &= \text{Value of composition of Hermite Functions with the Optical Flow along X at a frame f} \\
 g_y^f &= \text{Value of composition of Hermite Functions with the Optical Flow along Y at a frame f} \\
 g^f &= \text{Net Value of Composition of Hermite Function with Optical Flow at a frame f}
 \end{aligned}$$

Then we have,

$$|g^f| = |g_x^f| + |g_y^f|$$

The frame f that is chosen is the one with maximum value of g^f .

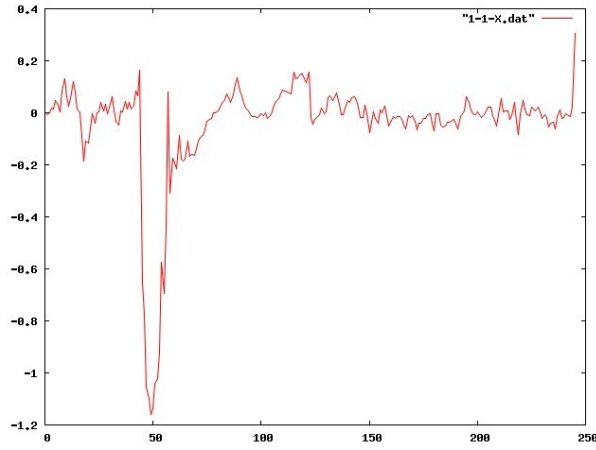


Figure 7: Optical Flow Velocities along X for frames in ball 0.1[3]

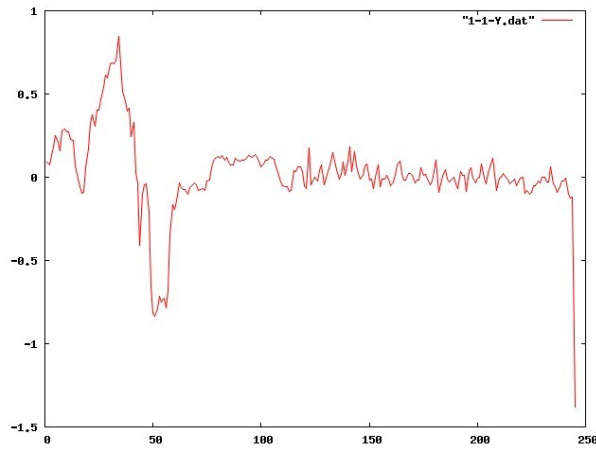


Figure 8: Optical Flow Velocities along Y for frames in ball 0.1[3]

The attribute L requires a length of 1 in the feature vector.

The attribute V_k for each block at a level, comprises of the average flow along X and Y, and a new feature, the net zoom in the block, computed using flow velocities at the lower levels of a block. Thus, each block requires a length of 3 in the feature vector and each level requires $4^k * 3$ length in the feature vector. Thus V_1 would need a total length of $3 + 4*3 = 15$ in the feature vector.

The attribute M requires a length of 2 in the feature vector for the components of displacement along X and Y.

The following set of feature vectors are experimented with for the purpose of achieving a semantic distinction

- L V_0 M - Total Vector Length = 6
- V_0 M - Total Vector Length = 5
- L V_1 M - Total Vector Length = 18
- V_1 M - Total Vector Length = 17

The results are described in the subsequent section.

7 Evaluation and Results

For testing our feature vectors, the first 10 overs of the Cricket World Cup between Australia and Sri Lanka were used. For each ball, the flow was computed and the required parameters were determined. Then, each ball was manually identified into one of the classes based on each of the 3 aforementioned categories. Finally, a Multiclass SVM with 4-fold cross validation was used to test the effectiveness of our feature vectors for the various semantic distinctions. The parameters used for the classifier were

- kernel - A Linear Kernel was used for all classifications
- loss function - No Loss function was used

The results of accuracy of the experiments are

Feature Vector	L V ₀ M	V ₀ M	L V ₁ M	V ₁ M
RUNS	77.05%	75.41%	72.13%	72.13%
WHERE THE BALL WAS HIT	88.52%	93.44%	83.61%	83.61%
TYPE OF STROKE	60.66%	62.30%	65.57%	63.93%

8 Conclusion

The highest accuracy achieved for RUNS is with the feature vector L V₀ M, while that for WHERE THE BALL WAS HIT is V₀ M. This is an experimental justification for the intuitive idea that the attribute L is redundant for computation of where the ball is hit. In fact, multi-scale velocities also do not play any role in the knowledge of where a ball was hit. The highest accuracy for the TYPE OF STROKE is obtained using the feature vector L V₁ M, suggesting the role of multi-scale velocities in determining the stroke.

The accuracy over Runs and Where the Ball was hit is a considerable improvement over existing work[2]. However, the accuracy over the Type Of Stroke doesn't fare as well, and a better set of feature vectors needs to be identified to improve its accuracy. Perhaps, the consideration of flow velocities from the point of hitting the ball to the point of farthest displacement would also be necessary in some way to know the stroke.

References

- [1] Ahmed K. Elmagarmid Haitao Jiang, Abdelsalam (Sumi) Helal and Anupam Joshi. Scene change detection techniques for video database systems. 1998.
- [2] Venkatesh S. Lazarescu M. and West G. On the automatic indexing of cricket using camera motion parameters. 2002.
- [3] Dipen Rughwani. M.tech thesis, shot classification and semantic query processing on broadcast cricket videos. 2008.
- [4] J. R. Wang and N. Parameswaran. Survey of sports video analysis: research issues and applications.
- [5] Xinguo Yu and Dirk Farin. Current and emerging topics in sports video processing. 2005.

I would like to thank my supervisor, Prof. Amitabha Mukherjee for giving me the opportunity to do this project and also for his invaluable guidance and support throughout the project.