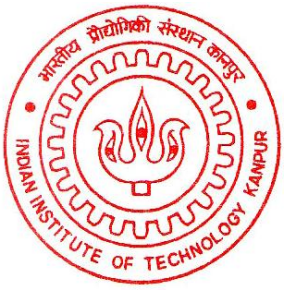
# AUTOMATIC HIGHLIGHTS EXTRACTION IN CRICKET




CS365A - ARTIFICIAL INTELLIGENCE
IIT KANPUR
2013-14

Anjani Kumar (11101)
Sumedh Masulkar (11736)
*{anjanik, sumedh}@iitk.ac.in*

Guided by:
Dr. Amitabha Mukerjee
*amit@cse.iitk.ac.in*

April 23, 2014

## ABSTRACT

This project deals with a very important branch of artificial intelligence, i.e. Vision. This project presents ways to detect important segments of a cricket video, and thus extract highlights automatically from a full length video. Similar techniques can be implied to create highlights of other sports too, thus reducing huge amount of efforts people put to create game highlights in sports. In this project, we specifically aim to produce better and reliable results using supervised learning. The extraction process is carried out at multiple levels, removing some unnecessary part of the video at each level. We achieve promising results, which are significant improvements over those of unsupervised methods prevalent for this problem.

## ACKNOWLEDGEMENTS

We would like to express our sincere gratitude towards Dr. Amitabha Mukerjee, Department of Computer Science and Engineering, IIT Kanpur, and our instructor for his invaluable support and guidance provided throughout the project. We consider ourselves very fortunate to have worked under his supervision. We are also grateful to Mr. MS Ram and Miss. Sunakshi Gupta for their suggestions. We are also thankful to Mr. Dipen Raghuwani for the dataset he provided.

And, lastly, we thank all our friends and colleagues for their support and assistance in the course of the project.

## *Coming Up….*

# Introduction

Highlights extraction of sports is a popular topic. Sports videos of full length contain uninteresting events too, and in todays time compressed world, people only want to view the interesting sequences of a sport match. Cricket is the second most watched game in the world. Our interest in cricket was another motivating factor to take this as a project.

- What people do not expect in cricket highlights?
    - Uninteresting part of the game, replays, and spectators cheering. So, we aim to automatically detect frames belonging to these sequences and remove them, thus producing highlights of the match. Actually, similar approaches can be used to extract highlights in other sports too.
- What else is expected from the project?
    - The detection of the events must be accurate enough, so that neither it misses out important events, nor does it fail to remove the uninteresting parts.

# Background

Automatic generation of cricket highlights using Hidden Markov Model(HMM) was proposed in [1][2][3]. [3] fused in audio information in addition to motion information. Whereas in [4], the author proposed an unsupervised event discovery and detection framework with use of color histogram(CH) or histogram of oriented gradients(HOG), which can potentially be generalized to different sports. The unigram and bigram statistics of detected events are then used to provide a compact representation of the video. [5] presented another novel approach towards highlights generation of sports videos by extracting events and semantic concepts. The method extracted event sequence from video and classifies each sequence into a concept by sequential association mining. The extracted concepts and events are then selected according to their degree of importance. This was further improved in [6].

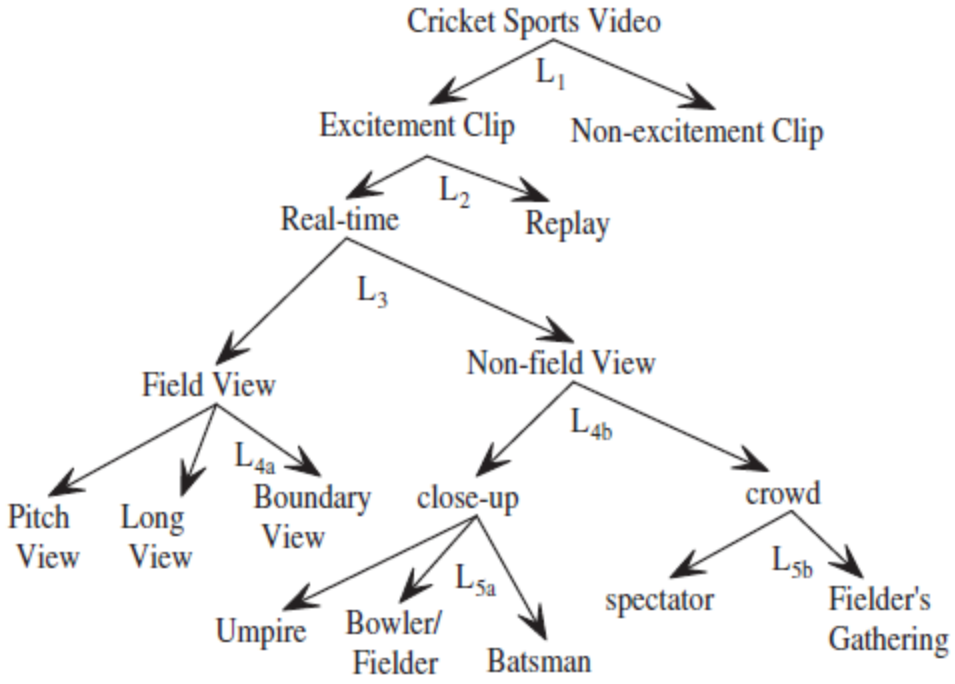
[6] presented a hierarchical framework and effective algorithms for cricket event detection and classification, which avoids shot detection and clustering. Extraction was divided into multiple levels(described in our approach below).

[7] again used shot detection techniques, and text processing on the commentary to identify action in each ball.

## Our Approach

We are primarily going to follow the work done in [6].

### Hierarchy of Extraction




(Image from [6])

**Figure 1. Tree Diagram of Hierarchical Frame-work**

- As can be seen in the diagram, there are 5 levels in the hierarchy.
- Level 1 - Excitement detection: A particular video frame is considered as an excitation frame if product of its audio excitement and zero crossing rate(ZCR) exceeds a certain threshold.
- Level 2 - Replay Detection: Replay segment is sandwiched by two logo transitions. Hence, replays can be detected using Hue-Histogram Difference(HDD) and removed.
- Level 3 - Field View Detection: Dominant Grass Pixel Ratio(DGPR) is calculated for a view, which varies between 0.16 to 0.24 for the field view. Thus, a non-field view can be removed.
- Level 4 - Field View and Close Up Detection: Percentage of field pixels in regions are calculated and some thresholds are fixed, and frame can then be classified as long-view,

boundary view or pitch view. Similarly, edge pixels are used to detect close views or crowd views.

- Level 5 - Fielders gathering or crowd Detection: Crowd frames are removed from the video. The detection is done by computing histogram distance of hue-histogram of frames.
- Thus, highlights for the given video are extracted.

## Level-1: Excitement Detection

Level 1 of the hierarchy involves -

- Spectator's cheer and commentator's speech analysis.
- Two popular content analysis techniques - Short-time audio energy(E) and Short-time Zero Crossing Rate(Z)[6].
- If E * Z is greater than a given threshold(some function of mean), the particular frame is an excitation frame, otherwise not.
- All the frames, marked unexcited may be removed from the sequence.

**Difficulty:** Matlab ran out of memory when we tried to read a file larger than 2-3 seconds. So we could not test this on the cricket video we were using. Even if we tried testing on smaller parts of video, that would have given us wrong mean, and incorrect results. But we tested it on a different file(so.wav, available with code), on which it seemed to work just fine.

## Level-2: Replay Detection

- Characteristics of action replay we tried to exploit for detection:
  - ❑ A replay is sandwiched between two logo(templates) transitions and the score bar is removed.

  - ❑ We can detect the logo transitions(and thus replay) by calculating hue-histogram difference of frames with the reference logo template.

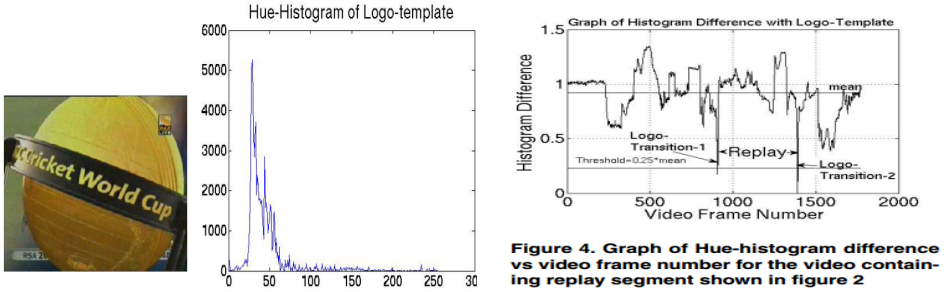



**Figure 4. Graph of Hue-histogram difference vs video frame number for the video containing replay segment shown in figure 2**

## Problems with the approach:

The threshold which should be used to classify a frame as logo transition, may differ from logo to logo, and thus, from match to match. If set at 1/4th of mean, it does not show much promising result. There are no false positives, but there are many true negatives.

## Our proposed solution:

**Method 1.** We propose a solution which is less dependent on the specific match for replay detection. We calculate ***r*** - correlation coefficient(corr2 in matlab) of the frames with the reference logo.

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

where $\bar{A}$ = mean2(A), and $\bar{B}$ = mean2(B).

corr2 calculates the degree similarity of images. For exactly similar images, its value is 1 and 0 for two very different images. Thus, applying corr2 on a replay template and frames, if the value of correlation coefficient is greater than some threshold, we can classify it as part of logo transition. Experimentally, we found best results when the threshold was taken to be 0.65. This method is also **much faster** than the approach in [6].

**Method 2(not in the poster).** Another method which shows better results would be to simply detect the scorebars, also suggested in [8]. If a frame does not contain scorebar, than it is a replay frame. Detecting scorebar is easy and efficient. We take a reference scorebar, then to detect if a frame contains scorebar, we take the part where the scorebar is supposed to be. For, our dataset, it was at bottom $6/7^{th}$ of the image. Then, we calculate hue-histogram difference of the part with reference scorebar. A threshold of 8000 gave promising results with accuracy>97%.

### Level-3: Field View Detection

- Dominant Grass Pixel Ratio(DGPR) is used to classify frames.
- DGPR = $(x_g/x)$ where $x_g$ is number of pixels of grass, and x is total number of pixels.
- For field view, DGPR values is greater than 0.07 whereas DGPR is smaller for non-field views.

## Problems with the approach:

The threshold may vary from match to match since color of grass in the match may vary. It is really difficult to calculate the threshold experimentally for every match.

**Our proposed solution:**

We propose to use supervised learning. We train svm on some training images of field view, and then use the trained svm to classify images as field-view or non-field view.

### Level-4a: Field View Classification

- Classified as pitch view, long view or boundary view.
- Introduces the concept of flux tensor - temporal variations of the optical flow field within the local 3D spatiotemporal volume.
- Percentage of field pixels used to differentiate between views.

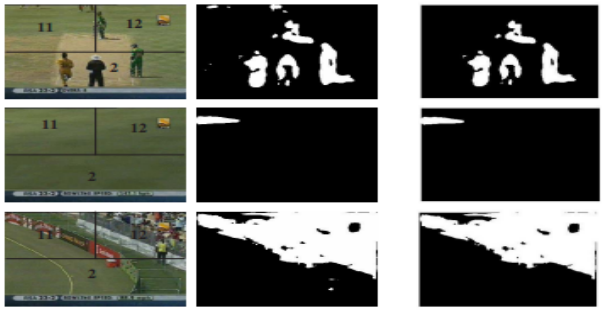


**Figure 7. Row-1 shows pitch view: (a) Image (b) motion-mask (c) connected component image, Row-2 shows long view: (d) Image (e) motion-mask (f) connected component image, Row-3 shows boundary view: (g) Image (h) motion-mask (i) connected component image**

**Algorithm[6]**

**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*

**Problems with the approach:**

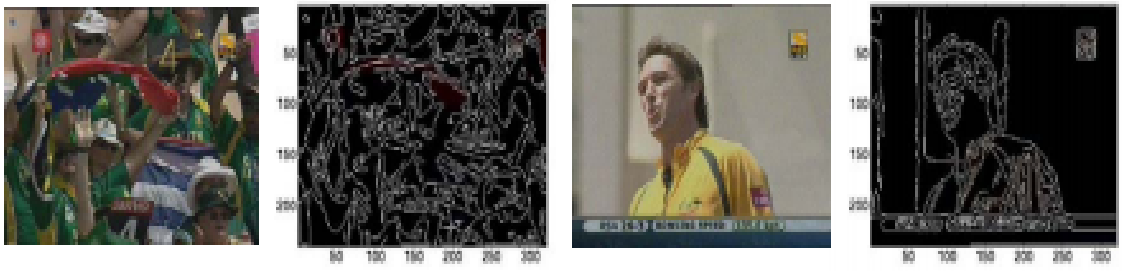
The threshold may vary from match to match since color of grass in the match may vary. It is really difficult to calculate the threshold experimentally for every match. And if the video is not classified as long-view or boundary view, due to little variation from threshold, it is incorrectly classified as pitch view, which gives a lots of false positives for pitch view, as visible in the results.

**Our proposed solution:**

We propose to use supervised learning. We train svm on some training images of field view, and then use the trained svm to classify images as long-view, boundary or pitch view.

### Level-4b: Non-Field View Classification

- Close-Up or Crowd view
  - ❑ RGB image is converted to $YC_bC_r$.
  - ❑ Percentage of edge pixels(EP) are calculated using Canny operator.
  - ❑ A threshold for EP classifies frames as close up view or crowd view.



**Note:** We also tried svm for this level, and it gave far better results for crowd detection, but not in case of close ups. The images of the **following** kind is always classified as crowd view, which is tagged as a closeup frame manually. It is important to realize that solution of this problem is not trivial. This is the only reason, supervised and unsupervised methods both gave <85-90% accuracies for level 4b.



## Level-5: Crowd Classification

- 5b - Crowd classification into spectators or fielders gathering.
- Fielders usually gather after an interesting event and have field as background, and have similar colored clothes. This frames should be kept in highlights.

Fielders gathering

Spectators view


# Results

### *Important:*

$$\text{Precision} = \frac{tp}{tp + fp} \quad \text{....(1)}$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad \text{....(2)}$$

**From wikipedia.*

Here *tp* = True positives, *fp* = False positives, *fn* = False Negatives

### *Experimental Results:*

**Level 2: Replay Detection**

- Using threshold 0.65 for corr2, we observed following values of precision and recall using our method and using approach suggested in [6].
- The precision and recall for [6] are based on code we developed and was tested on our dataset.

| | **Replay detection(Our approach)** | **Replay detection(approach in [6])** |
|---|---|---|
| **Precision** | 100% | 100% |
| **Recall** | 96.77% | 53.33% |

**Level 2: Scorebar Detection[7] (added after poster presentation)**

- We observed following recall and precision for detecting frames with no scorebars. The testing was done on over 20000 frames.
    - Precision - 96.99%
    - Recall - 100%

**Classification into Field-views and Non-field views:**

After training svm on 4254 images, and testing on 4176 images, we observed following precision and recall -

- Precision - 96.48%
- Recall - 88.03%

as compared to the following figures were produced by using approach proposed by [6], after performing the tests on same images,

- Precision - 72.70%

- Recall - 78.63%

All the following tests have been performed on over 4000 images, after training done by approximately same number of images, if not mentioned otherwise.

**Classification of Field view into pitch, long, boundary views:**

| | Pitch view(Our method) | Pitch view (approach in [6]) | Long view(Our method) | Long view (approach in [6]) | Boundary view(Our method) | Boundary view (approach in [6]) |
|---|---|---|---|---|---|---|
| **Precision** | 98.21% | 20.66% | 96.11% | 63.82% | 97.56% | 8.47% |
| **Recall** | 95.21% | 69.18% | 96.60% | 28.03% | 93.69% | 27.5% |

**Classification of Non Field view into Crowd view, Close-up views:**

| | Crowd view(Our method) | Crowd view (approach in [6]) | Closeup view(Our method) | Closeup view (approach in [6]) |
|---|---|---|---|---|
| **Precision** | 94.29% | 44.66% | 82.42% | 98.58% |
| **Recall** | 98.54% | 93.70% | 52.71% | 79.11% |

**Classification of crowd view into fielder's gathering, spectator's crowd:**

After training svm on 2444 images, and testing on 1184 images, we observed following precision and recall -

- Precision - 100%
- Recall - 99.42%

**Please see disclaimer.*

## Future Works

In future, we can work upon to improve the accuracy even more for the detection of events at different levels so as to extract the highlights without much human intervention. Anything less than perfect, may miss out some interesting events, and may instead add other sequences which may not be interesting to the viewer. This would be the reason manual extraction would be preferred over automatic extraction of sports highlights. Thus, future work includes improving the results by finding better alternatives. Another future work includes finding solutions to some problems such as close-up detection in level 4b.

## Conclusion

We applied supervised methods to perform event detection in cricket. The results observed were very significantly better than the unsupervised approach. We tested our approaches for each level on a dataset of more than 4000 frames. For the different levels in the extraction hierarchy, we achieved ~95% for almost all levels. This seems very promising for automatic highlights extraction in sports video. Hence, we achieved our goal.

# Bibliography

[1] Kamesh Namuduri. “Automatic extraction of highlights from a cricket video using MPEG-7 descriptors”.

[2] Jinjun Wang, Changsheng Xu, Engsiong Chng, Qi Tian. “Sports Highlight Detection from Keyword Sequences Using HMM”, in *Proceedings of the International Conference on Multimedia and Expo,* 2004.

[3] Chih-Cheih Cheng, Chiou-Ting Hsu. “Fusion of Audio and Motion Infromation on HMM-Based Highlight Extraction for Baseball Games”, in *Proceedings of the IEEE Transactions on Multimedia, vol. 8, no. 3,* June 2006.

[4] Hao Tang, Vivek Kwatra, Mehmet Emre Sargin, Ullas Gargi. “Detecting Highlights in Sports Videos: Cricket as a test case”, 2011.

[5] Maheshkumar H. Kolekar, Somnath Sengupta. “Semantic concept mining in cricket videos for automated highlight generation”, 2009.

[6] M. H. Kolekar, K. Palaniappan, S. Sengupta. “Semantic Event Detection and Classification in Cricket Video Sequence”, in *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing,* 2008.

[7] Dipen Rughwani. “Shot Classification and Semantic Query Processing on Broadcast Cricket Videos”. http://cse.iitk.ac.in/~vision/dipen/.

[8] N. Harikrishna, Sanjeev Satheesh, S. Dinesh Sriram, K.S. Easwarakumar. “Temporal Classification of Events in Cricket Videos”, 2011.

# Disclaimer

1. All the results shown using approach in [6] are based on the code we developed using the suggested approach, and our dataset.
2. The dataset were frames(>37000) from first 10 overs of Australia-Sri Lanka match provided by Dipen Rughwani[7].
3. All the code was developed by us and not taken from anywhere **except** the code for Level-1(audio analysis)- zero-crossing rate and short-time energy was taken from matlab website, **but** the code for testing was written by us again.
http://www.mathworks.in/matlabcentral/fileexchange/23571-short-time-energy-and-zero-crossing-rate

**Note:**

All code used for testing purposes is available here -
http://home.iitk.ac.in/~sumedh/cs365/project/code.zip
For other details, refer,
http://home.iitk.ac.in/~sumedh/cs365/project/