Cricket Activity Detection

Ashok Kumar
Javesh Garg
Advisor: Dr. Amitabha Mukerjee
Dept. of Computer Science and Engineering
{*ashokkrm,javeshg,amit*} *@iitk.ac.in*

Abstract

The use of Optical flow analysis along with the shot boundary detection can greatly help in the analysis of broadcasted sports' videos. In this project we classify different types of cricket strokes played by a batsman during the match. The agent first splits the cricket match video into shots using supervised learning, and finds out when the batsman plays the stroke. Agent then classifies the stroke type using optical flow technique.

keywords: shot boundary detection, frame classification, optical flow analysis, gradual fade, panning, zooming, angle of view, YUV color space

1 Introduction

In the field of computer vision, analysis of sports' videos is one of recently researched topic.[1], [2] In cricket, an important task in activity detection is to classify batsman stroke-play during a match. In our paper, we classify the various cricket strokes played by a batsman into four different directions.

This requires solving problem of shot boundary detection in a video. We split the video into shots by supervised learning approach using colour histograms. The shot boundaries can be any one of 'cut' or a 'dissolve/fade'. We then classify the video frames into four classes namely, ground, fielder, pitch and other using multi-class SVM. Using this classification, we can find out the video segments in which cricket stroke is played by a batsman. Now we have a basic entity of a few frames where the batsman hits the ball. We do optical flow analysis[3] on this part of the video to determine the direction of the stroke played by the batsman. This takes into account the camera pan and zoom also.

The complexity of the problem is due to the fact that videos are at $25fps$ and also the broadcasted videos use several cameras with multiple transition effects and action-replays. The completely dynamic environment also adds to the problem and finding any meaningful pattern in the video sequence can be quite cumbersome. Intermittent crowd view scoreboards etc. also adds to complexity the problem. The main motivation for choosing this work is that this can be a major step in further developing an automatic commentary system which would be a huge contribution to the field of sports. Some work like ball-start detection and cricket highlights retrieval have already been done in this field. Our approach although presently tested on complete videos, can be easily used with streaming videos also.

2 Implementation

2.1 Shot boundary detection

Shot boundary detection is one of the fundamental problem in field of computer vision. It is the very first step used for segmentation and classification of a video data.

A shot is an uninterrupted sequence of frames captured by a single camera. Two consecutive shots can be separated by various transitions like cut (Figure 1) or fades (Figure 2) (also called shot boundaries).

For detecting cuts and fades we have followed a histogram based approach [5].

In this approach we represent each frame as a feature vector.

2.1.1 Feature vector generation and classification

To generate feature vector corresponding to a frame we represent each frame as a YUV image histogram

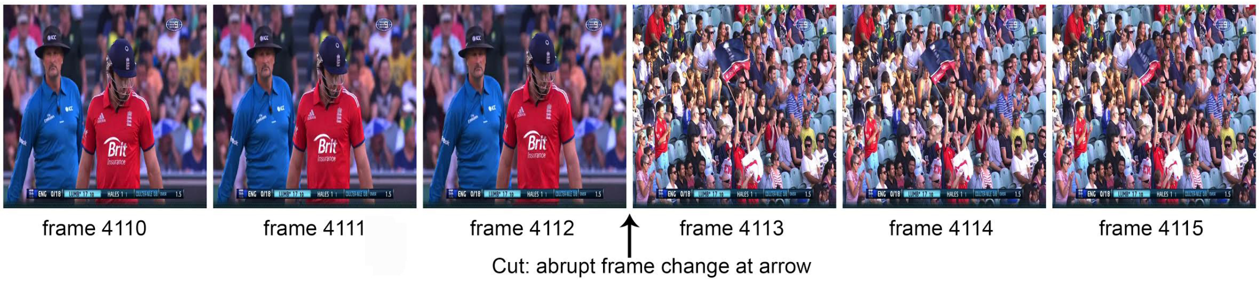

Figure 1: *cut between two frames*

Figure 2: *figure showing Gradual/Fade change in frames*

which stores the total number of pixels in each bin (distribution of colors in an image).
We compute both global and local histograms corresponding to a frame.

- Global histogram with n bins. Global histogram is less sensitive to camera motion but it fails on fades as the global difference is less and alone it does not give good results.

- To compute local histogram we first divide an image into $m \times m$ blocks. Local histogram includes color's distribution in different regions. It is a vector of many block histograms, where each histogram corresponds to a block of image.

- The local histogram takes the spatial information into account and combined with the global histogram can give good results.

- Still, the histograms are just based on colour distribution and thus two very different images may still appear to be similar using this approach.

We quantized the UV plane of the color space into 40x40 bins and using these histograms, compute squared difference with histograms of previous 30 frames.

The difference function is represented by given function:

$$Global_{diff}(f,g) = \frac{Hist_f(c) - Hist_g(c)}{max(Hist_f(c), Hist_g(c)}$$

$$Local_{diff}(f,g) = \sum_{i=1}^{m} \sum_{i=1}^{m} \frac{Hist_{f_{i,j}}(c) - Hist_{g_{i,j}}(c)}{max(Hist_{f_{i,j}}(c), Hist_{g_{i,j}}(c)}$$

Data: Frame number: f
Result: Feature vector corresponding to that frame

$V \leftarrow$ Initialize a vector of size 60
for $i \leftarrow$ *0 to 60* **do**
    $V(i) \leftarrow 0$
end
$index \leftarrow f - 1$
for $i \leftarrow f - 1$ *to* $max(f - 30, 0)$ **do**
    $V(index - i) \leftarrow global_{diff}(f, i)$
    $V(index + 30 - i) \leftarrow local_{Diff}(f, i)$
end
return V

Algorithm 1: Feature representation

- Training and testing were done using K-fold cross validation method with K=3.

- For cut and fade classification in testing data we used $K - NN$ algorithm with $K = 5$.

2.2 Frame classification

A frame in a cricket match video can be classified into 7 major classes - {Ground view, Batsman view, Crowd view, Long shot, Pitch view, Fielder view and others}. This is an important step in the analysis of the video to demarcate any major activity during the match like start of a ball, stroke play by batsman or a fielding attempt. Most common activities on the field can be tracked using these classes. Say we want to find out when does the bowler starts to bowl. Mostly at the start of a new ball, there is a cut in the video and

Figure 3: *Different classes of frames in video*

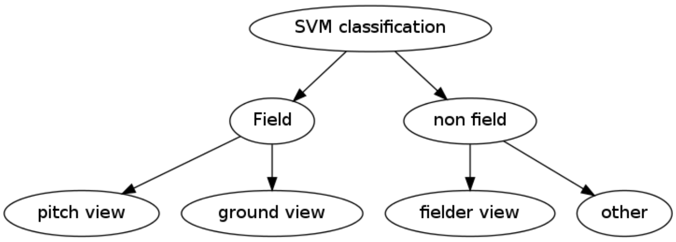

Figure 4: *Multi-class SVM*

after the cut there is a sequence of frames showing the pitch view. Such heuristics can help find patterns in the broadcast videos.

For the classification we first tried the grass pixel ratio - colour based approach[6] to classify the frames into field view (Ground view, Long shot and Pitch view) and non-field view (Crowd view, Batsman view, Fielder view and others). But supervised learning methods gave better results than this approach as it depends on the colour of grass and time of the game-play.

Then we used naive Bayesian and $K-NN$ approach to classify the frames into classes but still the results were not very accurate. The major misclassifications were that these algorithm classified the Pitch views into Ground views and also mixed Batsman views and Crowd views. Batsman and Crowd views got mixed as most of the times the background of batsman view was full of crowd. Also sometimes spectators usually wear same clothes as the teams they cheer and thus the algorithms misclassifies them.

In our work we did not require explicit classification into Batsman view , crowd view etc and thus classified into 4 major classes only - {Ground view, Pitch view, Fielder view and others}. Finally we used multi-class SVM to classify the frames into these classes (Figure 4)

We split the single camera shots obtained in Step 1 into parts where the batsman plays a stroke. We see that the event of start of a ball is marked by a series of frames classified into a pitch view (when the bowler takes a run up). So we know that the batsman will play the ball in a next few frames. Now if the batsman plays the shot, say a cover drive, then there is a sequence of ground view after a cut in the video. If the batsman misses the ball then we see no cut and a sequence of fielder view comes after that (the bowler or the wicket-keeper). Third case is when the batsman plays a defensive shot. In this case we find that there is longer sequence of pitch view and then there is a cut or fade in the video.

2.3 Stroke classification using optical flow

For stroke classification we first followed tree-structure approach [4] for batsman pose estimation, but this approach didn't work. Then we used optical flow approach to classify direction of the stroke played into one of the 4 directions.

After obtaining video segments in which batsman plays stroke from step 2, we first split that video into set of images and then compute net optical flow using Algorithm 2 [3]. It can be observed that direction of motion of significant pixels (which have velocity vectors $\geq$ average velocity vectors) is opposite to the direction of stroke played. This is because of the camera pan and zoom along that direction and thus the pixels show opposite motion. For example in Figure 6, the optical field lines are shown for a cover drive i.e. stroke played in direction 2. we need to take into account only the significant pixels as others tend to give false results due to players moving in random directions (batsman usually moves forwards in the pitch and the bowler is taking run ups etc.).

Using the video segments we can also classify whether the batsman missed the ball or played a defensive shot too. When there is a miss by the batsman then, we can classify the event using just frame classification only (stated in section 2.2 and for defensive play we find that the velocity vectors are very small in magnitude combined with the frame classification approach.

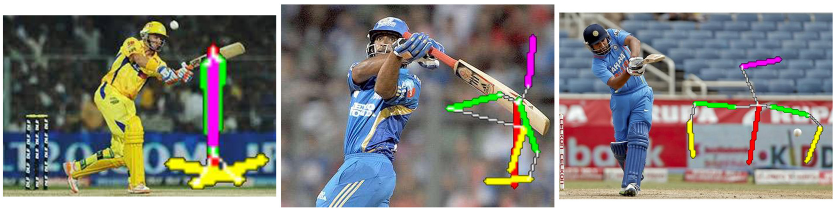

Figure 5: *Tree-structure approach*

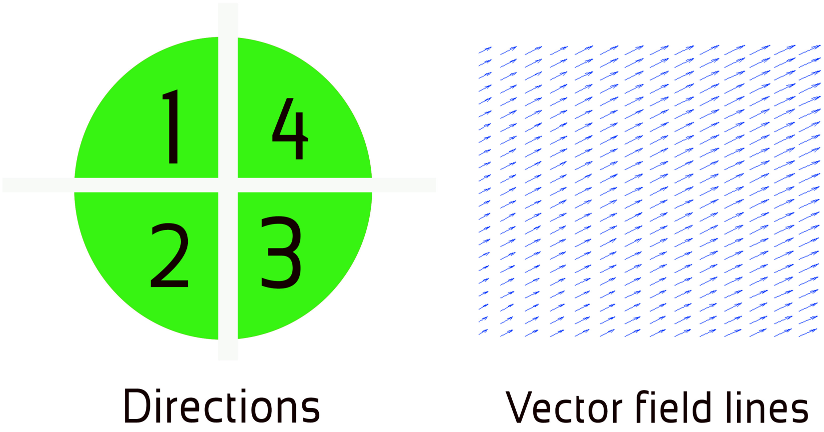

Figure 6: *Classification of stroke direction into 4 directions*

Data: continuous sequence of frames: $F (|F| \geq 2)$
Result: Direction of stroke
$[MV_x, MV_y] \leftarrow OpticalFlow(F)$ (using [3])
$[x, y] = size(MV_x)$ // All images are of $n \times n$
for $i \leftarrow 1$ *to* 4 **do**
    $count[i] \leftarrow 0$ // maintains no of pixel in direction i
end
$Avg_{velocity} = Compute\_Avg\_velocity(MV_x, MV_y)$
for $i \leftarrow$ *0 to x* **do**
    for $j \leftarrow 0$ *to* y **do**
        if $velocity(pixel(i, j)) > Avg_{velocity}$ **then**
            $count[Angle(pixel(i, j))] = count[Angle(pixel(i, j))] + 1]$
            // $Angle(pixel(i, j))$ and $velocity(pixel(i, j))$ are computed using MV_x, MV_y
        end
    end
end
return $Direction\_max\_val(count)$

Algorithm 2: Computing direction of stroke

3 Results and Datasets

3.1 Dataset

There is no standard dataset for the cricket match analysis available and hence we had to create our own dataset. For this we chose an 8 over cricket match video played between Australia and England (25 fps). Training and testing were done using K-fold cross validation method with K=3. Approximately 29000 frames were manually analyzed for training purposes and about 14000 frames were used for testing.

3.2 Result

The results for the shot boundary detection are stated in Table 1 and we have achieved comparable results for cuts detection and very good results for fades. We achieved good recall values for finding the video segments when the bowler bowls and the batsman plays the stroke or defensive shot.

- Recall: 90.90%
- Precision: 50%

The overall accuracy of stroke direction classification using optical flow analysis of the video segments is 80%. We also achieved good results for detecting defensive shots or balls missed by the batsman.

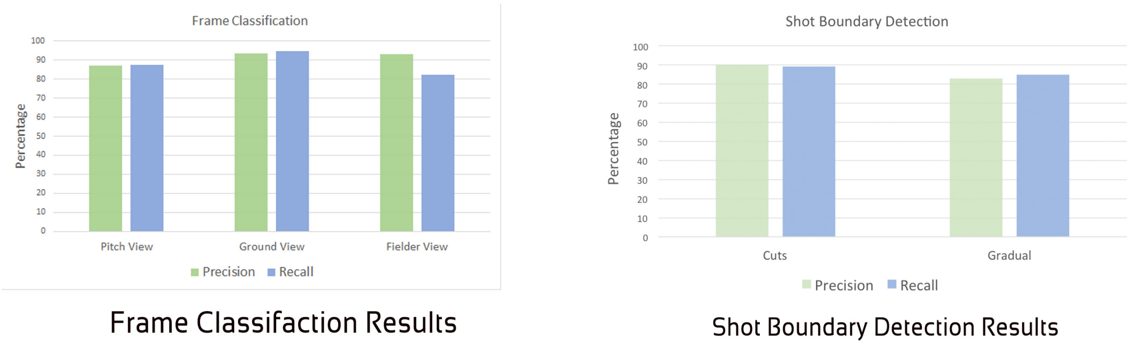

Figure 7: *Results*

Transition	Total Present	Total Obtained	Correct Obtained	Precision	Recall
Cuts	82	81	73	90.12 %	89.02 %
Gradual	46	47	39	82.97%	84.78%

Table 1: *Shot Boundary Detection*

Frame Class	Total Present	Total Obtained	Correct Obtained	Precision	Recall
Pitch View	571	573	499	87.00%	87.39%
Ground View	773	781	732	93.70%	94.60%
Fielder View	453	400	373	93.25%	82.30%

Table 2: *Frame Classification*

Figure 8: *Example of common errors obtained in shot boundary. Misses tend to be dominated in frames having similar background, while false positives frequently occur when there is a gradual change between frames.*

3.3 Analysis of Error

In shot boundary detection we observed 7 cuts correctly, Most of the false positives are observed in frames having fade/gradual change (Figure 8). And misses are observed in frames having similar background. For example - when the camera shows crowd view in two shots separated by a cut then, the background is almost same and the classifier misses the cut.

4 Acknowledgements

We would like to thanks Prof. Amitabha Mukerjee for his valuable suggestions and guidance throughout our project.
We also thank Prof. Vinay P. Namboodiri and Mr. Dipen S. Rughwani for guiding us.

References

[1] Rughwani, Dipen S. "Shot Classification and Semantic Query Processing on Broadcast Cricket Videos." M. Tech Thesis (2008).

[2] Singh, Khushdeep, and Aakash Verma. "Tennis Stroke Detection." (2012).

[3] B.K.P. Horn and B.G. Schunck, "Determining optical flow." Artificial Intelligence, vol 17, pp 185–203, 1981.

[4] Ramanan, Deva. "Learning to parse images of articulated bodies." NIPS. Vol. 1. No. 6. 2006.

[5] Irena Koprinska and Sergio Carrato. Temp oral video segmentation: A survey. In Signal Processing: Image Communication, volume 16(5), pages 477-500, 2001

[6] Kolekar, Maheshkumar H., Kannappan Palaniappan, and Somnath Sengupta. "Semantic event detection and classification in cricket video sequence." Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on. IEEE, 2008.