Automatic Analysis of Cricket And Soccer Broadcast Videos

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS By Research
in
Computer Science and Engineering

by

Rahul Anand Sharma
201203002
rahul.anand@research.iiit.ac.in

Center for Visual Information Technology (CVIT)
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2016

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Automatic Analysis of Cricket And Soccer Broadcast Videos" by Rahul Anand Sharma, has been carried out under our supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. C.V. Jawahar & Dr. Vineet Gandhi

To my Family, Friends and my Niece

Acknowledgments

This work would not have been possible without the continued support in terms of discussions, encouragements and suggestions of many people. First of all I would like to express my deepest gratitude towards my advisers Dr. C. V. Jawahar and Dr. Vineet Gandhi for their guidance, support, motivation and encouragement at every phase of the project. Many thanks to Dr. Visesh Chari and Dr Pramod Sankar for their invaluable contributions during couse of the project.

I would like to wholeheartedly thank my family, my parents and my sister for their love and unwavering support during my work. I would also like to thank all the CVIT faculty members - Dr. P.J. Narayanan, Dr. Anoop Namboodiri, Dr. Avinash Sharma and Dr. Jayanthi Sivaswamy for their valuable contributions in creating the CVIT and its wonderful environment

Special thanks to Rajan, Phani, Satya and Varun for all of their help on various occasions. I am thankful to my lab mates at IIIT for all the wonderful discussions and their valuable feedbacks Anand, Devendra, Aniket, Vijay, Aditya, Sourabh, Yashaswi, Praveen, Swetha, Koustav, Mohak, Suranjana and many others. Also i profusely thank Ayush, Prateek, Anubhav, Gaurav, Divyanshu and others for keeping me sane throughout the course of my research.

Abstract

In the past recent years, there has been a growing need to understand the semantics in sports games. Use of technology in analyzing player movements and understanding the action on a sports field has been growing in the past few years. Most of the systems today make use of certain tracking devices worn by players or markers with sensors placed around the play area. These trackers or markers are electronic devices that communicate with the cameras or cameramen. Other technologies such as the goal line technology popularly used in soccer helps game referees to make accurate decisions that are often misjudged by mere human perception. The primary challenges in these techniques is to make it cost effective and ease of installation and use. It is not convenient to setup markers and sensors around the playing field or to force players to wear certain recording or communication devices without affecting their natural style of playing. Placing a sensor in the game ball also poses a tricky problem of not altering the physical properties of the ball. Sports recorders and broadcasters are now looking for simple and yet effective solutions to get semantic information from a sports game. The big question here is - Can we get sufficient important data only from a video capture just as a human would without relying on external aids of markers and sensors? With advances in various computer vision algorithms and techniques the goal for the future is to analyze everything from captured video. This kind of solution is obviously more attractive to broadcasting and game recording companies as they dont need to setup extra equipment, or influence the authorities to change the match ball or players outfits.

We propose a set of algorithms that does the task of automatic analysis for broadcast videos for the sports of Cricket and Soccer.. Using our approach we can automatically detect salient events in Soccer, Temporally align Cricket video with corresponding text commentaries, Localize/Register a soccer image and others.

We also compare our algorithms with other state of the art approaches extensively on different datasets for a variety of tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

‘

1.1 Sports Video Analytics

Sports content has always been one of the most popular content available on the web. Due to its huge popularity and appeal it has attracted quite a lot of researchers to explore this field of sports video analytics such as - volleyball [70, 71], basketball [62, 48] ,cricket [59], handball [27], snooker [60],ice-hockey [53], football [80] etc. Areas such as performance assessment of players, strategy analysis which previously attracted attention of only coaches, are now finding application in broadcast and other mainstream media. Recently many researchers have tried to use Computer Vision and Machine Learning techniques for the task of - player tracking in handball[Figure 1.1], shot detections in volleyball, commentary generation for Tennis videos.[Figure 1.2], game analysis of ice-hockey[Figure 1.3]

But many of these solutions require extra equipments in terms of on field cameras or player sensors for this task. We focus primarily on Broadcast videos that are easily available for all sports. Broadcast

Figure 1.1 Sport Analysis in Handball: Player recognition and action recognition are being used to analyze performance statistics of players as well as team strategies

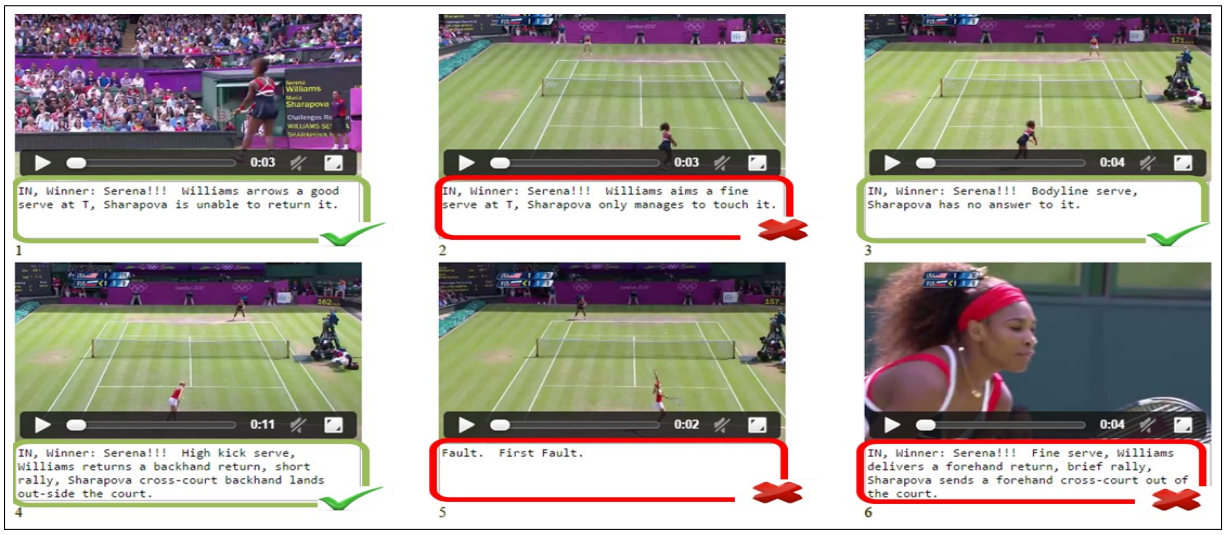


Figure 1.2 Sports Analysis in Tennis: Commentary generation for tennis videos

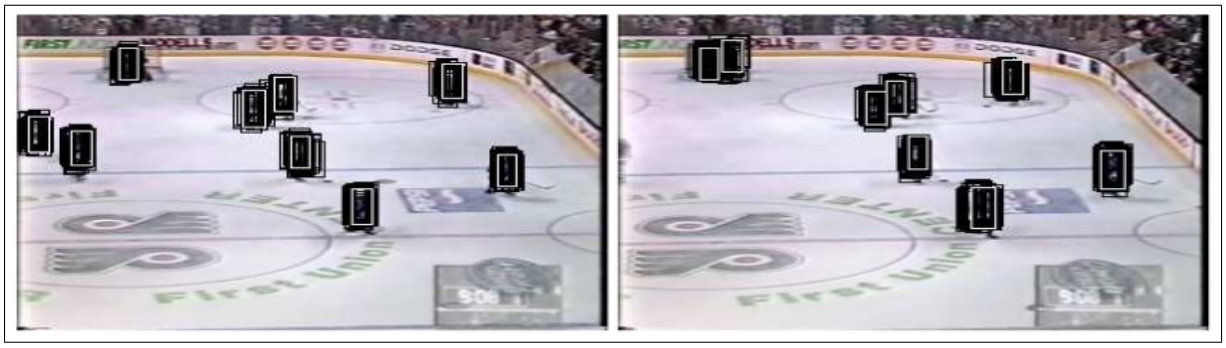


Figure 1.3 Sport Analysis in Ice-Hockey: Player recognition and tracking on field.

sports videos in general have very few portions of game that are interesting to viewers. These parts of video often have associated high level concepts such as Goal in Soccer, Six in Cricket etc. The detection and extraction of these events is termed as "Semantic Analysis" of sports video. The other type of analysis is named as "Tactical Analysis" of sports video which aims to discover strategies and tactic patterns in sports videos. Professional teams often use services of commercial systems such as "*Prozone* for Soccer" and others to do this kind of analysis for them. They use combination of sensory data from players as well as extra fixed cameras installed in stadiums to collect the data and manual alignment/tracking of players is then done to get these high level statistics about players and their strategies. Since these set up require manual intervention and extra equipments, these approaches are not feasible for most of the sports matches. In our work we propose a set of solutions that try and extract these high level statistics given just broadcast sports videos.

1.2 Problems of Interest

We chose the sport of Cricket for our research because of its huge popularity in Indian subcontinent and lack of existing practical research in this domain. Cricket is unique in terms of its huge playing area, wide camera angles, varying zooms, crowd shots, advertisements after every over, hence many of existing solutions in this domain can't be applied to cricket directly. Moreover we also have associated textual commentaries for cricket that our solution make use of to get fine grain annotations of cricket events. The next section will describe challenges associated with Cricket Analytics and how our proposed algorithm tried to address these issues.

1.2.1 Cricket Analytics

The recognition of human activities is one of the key problems in video understanding. Action recognition is challenging even for specific categories of videos, such as sports, that contain only a small set of actions. Interestingly, sports videos are accompanied by detailed commentaries available on line, which could be used to perform action annotation in a weakly-supervised setting. For the specific case of Cricket videos, we address the challenge of temporal segmentation and annotation of actions with semantic descriptions. Our solution for Cricket videos consists of two stages. In the first stage, the video is segmented into "scenes", by utilizing the scene category information extracted from text-commentary. The second stage consists of classifying video-shots as well as the phrases in the textual description into various categories. The relevant phrases are then suitably mapped to the video-shots. The novel aspect of this work is the fine temporal scale at which semantic information is assigned to the video. As a result of our approach, we enable retrieval of specific actions that last only a few seconds, from several hours of video. This solution yields a large number of labelled exemplars, with no manual effort, that could be used by machine learning algorithms to learn complex actions.

Unlike Cricket the textual commentary data is not available for Soccer Matches. So we have to look at some other techniques to automatically analyze the soccer videos. In next section we will briefly describe the proposed algorithms for the task of Event Detection and Top-view registration in Soccer videos.

1.2.2 Soccer Analytics

Soccer is a very fast paced game and because of its huge popularity has attracted researchers from all over the world. Many of the solutions in this domain make use of the data captured using external equipments such as fixed cameras in the stadium, sensors in players boots etc. Inspired by the success of these solutions we try to solve the similar problem using only broadcast soccer videos. Broadcast videos are typically compiled using input sources from various moving or fixed cameras and as such as have very strong contextual priors.As discussed in more detail in following section our first solution in

this domain harnesses this contextual data and show promising results for the task of event detection in soccer videos.

1.2.2.1 Event Detection

Typical Soccer Video contains various salient events such as Goal, Corner, Foul etc. The motivation here is to, Given a Soccer broadcast video assign each frame a label out of the five classes which are Goal, Foul, Corner, Substitution and Gameplay. The presence of standard video editing practices in broadcast sports videos, like football, effectively mean that such videos have stronger contextual priors than most generic videos. In this paper, we show that such information can be harnessed for automatic analysis of sports videos. Specifically, given an input video, we output per frame information about camera angles and the events (goal, foul, etc.). Our main insight is that in the presence of *temporal* context (camera angles) for a video, the problem of event tagging (fouls, corners, goals etc.) can be cast as *per frame* multi-class classification problem. We show that even with simple classifiers like linear SVM, we get significant improvement in the event tagging task when contextual information is included. We present extensive results for 10 matches from the recently concluded football world cup, to demonstrate the effectiveness of our approach.

Given a broadcast frame it is a challenging task to identify actual position of stadium where an event is happening. This problem is quite popular in other sports as well such as basketball, baseball etc. and is typically formulated as registering a broadcast frame with static top-view model. In the following section we will briefly describe our proposed approach for top-view registration in soccer videos.

1.2.2.2 Top-View Registration

The task of registering video frames with a static model is a prevalent problem in computer vision. Most automated methods approach this problem by computing point correspondences between the input image and the model, which are then used to numerically determine the projective transform. However, this standard approach struggles in scenarios where sufficient number of interest point correspondences are unavailable. In this paper, we investigate an alternate approach exploiting the edge information and demonstrate its success in a specific scenario of registering football broadcast video frames on the static top view model of the playing surface. We formulate this registration problem as a nearest neighbor search over a synthetically generated dictionary of edge map and homography pairs. The synthetic dictionary generation allows us to exhaustively cover a wide variety of camera angles and positions and reduce this problem to a minimal edge map matching procedure. Furthermore, we thoroughly study three different methods to match a query image edge map with the ones stored in the dictionary and present extensive results over both synthetic and actual video broadcasts from football World Cup 2014. We also propose couple of optimization procedures to further improve the output and show that our method is able to achieve nearly perfect results.

Figure 1.4 Figure shows typical View point changes in Soccer broadcast Videos. View points in Soccer broadcast videos can be classified into five different categories (from left to right) ground zoom-in,ground zoom-out, top zoom-in, top zoom-out and miscellaneous.

Figure 1.5 Figure shows typical View point changes in Cricket broadcast Videos. View points in Cricket broadcast videos can be classified into bowler run-up, batsmen shot, field, umpire, batsmen close-up, crowd and miscellaneous.

In the following section we will discuss typical challenges that are present in Broadcast Sports videos. Later on we will also discuss some relevant feature extraction/classification techniques that are used in our algorithm.

1.3 Challenges in Broadcast Video Analytics

1.3.1 View Point Changes

Since Broadcast Videos are typically compiled from a variety of different fixed/moving cameras. So typical view point/fixed camera approaches fail to perform given the Broadcast videos. Some examples of view point changes for Soccer and Cricket are presented in Figures 1.4 and 1.5

1.3.2 Labelled Data Scarcity

Due to the challenging nature of the problem, there are not many publically available datasets for Broadcast Videos. Typical datasets available for action recognition can not be used directly as they are usually captured using a single moving camera and doesn't capture typical camera movements in Broadcast sports videos. So for our task of Broadcast sports video analytics we have to create our own datasets.

Figure 1.6 Motion Blur in Soccer: Above figures demonstrates the problem of motion Blur in soccer broadcast Videos.

Figure 1.7 Figure demonstrates an example of advertisement shown during Cricket broadcast Video from IPL 2015 Match.

1.3.3 Motion Blur

Due to constant camera movement, the images typically suffer from motion blur. While introducing new algorithms we have to make sure that we take care of this issue. This motion blur can be due to the movement of players or the cameras. This issue becomes even more significant when we deal with high paced games like Soccer where cameras can move from one end to another in a span on 1-2 seconds

1.3.4 Advertisements

In Example of Broadcast Videos of Cricket we also have commercials after each over. Even though that can be an important cue, it can also cause problem if we don't handle it carefully.

1.3.5 Replays

Typical salient events in Soccer and Cricket Videos are sometimes followed by a replay of the event. Like in case of Goal, Foul, Cornet etc. in a Soccer match or Out, Six, Four etc in Cricket Match we always have replay of the event. This replay looks exactly like an actual event so our algorithm has to take care of the fact that it should not report the same event twice. To take care of replays we make

use of various cues present in broadcast videos. For e.g. we use scorecard detection in Cricket Videos to filter out the replays from the actual event as scorecard is there in original event but is not shown in replays.

1.4 Contributions

The main contributions of this thesis are

1. **Problem:** Given a Cricket Video and corresponding textual Commentary, temporally align them with no manual intervention
   We present a solution for building set of models for different outcomes such as 1 run, Six etc and utilize this models for the task of temporal alignment using Dynamic Programming Alignment.

2. **Problem:**Given a Soccer Match Video automatically identify salient events such as goal, foul etc.
   We propose a feature vector that encodes Contextual information present in a clip and use sliding window SVM as classifier.

3. **Problem:** Register a Broadcast soccer frame with its corresponding Top View
   We propose a nearest neighbor approach that utilizes the dictionary of image-homography pairs created in a semi-supervised manner.

4. **Datasets:** To evaluate our algorithm on all the above defined problem statements we create several datasets. More details about each dataset is explained in corresponding chapters.

Chapter 2

Background

In Section 2.1, we look at the popular linear as well as non-linear feature extraction strategies used for upcoming tasks. Section 2.2, describes various classification techniques used for both action classification and registration in Soccer Videos. The penultimate section describes the optimization techniques used for various tasks. We end the chapter by describing evaluation techniques used for both quantitative and qualitative analysis.

2.1 Feature Extraction

Feature extraction begins from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the learning and generalization, and in some cases leading to better human interpretations. In (majority of) cases it is related to dimensionality reduction and involves reducing the amount of resources required to describe a large set of data. Feature extraction is a general term of constructing combinations of the variables to create a succinct representation and to get around the problems of large memory and power requirements while still describing the data with sufficient accuracy. The extracted features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

First we talk about Histogram of Gradients (HOG) which is quite popular feature extraction technique used for various tasks such as event detection, player tracking etc.

2.1.1 HOG - Histograms of Oriented Gradients

Detecting humans in tennis videos is a challenging task owing to the variability in appearances and poses. We need a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination. Histogram of oriented gradients HOG is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window,

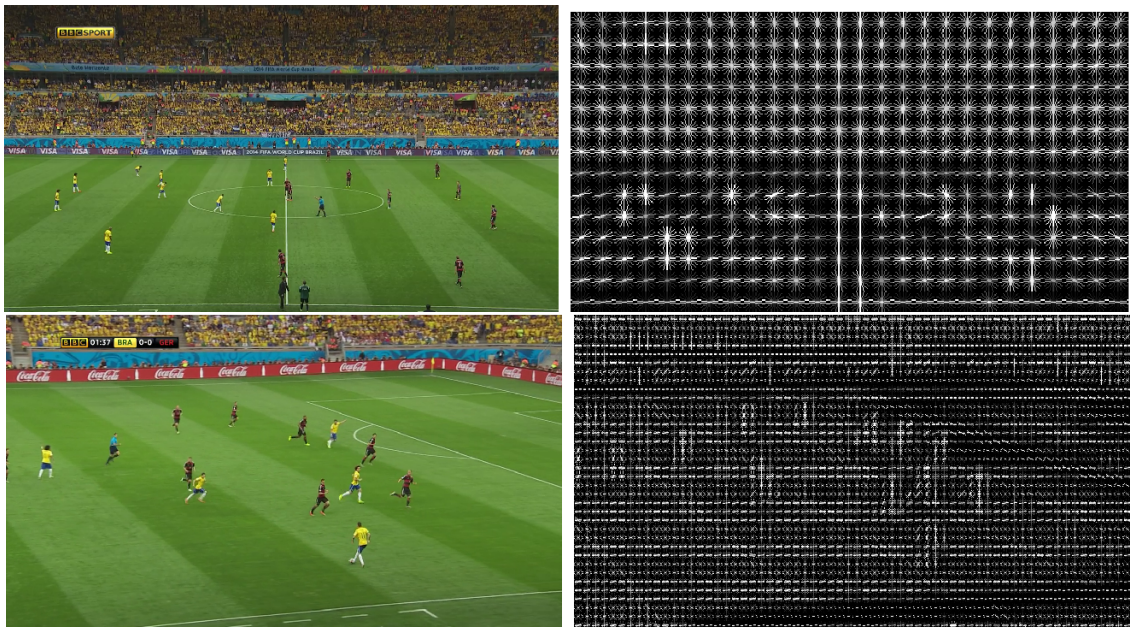


Figure 2.1 Visualization of HOG descriptors: Figures demonstratingHOG descriptors of two broadcast soccer frames

or region of interest. The HOG descriptors are reminiscent of edge orientation histograms SIFT descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance.

HOG divides the input image into square cells of size 'cellSize', fitting as many cells as possible, filling the image domain from the upper-left corner down to the right one. For each row and column, the last cell is at least half contained in the image. Then the image gradient is computed by using central difference (for colour image the channel with the largest gradient at that pixel is used). The gradient is assigned to one of $2 \times numOrientations$ orientation in the range $[0, 2\pi)$ (see Conventions for details). Contributions are then accumulated by using bilinear interpolation to four neighbor cells, as in Scale Invariant Feature Transform SIFT. This results in an histogram h_d of dimension $2 \times numOrientations$, called of directed orientations since it accounts for the direction as well as the orientation of the gradient. A second histogram h_u of undirected orientations of half the size is obtained by folding h_d into two.

Let a block of cell be a 2×2 sub-array of cells. Let the norm of a block be the l_2 norm of the stacking of the respective unoriented histogram. Given a HOG cell, four normalization factors are then obtained as the inverse of the norm of the four blocks that contain the cell. Each histogram h_d is copied four times, normalized using the four different normalization factors, the four vectors are stacked, saturated at 0.2, and finally stored as the descriptor of the cell. This results in a $numOrientations \times 4$ dimensional cell descriptor. Blocks are visited from left to right and top to bottom when forming the final descriptor.

In next section we will talk about Distance Transform. Distance transform is a very popular feature extraction technique that is used for various tasks such as shape based matching, chamfer matching etc. Used along with chamfer matching its a very powerful feature descriptor that encodes shape information of an object.

2.1.2 DT - Distance Transform

A distance transform, also known as distance map or distance field, is a derived representation of a digital image. The choice of the term depends on the point of view on the object in question: whether the initial image is transformed into another representation, or it is simply endowed with an additional map or field. Distance fields can also be signed, in the case where it is important to distinguish whether the point is inside or outside of the shape.[1] The map labels each pixel of the image with the distance to the nearest obstacle pixel. A most common type of obstacle pixel is a boundary pixel in a binary image. See the image for an example of a chessboard distance transform on a binary image.

A distance transformation Usually the transform/map is qualified with the chosen metric. For example, one may speak of Manhattan distance transform, if the underlying metric is Manhattan distance. Common metrics are: Euclidean distance Taxicab geometry, also known as City block distance or Manhattan distance. Chessboard distance Applications are digital image processing (e.g., blurring effects, skeletonizing), motion planning in robotics, and even path finding Uniformly-sampled signed distance fields have been used for GPU-accelerated font smoothing, for example by Valve Corporation researchers.[2] Signed distance fields can also be used for (3D) solid modeling Rendering on typical

GPU hardware requires conversion to polygon meshes, e.g. by the marching cubes algorithm.[3] The distance transform can also be used for font rendering using vectors instead of sampling from texture, as in the open-source project GLyphy

2.2 Classification

The term 'classification' in machine learning is the problem of identifying the category (or class) to which a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. It is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The individual observations are converted into a set feature vectors and thereafter classified into respective category using pre-trained models. An algorithm that implements classification is known as a classifier. Its a mathematical function that maps input data to a category.

SVM's are one of the most popular classifiers that are currently used. It has found its application in various domains such as NLP, Vision, Signal processing etc. Inspired by its success we also use SVM classifier for various classification tasks proposed in our algorithms.

2.2.1 SVM - Support Vector Machines

Given a set K training samples from two lineally separable classes P and N: $\{(\mathbf{x}_k, y_k), k = 1, \cdots, K\}$, where $y_k \in \{1, -1\}$ are class labels. We find a hyper-plane in terms of $\mathbf{w}$ and b, that linearly separates the two classes. For a decision hyper-plane $\mathbf{x}^T\mathbf{w} + b = 0$ to separate the two classes P $(\mathbf{x}_i, 1)$ and N $(\mathbf{x}_i, -1)$, it should satisfy

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 0$$

for both $\mathbf{x}_i \in P$ and $\mathbf{x}_i \in N$. Among all such planes satisfying this condition, we find the optimal one that separates the two classes with the maximal margin (the distance between the decision plane and the closest sample points).

The optimal plane should be in the middle of the two classes, so that the distance from the plane to the closest point on either side is the same. We define two additional planes H_+ and H_- that are parallel to H_0 and go through the point closest to the plane on either side:

$$\mathbf{x}^T\mathbf{w} + b = 1, \quad \text{and} \quad \mathbf{x}^T\mathbf{w} + b = -1$$

All points $\mathbf{x}_i \in P$ on the positive side satisfy $\mathbf{x}_i^T\mathbf{w} + b \geq 1,\ y_i = 1$ and all points $\mathbf{x}_i \in N$ on the negative side satisfy $\mathbf{x}_i^T\mathbf{w} + b \leq -1,\ y_i = -1$. These can be combined into one inequality:

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1, \quad (i = 1, \cdots, m)$$

The equality holds for those points on the planes H_+ or H_-. Such points are called *support vectors*, for which $\mathbf{x}_i^T\mathbf{w} + b = y_i$ i.e., the following holds for all support vectors:

$$b = y_i - \mathbf{x}_i^T\mathbf{w} = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

Moreover, the distances from the origin to the three planes H_-, H_0 and H_+ are, respectively, $|b-1|/||\mathbf{w}||$, $|b|/||\mathbf{w}||$, and $|b+1|/||\mathbf{w}||$, and the distances between planes H_- and H_+ is $2/||\mathbf{w}||$, which is to be maximized. Now the problem of finding the optimal decision plane in terms of $\mathbf{w}$ and b can be formulated as:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}||\mathbf{w}||^2 \qquad \text{(objective function)} \\ &\text{subject to} \quad y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1, \;\; \text{or} \;\; 1 - y_i(\mathbf{x}_i^T\mathbf{w} + b) \leq 0, \quad (i = 1, \cdots, m) \end{aligned}$$

This QP problem is solved by Lagrange multipliers method to minimize

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(\mathbf{x}_i^T\mathbf{w} + b))$$

with respect to $\mathbf{w}$, b and the Lagrange coefficients $\alpha_i \geq 0$ $(i = 1, \cdots, \alpha_m)$. We let

$$\frac{\partial}{\partial W} L_p(\mathbf{w}, b) = 0, \quad \frac{\partial}{\partial b} L_p(\mathbf{w}, b) = 0$$

These leads, respectively, to

$$\mathbf{w} = \sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_j, \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

Substituting these two equations back into the expression of $L(\mathbf{w}, b)$, we get the *dual problem* (with respect to α_i) of the above *primal problem*:

$$\begin{aligned} &\text{maximize} \quad L_d(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T, \mathbf{x}_j \\ &\text{subject to} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

Solving this dual problem (an easier problem than the primal one), we get α_i, from which $\mathbf{w}$ of the optimal plane can be found. Those points $\mathbf{x}_i$ on either of the two planes H_+ and H_- (for which the equality $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ holds) are called *support vectors* and they correspond to positive Lagrange multipliers $\alpha_i > 0$. The training depends only on the support vectors, while all other samples away from the planes H_+ and H_- are not important.

For a support vector $\mathbf{x}_i$ (on the H_- or H_+ plane), the constrained condition is $y_i\left(\mathbf{x}_i^T\mathbf{w} + b\right) = 1$, $(i \in sv)$. Here, sv is a set of all indices of support vectors $\mathbf{x}_i$ (corresponding to $\alpha_i > 0$). Substituting

$$\mathbf{w} = \sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_j = \sum_{j \in sv} \alpha_j y_j \mathbf{x}_j$$

we get

$$y_i(\sum_{j\in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j + b) = 1$$

For the optimal weight vector $\mathbf{w}$ and optimal b, we have:

$$\begin{aligned} ||\mathbf{w}||^2 &= \mathbf{w}^T\mathbf{w} = \sum_{i\in sv} \alpha_i y_i \mathbf{x}_i^T \sum_{j\in sv} \alpha_j y_j \mathbf{x}_j = \sum_{i\in sv} \alpha_i y_i \sum_{j\in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j \\ &= \sum_{i\in sv} \alpha_i (1 - y_i b) = \sum_{i\in sv} \alpha_i - b \sum_{i\in sv} \alpha_i y_i \\ &= \sum_{i\in sv} \alpha_i \end{aligned}$$

The last equality is due to $\sum_{i=1}^{m} \alpha_i y_i = 0$ shown above. The distance between the two margin planes H_+ and H_- is $2/||\mathbf{w}||$, and the margin, the distance between H_+ (or H_-) and the optimal decision plane H_0, is

$$\frac{1}{||\mathbf{w}||} = \left(\sum_{i\in sv} \alpha_i\right)^{-1/2}$$

2.2.1.1 Kernel Mapping: Non-Linear Case

The algorithm above converges only for linearly separable data. If the data set is not linearly separable, we can map the samples $\mathbf{x}$ into a feature space of higher dimensions:

$$\mathbf{x} \longrightarrow \phi(\mathbf{x})$$

in which the classes can be linearly separated. The decision function in the new space becomes:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w} + b = \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x})^T \phi(\mathbf{x}_j)) + b$$

where $\mathbf{w} = \sum_{j=1}^{m} \alpha_j y_j \phi(\mathbf{x}_j)$ and b are the parameters of the decision plane in the new space. As the vectors $\mathbf{x}_i$ appear only in inner products in both the decision function and the learning law, the mapping function $\phi(\mathbf{x})$ does not need to be explicitly specified. Instead, all we need is the inner product of the vectors in the new space. The function $\phi(\mathbf{x})$ is a kernel-induced *implicit* mapping. A kernel is a function that takes two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ as arguments and returns the value of the inner product of their images $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)$$

As only the inner product of the two vectors in the new space is returned, the dimensionality of the new space is not important. The learning algorithm in the kernel space can be obtained by replacing all inner products in the learning algorithm in the original space with the kernels:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w} + b = \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j) + b$$

The parameter b can be found from any support vectors $\mathbf{x}_i$:

$$b = y_i - \phi(\mathbf{x}_i)^T \mathbf{w} = y_i - \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)) = y_i - \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

In next section we will briefly talk about nearest neighbor classifiers. We use nearest neighbor classifier for the task of Top-View registration.

2.2.2 Nearest Neighbors

Nearest neighbors are one of the most classifiers, possibly because it does not involve any training. This technique simply consists of storing all the labeled training examples and given a test images, the label of closest training sample(or majority label of k-neighbors) is assigned to the test image. Nearest neighbors can be easily kernelized or coupled with metric learning. For classification techniques such as LDA or linear SVM, only a single weight vector needs to be stored per class(in a one-vs-rest setting), however, a nearest neighbor classification strategy typically consists of storing all the training samples. This is one of the major demerits of nearest neighbors.

2.3 Graphical Models - Optimization

A graphical model (in probability theory, statisticsparticularly Bayesian statisticsand machine learning) is a probabilistic model for which a graph expresses the conditional dependence structure between random variables. PGMs use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution.

Since we are dealing with visual data there is strong temporal dependencies in the data. To model these kind of dependencies we use Markov Random Fields (MRF)

2.3.1 MRF - Markov Random Fields

A Markov Random Field (MRF) is a graphical model of a joint probability distributions that encode spatial dependencies. It consists of an undirected graph $G = (N, E)$ in which the nodes N represent random variables. Let X_S be the set of random variables associated with the set of nodes S. Then, the edges E encode conditional independence relationships via the following rule: given disjoint subsets of nodes A, B, and C, is conditionally independent of given if there is no path from any node in A to any node in B that doesn't pass through a node of C. The neighbour set N_n of a node n is defined to be the set of nodes that are connected to n via edges in the graph. Given its neighbour set, a node n is independent of all other nodes in the graph (Markov Property). MRF has plentiful applications in both vision and language community [34].

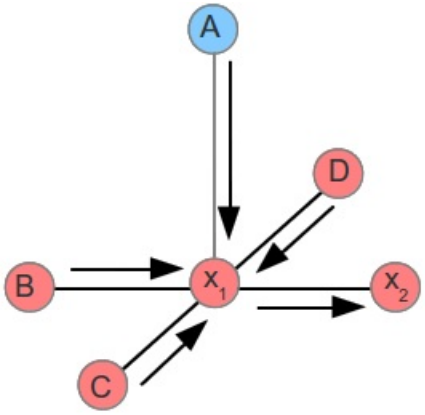


Figure 2.2 Belief Propagation : Figure shows an example of a message passing from x_1 to x_2. A node passes a message to an adjacent node only when it has received all incoming messages, excluding the message from the destination node to itself. x_1 waits for messages from nodes A, B, C, D before sending its message to x_2.

We solve an optimization problem over a MRF network to smoothen our phrase prediction result. An optimization problem is one that involves finding the extrema of a quantity or function. Such problems often arise as a result of a source of uncertainty that precludes the possibility of an exact solution. Optimization in an MRF problem involves finding the maximum of the joint probability over the graph, usually with some of the variables given by some observed data. Equivalently, as can be seen from the equations above, this can be done by minimizing the total energy, which in turn requires the simultaneous minimization of all the clique potentials. Techniques for minimization of the MRF potentials are plentiful. Many of them are also applicable to optimization problems other than MRF. Gradient descent methods are well-known techniques for finding local minima, while the closely-related method of simulated annealing attempts to find a global minimum.

After modeling our dependencies as MRF we use Belief Propagation to find optimal inference from source to destination that minimizes the Unary as well as Pairwise potentials.

2.3.2 BP - Belief propagation for Inference

Belief propagation (sum-product message passing), is a message passing algorithm for performing 'inference' on graphical models – Bayesian networks and Markov random fields. It computes the marginal distribution for each unobserved node, conditional on any observed nodes [78]. BP are presented as message update equations on a factor graph, involving messages between variable nodes and their neighboring factor nodes and vice versa. Considering messages between regions in a graph is one way of generalizing the belief propagation algorithm. There are several ways of defining the set of regions in a graph that can exchange messages, e.g. Kikuchi's cluster variation method. Improvements in

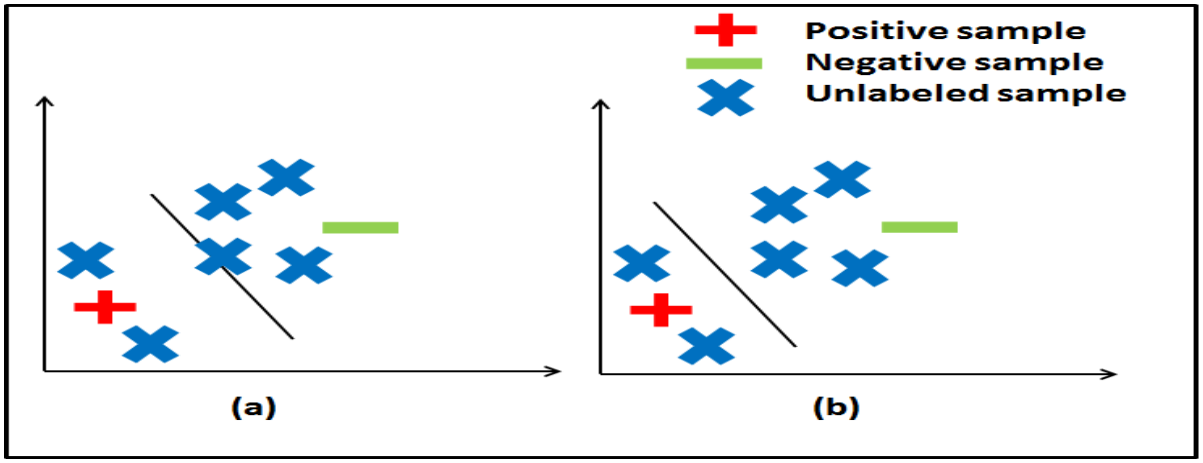


Figure 2.3 Figure shows how taking into account the unlabeled samples may guide the supervised learning algorithm in case only a few labeled samples are available. Fig(a) shows classifier learned using just the labeled examples whereas Fig(b) shows classifier which also considers the unlabeled examples.

the performance of belief propagation algorithms are achievable by breaking the replicas symmetry in the distributions of the fields (messages).

The original belief propagation algorithm was proposed by Pearl in 1988 for finding exact marginals on trees. Trees are graphs that contain no loops. It turns out the same algorithm can be applied to general graphs, those that contain loops, hence the 'loopy BP' [50]. LBP is a message passing algorithm. A node passes a message to an adjacent node only when it has received all incoming messages, excluding the message from the destination node to itself. The choice of using cost/penalty or probabilities is dependent on the choice of the MRF energy formulation. LBP is an iterative method. It runs for a fixed number of iterations or terminate when the change in energy drops below a threshold.

Often times we don't have sufficient labelled data for our algorithms. We then use two approaches to solve this problem a.) Semi-Supervised Learning to automatically generate a large dataset given very few manually labelled samples b.) Transfer Learning to use features learned on a large dataset for some other task and adapt them to use it for our task.

2.3.3 Semi-Supervised Learning

The goal of semi-supervised learning strategies is to utilize labeled as well as unlabeled data in order to obtain better performance than that obtained by using just the labeled data. These strategies are useful in the scenarios where a large amount of unlabeled data is also available along with the labeled data. The large amount of unlabeled images and videos, available over the internet, makes this scenario very relevant for various computer vision tasks. In Figure 2.3, we present the general idea behind a semi-supervised classification strategy. Unlabeled data is also considered while learning the classifier.

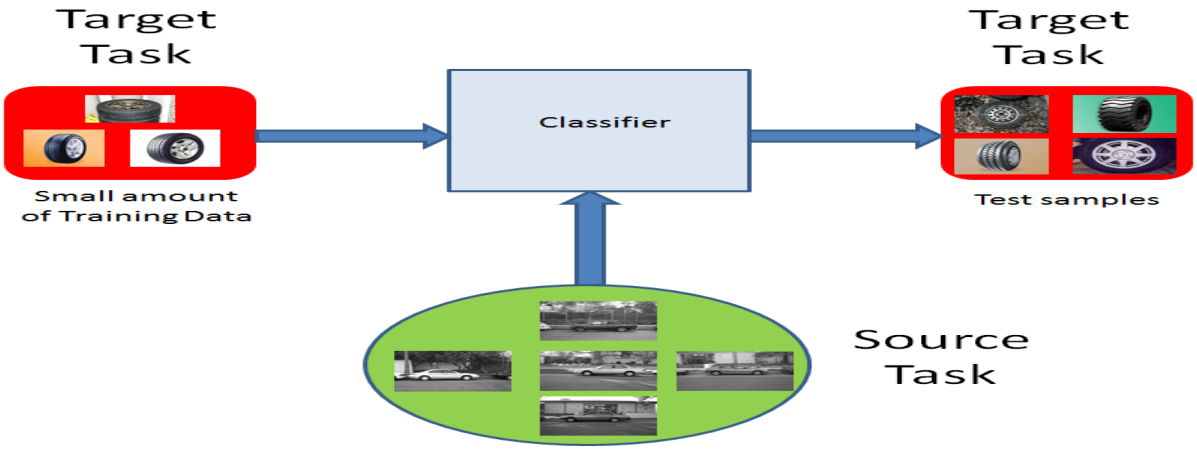


Figure 2.4 Figure shows central idea behind transfer learning. Knowledge from source task, in the form of the trained source classifiers, can be utilized alongwith data from related target task to learn classifers that perform well on the target task.The source and the target task should be related, in order for the transfer to work well.

Semi-supervised algorithms can be broadly classified into two types

- Algorithms which allow usage of unlabeled data in supervised approaches, for example, self-training algorithms [81].
- Algorithms which allow for use some form of supervision in originally unsupervised algorithms, for example constrained clustering algorithms such as constrained k-means [14].

Self-training models are a popular class of semi-supervised classification strategies. In self-training models, a classifier is trained using the labeled examples and the unlabeled examples over which the classifier is confident are labeled using the classifier output. These examples are then added to the labeled set and the classifier is retrained and the entire process is repeated a number of times. For a detailed description of other semi-supervised learning strategies, we refer the reader to Zhu and Goldberg [81].

Constrained k-means algorithms [14] typically consist of must-link and cannot-link constraints. As the name suggests, must-link constraints enumerate such samples which must always occur in the same clusters, and cannot-link constraints specify those samples which must never be assigned to the same clusters.

2.3.4 Transfer Learning

Consider the object detection task in computer vision. Assume we have already trained a detector for one object category, say car and want to learn the detector for a new category, say car tyres. As the two categories share some similarities, i.e. all the cars have tyres, it might be a good idea to utilize the

car detector in some way while learning the car-tyre detector. Using this prior knowledge might reduce the number of labeled examples needed to train the car-tyre detector. This idea has been presented in Figure 2.4.

Transfer learning techniques deal with transferring knowledge learned from one or more source tasks to a related target task. If application of a transfer learning strategy improves the classifier performance then *positive transfer* is said to occur whereas if the classifier performance deteriorates then *negative transfer* is said to occur. One of the challenges of transfer learning is to allow for positive transfer for related tasks while keeping in check the negative transfer between the unrelated tasks.

Transfer learning techniques have been presented for computer vision tasks such as object category recognition [7]. In this work, while training a support vector machine(SVM) for a new category by using just a few examples, the SVM classifiers trained previously for related categories are used to provide regularization.

2.4 Discussion

We have discussed and explained various feature learning and classification techniques in the above sections. Many of these techniques are good for a particular problem and may not be so good for other. Keeping in mind pros and cons of these techniques we judicially use these based on the problem at hand.

In Chapter 5 we use semi-supervised learning technique to generate a very large dictionary using only a very few set of manual initializations. We then use HOG, DISTANCE TRANSFORM and Deep Transfer Learned feature (On Imagenet task) as feature vector and use nearest neighbour algorithm to find closest top-view registration given a broadcast frame.

In Chapter 4 we use Bag of Visual words as a feature vector and multi-class SVM as classifier. We then formulate it as MRF optimization to enforce temporal consistency. Belief propagation algorithm is then used to get coherent temporal segmentation of a video

In Chapter 3 we use our own hand crafted features along with Belief propagation algorithm to automatically align video with corresponding textual information. Moreover we also use Bag of Visual words as a feature vector and SVM classifier for the task of shot classification. More details about the features used and how we do video-text alignment is discussed in the following chapter.

Chapter 3

Fine Grain Annotation of Cricket Videos

The labeling of human actions in videos is a challenging problem for computer vision systems. There are three difficult tasks that need to be solved to perform action recognition: 1) identification of the sequence of frames that involve an action performed, 2) localisation of the person performing the action and 3) recognition of the pixel information to assign a semantic label. While each of these tasks could be solved independently, there are few robust solutions for their joint inference in generic videos.

Certain categories of videos, such as Movies, News feeds, Sports videos, etc. contain domain specific cues that could be exploited towards better understanding of the visual content. For example, the appearance of a Basketball court [66] could help in locating and tracking players and their movements. However, current visual recognition solutions have only seen limited success towards fine-grain action classification. For example, it is difficult to automatically distinguish a "forehand" from a "half-volley" in Tennis. Further, automatic generation of semantic descriptions is a much harder task, with only limited success in the image domain [32].

Instead of addressing the problem using visual analysis alone, several researchers proposed to utilize relevant *parallel* information to build better solutions [17]. For example, the scripts available for movies provide a weak supervision to perform person [17] and action recognition [40]. Similar parallel text in sports was previously used to detect events in soccer videos, and index them for efficient retrieval [74, 79]. Gupta *et al.* [25] learn a graphical model from annotated baseball videos that could then be used to generate captions automatically. Their generated captions, however, are not semantically rich. Lu *et al.* [43] show that the weak supervision from parallel text results in superior player identification and tracking in Basketball videos.

In this work, we aim to label the actions of players in *Cricket* videos using parallel information in the form of online text-commentaries [1]. The goal is to label the video at the shot-level with the semantic descriptions of actions and activities. Two challenges need to be addressed towards this goal. Firstly, the visual and textual modalities are not aligned, i.e. we are given a few pages of text for a four hour video with no other synchronisation information. Secondly, the text-commentaries are very descriptive, where they assume that the person reading the commentary understands the keywords being used. Bridging the semantic gap with visual data is much tougher than, for example, images and object categories.

Batsman: *Gambhir*
Description: *"he pulls it from outside off stump and just manages to clear the deep square leg rope"*

Figure 3.1 The goal of this work is to annotate Cricket videos with semantic descriptions at a fine-grain spatio-temporal scale. In this example, the batsman action of a "pull-shot", a particular manner of hitting the ball, is labelled accurately as a result of our solution. Such a semantic description is impossible to obtain using current visual-recognition techniques alone. The action shown here lasts a mere 35 frames (1.2 seconds).

Bowler Run-Up | Batsman Stroke | Ball-in-the-air (Sky) | Umpire Signal | Replay | Crowd Reaction

Figure 3.2 Typical visuals observed in a *scene* of a Cricket match. Each event begins with the *Bowler* running to throw the ball, which is hit by the *Batsman*. The event unfolds depending on the batsman's stroke. The outcome of this particular scene is *6-Runs*. While the first few shots contain the real action, the rest of the visuals have little value in post-hoc browsing.

3.1 Our Solution

We present a two-stage solution for the problem of fine-grained Cricket video segmentation and annotation. The first stage consists of a joint synchronisation and segmentation of the video with the text commentary. At this stage, the goal is to *align* the two modalities at a "scene" level. Each scene is a meaningful event that is a few minutes long (Figure 3.2), and described by a small set of sentences in the commentary (Figure 3.3). The solution for this stage is inspired from the approach proposed in [56], and presented in Section 3.2.

Given the scene segmentation and the description for each scene, the next step is to align the individual descriptions with their corresponding visuals. At this stage, the alignment is performed between the video-shots and *phrases* of the text commentary. This is achieved by classifying video-shots and phrases into a known set of categories, which allows them to be mapped easily across the modalities, as described in Section 3.3. As an outcome of this step, we could obtain fine-grain annotation of player actions, such as those presented in Figure 3.1.

Our experiments, detailed in Section 3.4 demonstrate that the proposed solution is sufficiently reliable to address this seemingly challenging task. As a consequence of this work, we could build a

5.6 Muralitharan to Gambhir, 2 runs, that hurried into the left-hander with the angle, he went back to drag a pull in the air over square leg, Gambhir has to dive in for the second, better throw would have had him in trouble, it was a wide one

Figure 3.3 An example snippet of commentary obtained from Cricinfo.com. The commentary follows the format: event number and player involved along with the outcome (2 runs). Following this is the descriptive commentary: (red) bowler actions, (blue) batsman action and (green) other player actions.

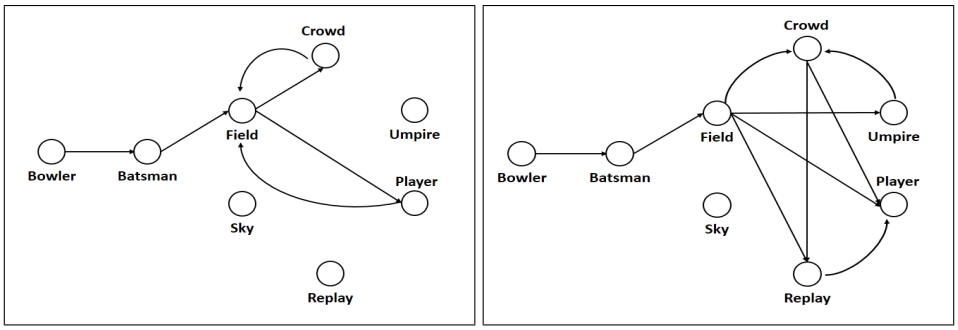


Figure 3.4 State transition diagrams for two scene categories: (left) One Run and (right) Four-Runs. Each shot is classified into one of the given states. Only the prominent state-transitions are shown, each transition is associated with a probability (not shown for brevity). Notice how the one-run scene includes only a few states and transitions, while the four-run model involves a variety of visuals. However, the "sky" state is rarely visited in a Four, but is typically seen in Six-Runs and Out models.

retrieval system that can search across hundreds of hours of content for specific actions that last only a few seconds.

3.2 Scene Segmentation

A typical scene in a Cricket match follows the sequence of events depicted in Figure 3.2. A scene always begins with the *bowler* (similar to a pitcher in Baseball) running towards and throwing the ball at the *batsman*, who then plays his *stroke*. The events that follow vary depending on the manner of the batsman's hit. Each such scene is described in the text-commentary as shown in Figure 3.3. The commentary consists of the event number, which is not a time-stamp; the player names, which are hard to recognise; and detailed descriptions that are hard to automatically interpret.

It was observed in [56] that the visual-temporal patterns of the scenes are conditioned on the *outcome* of the event. In other words, a *1-Run* outcome is visually different from a *4-Run* outcome. This can be observed in the state-transition diagrams in Figure 3.4. When a given scene is segmented into distinct shots, each shot can be represented by mid-level features such as *ground*, *sky*, *play-area*, *players*, etc. For a typical Four-Runs video, the number of shots and their transitions are lot more complex than that of a 1-Run video. Several shot classes such as *replay* are typically absent for a 1-Run scene, while a replay is expected as the third or fourth shot in a Four-Runs scene.

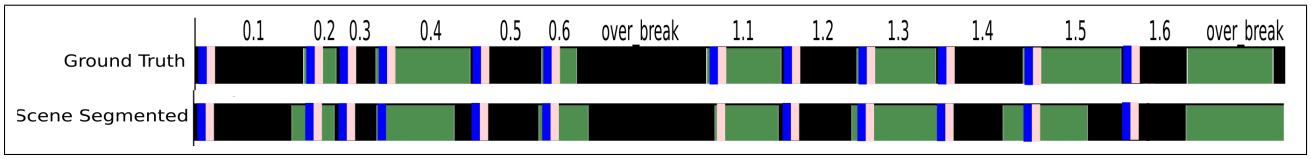


Figure 3.5 Results of Scene Segmentation depicted over two-*overs*. Each scene is alternatively colored black and green.The shot consisting of the bowler and batsman actions are given in blue and pink respectively. Some of the bowler and batsman shots were not recognized correctly, hence missing from the bottom row.

The presence of the outcome in the commentary could now provide the weak-supervision for locating the scene in the video. While the visual model described above could be useful in recognizing the scene category for a given a video segment, it cannot be immediately used to *spot* the scene in a full-length video. Further, the temporal segmentation of the full-length video is not possible without using an appropriate model for the individual scenes themselves. This chicken-and-egg problem can be solved by utilizing the scene-category information from the parallel text-commentary.

The textual information provides an additional prior on the expected sequence of events in the given video. This allows us to build a model for the entire video, by concatenating the models for each event category in the sequence. Given that this model of the video H, closely resembles the observed data V, the two sequences should align closely. Let us say F_i represents one of the N frames and S_k represent the category of the k-th scene. The goal of the scene segmentation is to identify anchor frames F_i, F_j, which are most likely to contain the scene S_k. The optimal segmentation of the video can be defined by the recursive function

$$C(F_i, S_k) = \max_{j \in [i+1, N]} \{p(S_k \mid [F_i, F_j]) + C(F_{j+1}, S_{k+1})\},$$

where $p(S_k \mid [F_i, F_j])$ is the probability that the sequence of frames $[F_i, F_j]$ belong to the scene category S_k. This probability is computed by matching the learnt scene models with the sequence of features for the given frame set. The optimisation function could be solved using Dynamic Programming (DP). The optimal solution is found by backtracking the DP matrix, which provides the scene anchor points $F_{S_1}, F_{S_2}, ..., F_{S_K}$.

We thus obtain a temporal segmentation of the given video into its individual scenes. A typical segmentation covering 12 scenes is shown in Figure 3.7. Further, the descriptive commentary from the parallel-text could be used to annotate the scenes, for text-based search and retrieval. However, in this work, we would like to annotate at a much finer temporal scale than the scenes, i.e., we would like to annotate at the shot-level. The solution towards shot-level annotation is presented in the following section.

3.3 Shot/Phrase Alignment

Following the scene segmentation, we obtain an alignment between minute-long clips with a paragraph of text. To perform fine-grained annotation of the video, we must segment both the video clip and

Algorithm 1 Train_Model

Find average length L_{s_i} of the Videos
Set $S_s = NULL$
for $V_i = V_1$ to V_n **do**
    Identify shots $C_{i_1}, C_{i_2}...C_{i_n}$ in V_i
    for each shot $j = 1$ to m and each shot-class $k = 1$ to i **do**
        Find probability $P_k(C_{i_j})$, of shot C_{i_j} belonging to the kth shot-class
    end for
    /*Build scene representation S_{s_i} as */
    for $j = 1$ to L_{v_i} and each shot-class $k = 1$ to 1 **do**
        Append $P_k(C_{i_j})$ to $S_{s_{i_k}}$
    end for
    Scale $S_{s_{i_k}}$ to average Length L_{s_i}
    For Each k = 1 to 1, Append $S_{s_{i_k}} to S_{s_i}$
end for
Average S_{s_i} over $i = 1$ to n to obtain S_s
return S_s

Algorithm 2 Generate Video Representation (Match_Commentary)

Set $G = NULL$
Parse $Match\_Commentary$ and identify the Ball B_i and their corresponding Outcomes O_i
for cach ball i = 1 to n **do**
    Identify Scene Model S_s for O_i
    Append S_s to G
end for
return G

Algorithm 3 Segment_Video (V,G)

Set $S = NULL$
Identify shots $C_{i_1}, C_{i_2}...C_{i_n}$ in V
for each shot $j = 1$ to m and each shot-class $k = 1$ to 1 **do**
    Find probability $P_k(C_j)$, of shot C_j belonging to the kth shot-class
end for
/*Build scene representation S as */
for $j = 1$ to L_{v_i} and each shot-class $k = 1$ to 1 **do**
    Append $P_k(C_j)$ to S_k
end for
Compute D = Dynamic_Programming(S, G)
Find optimal path using P = Back_Track(D)
/*Segment Video*/
for each scene $s = 1$ to n **do**
    Find the scene segment G_s corresponding to s in generated video G
    Find correspondence of G_s in P
    Output V_s corresponding to G_s in P
    Annotate V_s with the scene description of s
end for

the descriptive text. The scenes could be segmented into shots, such that it is unlikely to map multiple actions into the same shot. Given that the commentary is free-flowing, the action descriptions need to be identified at a finer-grain than the sentence level. Hence we choose to operate at the phrase-level, by segmenting sentences at all punctuation marks. Both the video-shots and the phrases are classified into one of these categories: {*Bowler RunUp, Batsman Stroke, Others*}, by learning suitable classifiers for each modality. Following this, the individual phrases could be mapped to the video-shots that belong to the same category.

3.3.1 Video-Shot Recognition

In order to ensure that the video-shots *atomic* to an action/activity, we perform an over-segmentation of the video. We use a window-based shot detection scheme that works as follows. For each frame F_i, we compute its difference with every other frame F_j, where $j \in [i - w/2, i + w/2]$, for a chosen window size of w centered on F_i. If the maximum frame difference within this window is greater than a particular threshold τ, we declare F_i as a shot-boundary. We choose a small value for τ to ensure over-segmentation of scenes.

Each shot is now represented with the classical Bag of Visual Words (BoW) approach [64]. SIFT features are first computed for each frame independently, which are then clustered using the K-means clustering algorithm to build a visual vocabulary (where each cluster center corresponds to a visual word). Each frame is then represented by the normalized count of number of SIFT features assigned to each cluster (BoW histograms). The shot is represented by the average BoW histogram over all frames present in the shot. The shots are then classified into one of these classes: {*Bowler Runup, Batsman Stroke, Player Close-Up, Umpire, Ground, Crowd, Animations, Miscellaneous*}. The BoW histogram features are classified using a multiclass Kernel-SVM, into one of these classes.

The individual shot-classification results could be further refined by taking into account the temporal neighbourhood information. Given the strong structure of a Cricket match, the visuals are predictable according to the sequence of events. Such a sequence could be modelled as a Linear Chain Conditional Random Field (LC-CRF) [39]. The LC-CRF, consists of nodes corresponding to each shot, with edges connecting each node with its previous and next node, resulting in a linear chain. The goal of the CRF is to model $P(y_1, ..., y_n|x_1, ..., x_n)$, where x_i and y_i are the input and output sequences respectively. The LC-CRF is posed as the objective function

$$p(Y\|X) = \exp(\sum_{k=1}^{K} a_u(y_k) + \sum_{k=1}^{K-1} a_p(y_k, y_{k+1}))/Z(X)$$

Here, the unary term is given by the class-probabilities produced by the shot classifier, defined as

$$a_u(y_k) = 1 - P(y_k|x_i).$$

The pair-wise term encodes the probability of transitioning from a class y_k to y_{k+1} as

$$a_p(y_k, y_{k+1}) = 1 - P(y_{k+1}|y_k).$$

The function $Z(X)$ is a normalisation factor. The transition probabilities between all pairs of classes are learnt from a training set of labelled videos. The inference of the CRF is performed using the forward-backward algorithm.

3.3.2 Text Classification

The phrase classifier is learnt entirely automatically. We begin by crawling the web for commentaries of about 300 matches and segmenting the text into phrases. It was observed that the name of the bowler or the batsman is sometimes included in the description, for example, "*Sachin* hooks the ball to square-leg". These phrases can accordingly be labelled as belonging to the actions of the Bowler or the Batsman. From the 300 matches that we mine, we obtain about 1500 phrases for bowler actions and about 6000 phrases for the batsman shot. We remove the names of the respective players and represent each phrase as a histogram of its constituent word occurrences. A Linear-SVM is now learnt for the bowler and batsman categories over this bag-of-words representation. Given a test phrase, the SVM provides a confidence for it to belong to either of the two classes.

The text classification module is evaluated using 2-fold cross validation over the 7500 phrase dataset. We obtain a recognition accuracy of **89.09%** for phrases assigned to the right class of bowler or batsman action.

3.4 Experiments

3.4.1 Dataset:

Our dataset is collected from the YouTube channel for the Indian Premier League(IPL) tournament. The format for this tournament is 20-overs per side, which results in about 120 scenes or events for each team. The dataset consists of 16 matches, amounting to about 64 hours of footage. Four matches were groundtruthed manually at the scene and shot level, two of which are used for training and the other two for testing.

3.4.2 Scene Segmentation

For the text-driven scene segmentation module, we use these mid-level features: {*Pitch, Ground, Sky, Player-Closeup, Scorecard*}. These features are modelled using binned color histograms. Each frame is then represented by the fraction of pixels that contain each of these concepts, the scene is thus a spatio-temporal model that accumulates these scores.

The limitation with the DP formulation is the amount of memory available to store the DP score and indicator matrices. With our machines we are limited to performing the DP over 100K frames, which amounts to 60 scenes, or 10-*overs* of the match.

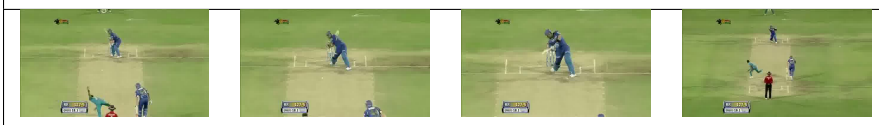
	on the pads once more Dravid looks to nudge it to square leg off the pads for a leg-bye
	that is again slammed towards long-on
	shortish and swinging away lots of bounce

Figure 3.6 Examples of shots correctly labeled by the batsman or bowler actions. The textual annotation is semantically rich, since it is obtained from human generated content. These annotated videos could now be used as training data for action recognition modules.

Kernel	Linear	Polynomial	RBF	Sigmoid
Vocab: 300	78.02	80.15	81	77.88
Vocab: 1000	**82.25**	81.16	82.15	80.53

Table 3.1 Evaluation of the video-shot recognition accuracy. A visual vocabulary using 1000 clusters of SIFT-features yields a considerably good performance, with the Linear-SVM.

The accuracy of the scene segmentation is measured as the number of video-shots that are present in the correct scene segment. We obtain a segmentation accuracy of **83.45%**. Example segmentation results for two scenes are presented in Figure 3.7, one can notice that the inferred scene boundaries are very close to the groundtruth. We observe that the errors in segmentation typically occur due to events that are not modelled, such as a player injury or an extended team huddle.

3.4.3 Shot Recognition

The accuracy of the shot recognition using various feature representations and SVM Kernels is given in Table 3.1. We observe that the 1000 size vocabulary works better than 300. The Linear Kernel seems to suffice to learn the decision boundary, with a best-case accuracy of 82.25%. By the application of the CRF based refinement of the SVM predictions given by this setting, an improved accuracy of 86.54. Specifically, the accuracy of the batsman-shot and the bowler-shot, the categories that are of most interest for our work is 89.68%. Examples of shot recognition for these categories is presented in Figure 3.7.

3.4.4 Shot Annotation Results

The goal of the shot annotation module is to identify the right shot within a scene that contains the bowler and batsman actions. As the scene segmentation might contain errors, we perform a search in a shot-neighbourhood centered on the inferred scene boundary. We evaluate the accuracy of finding the right bowler and batsman shots with a neighbourhood region R of the scene boundary, which is given

R	Bowler Shot	Batsman Shot
2	22.15	39.4
4	43.37	47.6
6	69.09	69.6
8	79.94	80.8
10	87.87	88.95

Table 3.2 Evaluation of the neighbourhood of a scene boundary that needs to be searched to find the appropriate bowler and batsman shots in the video. It appears that almost 90% of the correct shots are found within a window size of 10.

in Table 3.2. It was observed that 90% of the bowler and batsman shots were correctly identified by searching within a window of 10 shots on either side of the inferred boundary.

Once the shots are identified, the corresponding textual comments for bowler and batsman actions, are mapped to these video segments. A few shots that were correctly annotated are shown in Figure 3.6.

3.5 Summary

In this chapter, we present a solution that enables rich semantic annotation of Cricket videos at a fine temporal scale. Our approach circumvents technical challenges in visual recognition by utilizing information from online text-commentaries. We obtain a high annotation accuracy, as evaluated over a large video collection. The annotated videos shall be made available for the community for benchmarking, no data has been publicly available so far in such quantity or quality. In future work, one could learn classifiers for fine-grain activities in Cricket from such a dataset.

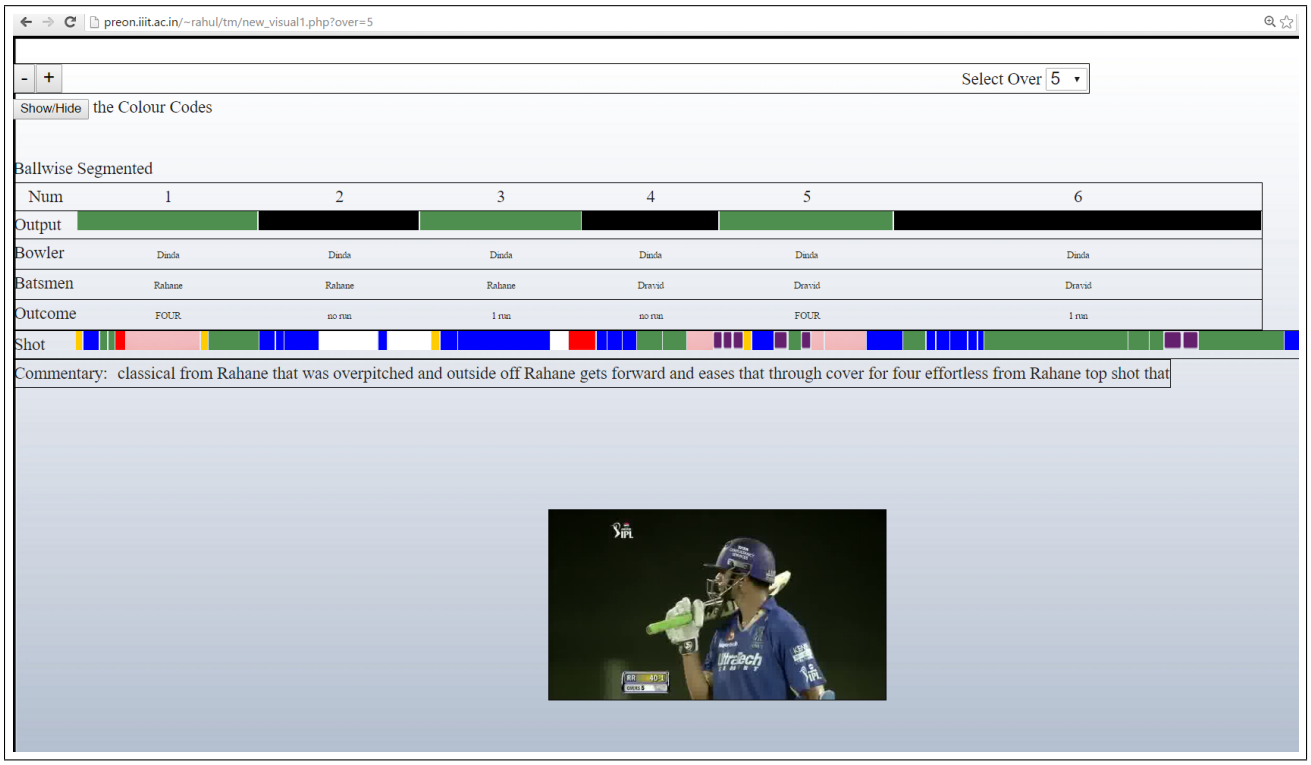


Figure 3.7 Example of Web based visualization tool based on the proposed algorithm.

Chapter 4

Automatic analysis of broadcast football videos using contextual priors

Modern developments in multimedia creation, storage, and compression technologies have paved the way for extensive archiving of video content. Building applications for search, summarization or editing on such large databases requires extensive information about the content of these videos. Current sources of such descriptions are only limited to textual content. Since textual descriptions are both inefficient (descriptions are subjective and vary from person to person) and incomplete (it is difficult to describe all content in a video to facilitate search, summarization or editing), it is important to build tools to automatically analyze video content and identify salient parts, to generate textual and other kinds of descriptions over the timeline. Given the diverse nature of video content on the web, this task is easier said than done. One approach to this problem is to isolate and process videos by genre. Such an approach has a two-fold advantage: 1) each genre could be associated with a set of rules for video creation that might make it easier to design video understanding algorithms 2) it is easier to distinguish between relevant and irrelevant semantic content when information about genre is given (for example, information about crowds in a football match is rarely searched for, and hence can be ignored). Recently, problems related to sports video analysis have particularly received much attention in this direction with many direct applications like automated highlights generation [6] or analysis of team activities and strategies[45], being built upon semantic analysis of video content.

Following these lines, we have looked at the problem of automatic semantic analysis of football broadcast videos. Our work is built on the fact that broadcast videos of football matches are constructed in a very structured manner, thus imposing some useful restrictions on the content. Firstly, there are fixed vantage points in a football field where PTZ cameras are placed for recording such events, thus limiting the number of views. Secondly, the edited video switches between these cameras in strong correlation with events occurring on the field.

This inherent structure of editing in broadcast sports videos motivates us to ask the question, how to define context in such structured settings? We answer this problem based on two key realizations. Firstly, most camera angles associated with the events like goal or corners are pre-determined. For example, in the event of a goal, a broadcast video automatically switches to focus in on player huddles/celebrations, which is unlikely to happen during normal game play. Secondly, this strong asso-

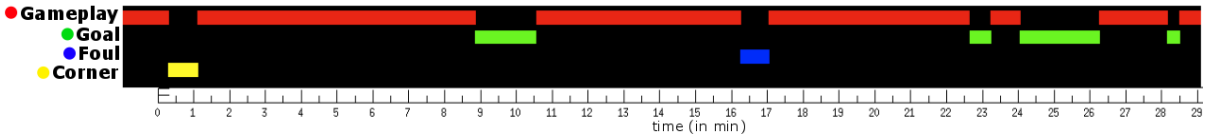


Figure 4.1 An example of event tagging using the proposed approach on the first thirty minutes of the semifinal match between Brazil and Germany in World Cup 2014. The plot shows the occurrence of four different events (Goal, Foul, Corner and Gameplay) over the timeline. The proposed method successfully detects all the goal events.

ciation also extends temporally since editing rarely switches between unrelated camera angles, which is to say that camera angles vary *smoothly* across time, except when shot changes occur. This strong temporal and event correlation with camera angles pro-
mpts us to argue that *contextual* information in sports video analysis can largely be based upon knowledge of camera angles.

Based on the above discussion, we argue that for the problem of automated sports video analysis two main tasks gain prominence. Firstly, classification of images into different camera views forms the initial basis of *contextual* understanding of a sports video. Secondly, the analysis of *events* in such videos gives an almost complete summary of the match (a motivating example is illustrated in Figure 4.1). Accordingly, we focus on these two aspects in the current paper. Furthermore, as evidence of the usefulness of such information, we present an application of context reliant targeted spatial segmentation. Formally, we make the following contributions

- We present an automatic approach to first identify the camera view type for each frame in a video.
- Using predicted view types we then propose an algorithm to accurately predict salient events like goal, foul, corner, substitution etc.
- We also present an application of spatial segmentation that benefits from such contextual information.

4.1 Related Work

Previous work on semantic segmentation of videos has looked into two different directions, first is segmentation based on shots and second is based on events. Shot based temporal segmentation divides the video into smaller segments by identifying transitions from a camera to another. Most existing methods [36][63][47] performs shot segmentation by detecting transitions (cut, fade, dissolve etc) based on difference of features in consecutive frames. The segmented shots are then used as minimal units for further tasks.

Figure 4.2 We classify camera viewpoints into five different categories namely (from left to right) ground zoom-in, ground zoom-out, top zoom-in, top zoom-out and miscellaneous (covering mainly the crowd view).

In [77], Xu et al. classified football video shots into the views of global, zoom-in and close-up using the grass area ratio. In [76], Gu et al. used motion vectors in addition to dominant color to classify different view types in football broadcasts. Duan et al. [15] used a more elaborate set of features fusing motion vectors, color, texture, shape and shot length information for a similar task. These approaches suffer from the drawback of strong reliance on detecting transitions which may not be robust due to strong correlation between different shots (for example, a camera change may not result into difference in frame color histograms which are commonly used features in these approaches). To suppress the negative effects of inaccurate shot boundaries, it is also common to manually label the transition frames [77]. Our work addresses this issue in a more elaborate manner, by first training detectors for different camera types (illustrated in Figure 5.6) and then assigning each frame an unique camera label. By merging consistent camera labels over time our method in turn can be used to obtain shot based segmentation.

On the other hand, event based segmentation divides the video into shorter clips where each clip contains only one event. The granularity of event is lower than that of shot (an event can compass several shots) and there is no standard relation between an event and the number of shots in it. In [57], Qian et al. define an event clip as a set of sequential video frames which begins with a global view (far zoom out view) and ends with non-global views. They further use hidden conditional random fields to classify event clips into five different categories like goal, shoot, normal etc. But such a hard coded definition of an event may not always hold. Sigari et al. [63] segment the video based on camera motion (estimated using block matching between consecutive frames) and uses this information to detect event like counter-attack. Xie et al. [73] used a sliding window approach to classifying the football video into play vs break segments.

Heuristic rule based approaches [73][6] have also been proposed for the detection of event boundaries. These approaches perform low level analysis to detect marks (field, lines, arcs, and goalmouth), player positions, ball position etc. A set of hard coded rules based on these low level features are then used to detect corresponding events. Assfalg et al. [6] used such a rule based approach for identification of salient events like goal, corner, kick off etc.

Recently purely data driven approaches have gained importance in sports broadcast, for example [11] learnt directorial styles by training classifiers on a training set of previous broadcasts. They suggest that such an approach could also be useful to determine the boundary of the salient events. Following these line, we employ a simple bag-of-features [12] representation combined with Support Vector Machines

(SVM) for both view-type and event classification. The advantage of our approach is that it is independent of any ad-hoc rules or hard coded definitions and thus it is more generalized and can be easily extended to different sports.

Multimodel approaches are also common. Previous work [31] has looked at the problem of event detection using fusion of audio visual features. The main idea in these approaches is that the increased audio activity is a cue for important moments in the game. Textual information has also been used with visual features for event detection [75]. These methods typically use the match report and game log obtained from the web as the text source and require the alignment of the web-casting text with the broadcast sports video. On the contrary, the proposed method in this chapter is purely based on visual data.

Recent work [45] has looked at the problem of discovering team behaviours or detecting the locations where the play evolution will proceed [33] by analyzing plan view tracks of the players. The plan view tracks are usually obtained using a set of static cameras manually installed around the corresponding sports field. Although these methods have demonstrated impressive results, they are not applicable for analysis in broadcast videos where only the feed from a single moving camera is available at a given time.

Far zoomed out shots have been used for the context of sports classification [30] and recognition of group activities in football videos [35]. The work in [30] exploits the fact that the playing surface is largely visible in far zoomed out viewpoints and different sports can be classified based on the type of playing surface. In [35] Kong et al. built a local motion descriptor by grouping SIFT keypoint matches into foreground point set and background point set and then used it to classify events like left side attacking or left side defending etc. Applicability to only far zoomed out shots limits these approaches in the case of broadcast videos which require ability to process input from different viewpoints (as illustrated in Figure 5.6). Our approach, on the other hand takes advantage of different view-points and presents a comprehensive solution.

4.2 Method

In this section we discuss the two key components of the proposed method for automatic analysis of football videos, namely, camera viewpoint estimation for each frame of the video and marking salient events on the timeline (event tagging).

4.2.1 Camera viewpoint estimation

Football broadcast videos are usually edited from a set of source videos recorded from different viewpoints. By analysing 52 matches of world cup 2014, we found out that these camera viewpoints can be broadly categorised into five different categories, which are illustrated in Figure 5.6). The inclusion of camera angles (ground or top) accounts for the main difference over the previous works, which segregate

Figure 4.3 Typical viewpoint transitions in a goal event (Top zoom out → Top Zoom in → Ground Zoom in → Ground Zoom out → Miscellaneous). The camera framing changes with respect to both size and angle.

Figure 4.4 We classify events into five different categories namely (from left to right) goal, corner, foul, substitution and gameplay.

the camera view points only on the basis of their shot sizes (for e.g. medium shot or a close up). Although most of the game play is covered by top view cameras located in the crowd area, the ground view cameras (placed at sidelines at field level) plays an important role in viewing experience. They are quite handy during breaks and salient events and are used as standard by all most all broadcasters. An example of camera switches during a goal event is illustrated in Figure 4.3.

Our goal is to automatically predict the camera viewpoint for each frame of an edited football broadcast video, as this provides a strong context for other higher level tasks (for example, the pattern of transitions between camera viewpoints is correlated with different salient events as illustrated in Figure 4.3). We approach this problem by learning a per frame multi-class classifier.

4.2.2 Frame representation:

Each frame is represented using the classical Bag of Words (BoW) approach [12]. SIFT features are first computed for each frame independently, which are then clustered using k-means clustering algorithm to build a visual vocabulary (where each cluster corresponds to a visual word). Each frame is then represented by the normalized count of number of SIFT features assigned to each cluster (BoW histograms). The length of the feature vector is equal to the number of clusters.

Using the BoW feature representation, our method learns a multi class SVM to classify different camera labels. Training is performed by manually annotating a set of frames with corresponding camera labels. The classification is performed per frame basis but for classifying a frame we consider features from a temporal window of 40 frames centred around it.

We further assume that the camera transitions (cuts) are smooth and each camera viewpoint is maintained for a minimum amount of time before cutting to a different camera. This is a fair assumption, as fast and abrupt camera transitions may appear disturbing to the viewer and are avoided by the ex-

pert editors [68]. We benefit from this assumption to smooth out the noise in per frame camera label prediction, using a Markov Random Field (MRF) optimization approach.

The optimization method takes as input the multi-class SVM scores for every frame t of the video. It outputs a sequence $\xi = \{s_t\}$ of camera labels $s_t \in [1 : M]$, where M is the total possible camera labels (five in our case) for all frames $t = \{1 : N\}$. We minimize the following global cost function:

$$E(\xi) = \sum_{t=1}^{N} E_d(s_t) + \sum_{t=2}^{N} E_s(s_{t-1}, s_t). \tag{4.1}$$

The cost function consists of a data term E_d that measures the evidence of the object state given the SVM scores and a smoothness term E_s which penalizes camera transitions. The data term and the smoothness term are defined as follows:

$$E_d(s_t) = -\log(P(s_t, t)) \ . \tag{4.2}$$

Here, $P(s_t, t)$ is the SVM classification score for camera label s_t at frame t. And,

$$E_s(s_{t-1}, s_t) = \begin{cases} 0 & \text{if } s_{t-1} = s_t \ , \\ \lambda & \text{otherwise} \ . \end{cases} \tag{4.3}$$

Where λ is a constant, which is determined empirically. Finally, we use dynamic programming (DP) to solve the optimization problem presented in Equation 5.4. The DP algorithm constructs a graph with $M \times N$ nodes (M rows, N columns) and computes the minimum cost to reach each node. Finally, we backtrack from the minimum cost node in the last column.

4.2.3 Event Tagging

Given a video clip, the goal of event tagging is to mark all the salient events on the timeline. It is an important problem in sports analysis as it can produce activity description and other high level results (summarization, highlights generation etc). In our work, we consider four different salient events (Goal, Corner, Substitution, Foul) and a gameplay event (which broadly covers rest of the possible events). The selection was motivated by the online textual commentaries where these four salient events are distinctly marked. An instance for of each of the five classes is illustrated in Figure 4.4.

Similar to camera label estimation, we design event tagging problem as a per frame classification task. For classifying a frame, the feature vector is built using a total of 40 frames centred at that frame, where each individual frame is represented using the BoW histograms, camera label and motion features. The BoW histograms capture the correlation between the events and the distribution of features, they are obtained in a similar manner as in previous section. The camera label per frame is defined as a five dimensional boolean vector and captures the correlation of an event with camera transitions.

The motion features represent the correlation of player movements with different events. They are computed from the player tracks obtained using the combination of discriminative trained deformable

part models (DPM) proposed by Fezenswalb et al. [18] and the data association approach of Pirsiavash and Ramanan [55]. The player detections are performed per frame basis using a DPM model specifically trained for the case of football videos.

4.2.4 Training DPM:

While training DPM's, the choice of negative examples strongly impacted the quality of the detector. Using a fixed set of 200 images with manually annotated bounding boxes for players (2120 instances of players) as positive examples and varying the negative examples, we trained three different versions of DPM:

1. Using randomly sampled windows from frames of football videos as negative examples (not overlapping with players and similar in size). Approx 20000 negative examples were used.
2. Using entire VOC 2007 dataset for negative samples (all classes except pedestrians)
3. Using both VOC 2007 dataset and randomly sampled windows from football videos

We then compared the detection performance of these three cases with the pre trained pedestrian detector on a set of 75 test images. The DPM model trained with only VOC dataset as negative examples gave best results and was finally used for obtaining the player detections.

The per frame detections are then combined into player tracks using [55]. The result is a set of bounding boxes represented by center (x, y) and height(h) and their corresponding labels (representing different tracks). Player tracks shorter than four frames are ignored. Using the corresponding bounding boxes in consecutive frames, we compute a nine dimensional motion feature vector (mean xy-motion; median xy-motion; average h, median h, min-h, max-h and number of corresponding bounding boxes).

The BoW histograms, camera label and motion features are then concatenated for each frame individually. Finally we perform experiments on event classification using both the mean and concatenation of features from 40 individual frames. The classification is performed using a five class SVM which is trained using manually annotated ground truth data. The per frame classification results are then temporally smoothed in a similar way as in Equation 5.4, penalizing frequent event transitions.

4.3 Experiments

We perform experiments on broadcast video sequences from 10 matches of football world cup 2014. We present results on the camera label estimation and the salient event classification. We make quantitative comparisons in each case using manually annotated ground truth data. Furthermore, using an application of spatial segmentation, we show that even a simple algorithm with the knowledge of camera viewpoint can bring as much as 20% improvement over the state of the art. Each of these experiments are described with detail in the proceeding sections:

	Top-zout	Top-zin	Ground-zout	Ground-zin	Misc
Top-zout	**81.4**	6.9	8.2	3.4	0.1
Top-zin	68.7	**15.9**	9.3	6.1	0
Ground-zout	3	7	**79.7**	10.3	0
Ground-zin	3.3	5.7	70.4	**20.6**	0
Misc	2.5	1.8	33.6	55.7	**6.4**

Table 4.1 Camera label estimation results using dominant color ratio with five different camera viewpoints (percentages)

	Top-zout	Top-zin	Ground-zout	Ground-zin	Misc
Top-zout	**88.8**	2.3	5.9	2.3	0.7
Top-zin	36.2	**25.9**	17.3	17.7	2.9
Ground-zout	33.6	8.9	**38.4**	16.2	2.9
Ground-zin	28.7	11.6	24.4	**32.4**	2.9
Misc	30	14	22	16	**18**

Table 4.2 Camera label estimation results using optical flow.

	Top-zout	Top-zin	Ground-zout	Ground-zin	Misc
Top-zout	**98**	0.7	1.2	0.1	0
Top-zin	0.8	**65.9**	8.1	25.1	0.1
Ground-zout	0.4	3.4	**79.6**	15.1	1.5
Ground-zin	0.1	1.4	19	**78.5**	1
Misc	0	0.1	0.2	4	**95.7**

Table 4.3 Camera label estimation results using our method

4.3.1 Camera label estimation

We manually annotated two 45 minutes videos (324000 frames), from two different matches with camera labels (with each frame assigned an unique label) for the quantitative analysis of camera label estimation. One part was used for training and another was used for testing.

We compare our method with two commonly used approaches from the previous work based on color [51, 77] and the motion vectors [76]. The color based approaches classify shots into different classes based on the ratio of the green color or the color histogram analysis. We implemented a similar color based algorithm using ten bin histograms and classified shots into different categories using ratio of the dominant color (we used dominant color instead of a fixed shade of green to bring further robustness). The ratios for each class were learnt from the training data.

For motion vector based classification, we learnt a SVM classifier based on optical flow between consecutive frames. The optical flow was computed by down-sampling the image to $p \times p$ pixels, where p denotes the bin size. We tested different bin sizes and only the best result is presented in the chapter (using $p = 20$).

We obtained an average accuracy of 45%, 61% and **85**% using color based approach, optical flow based approach and our method respectively using independent per frame classification. The accuracy improved to 56%, 65.4% and **92.2**% respectively, after the MRF smoothing. The confusion matrices of results obtained after MRF smoothing are shown in Table 4.1, Table 4.2 and Table 4.3.

The dominant color ratio based approach (Table 4.1) fails to do the classification accurately. Strong misclassification between Top-zoom out and Top zoom-in shots can be observed. This occurs due to the large ratio of green color in both top zoom in (usually close up shots from top angle) and top zoom out shots. The color based approach completely fails to classify Miscellaneous (crowd) shots and also heavily confuses among ground zoom-out and ground zoom-in shots. The optical flow based approach also gives noisy results and strongly confuses top zoom out shots with other viewpoints.

	Corner	Foul	Goal	Gameplay	Subst.
Corner	**43.4**	4.9	31.3	9.8	10.6
Foul	0	**48.5**	32.9	13.7	4.9
Goal	2.5	8.9	**82**	4.5	2.1
Gameplay	3.9	4.2	5.8	**85.2**	0.9
Subst.	1.6	25.5	41.1	0.7	**31.1**

Table 4.4 Event tagging results using only the BoW histograms considering five different events (percentages).

	Corner	Foul	Goal	Gameplay	Subst.
Corner	**71.2**	0	28.8	0	0
Foul	0.5	**43.8**	40.2	8	7.5
Goal	0.6	8.7	**81.2**	4.2	5.3
Gameplay	3.4	3	2.4	**91.2**	0
Subst.	0	12.7	32.7	0	**54.6**

Table 4.5 Event tagging results using the combination of camera label information with the BoW histograms.

	Corner	Foul	Goal	Gameplay	Subst.
Corner	**75.1**	5.4	18	0.6	0.9
Foul	0.2	**57.4**	22	10.4	10
Goal	2.7	3.4	**84.6**	4.2	5.1
Gameplay	0.9	1.5	1.5	**94.6**	1.5
Subst.	1.3	9.6	24	2.7	**62.4**

Table 4.6 Event tagging results using the combination of BoW histograms; camera label information and the motion features.

On the other hand, our method provides highly accurate results(Table 4.3). It almost perfectly classifies top zoom out shots which typically cover major part of the football game. The confusion of other classes with top zoom out shots is also negligible (column one in Table 4.3). In general we give high accuracy for most classes (near 80%) and the confusion when occurs is understandable. For example, in tight close up shots, it is difficult to distinguish between top zoom-in and ground zoom-in camera angles (the background is blurred in both cases and only the player profile contributes to the features).

4.3.2 Event tagging

For event tagging experiment, we created a dataset of 176 clips encompassing all the five events. The dataset includes 43 goal sequences, 30 corner sequences, 42 substitution sequences, 27 fouls sequences and 34 gameplay sequences. The clips were extracted from the video by manually annotating the start and the end of the event. The length of each sequence is different and is defined based on the knowledge of the game. For example, a goal event starts from the point the ball enters the goal post and ends at the new kickoff (so the goal event includes all the celebrations). Similarly, a corner event starts just before the corner kick and ends at the first deflection. The sequences for gameplay event were of random length. We used 60% of the data for training/validation and 40% for testing. Example videos of each kind of event with qualitative results are provided in the supplementary material. Another qualitative result on an interesting 30 minutes sequence (with five goals in quick succession) is shown in Figure 4.1.

For quantitative analysis, we sampled test data clips from different classes and joined them in random order to create a large video sequence. The event classification task was then performed per frame basis on this large sequence. We tested classification task using both the mean (taking mean of features from 40 frames) and concatenation (concatenating features from 40 frames). The average accuracy of around 80% was obtained in both the cases, which improved to 85.5% after MRF smoothing.

The confusion matrix of results after MRF smoothing (using mean features from 40 frames) is given in Table 4.6. The goal event and gameplay event are predicted with an accuracy over 85%. The confusion when occurs, is understandable, for example some corner events lead into goals or near misses and in such instances the corner event is misclassified as goal event. Similarly substitution event is often followed with the crowd viewpoint (cheering the player) and replay of goals (or assists) by substituted player, which leads to confusion between substitution and goal event.

Comparing results in Table 4.4, Table 4.5 and Table 4.6 we can observe that both context(camera label information) and motion features bring significant improvements in event recognition task. For example, the accuracy of predicting corner improves by almost 30% after including camera label information (Table 4.4 and Table 4.5). Similarly, including motion information brings almost 15% improvement in predicting Foul event (Table 4.5 and Table 4.6). Overall, the average accuracies increased from 77% to 81.8% after adding camera context. The average accuracy further improved to 85.5% after the inclusion of motion features. In terms of time complexity, the initial method using only BoW histograms runs at around 1 fps in our current implementation on a single core intel-i5 CPU with 16GB memory. Adding camera label information does not bring any significant change, but including motions cues increases computational time to around 0.4 fps.

We further compare our method with a hidden conditional random field (HCRF) based method [57] and a recent action recognition using trajectory-pooled-deep-convolutional descriptors [72] (with combined spatial and temporal features). For comparison with [57] we use our own implementation for computing features with sequence classification toolbox from [69]. The results with precision for predicting each event is presented in Table 4.7. We can clearly observe that our approach outperforms the generic state of the art action recognition method [72], which shows the importance of the context. Our method also improves the precision over a more sophisticated(using ad-hoc domain specific features) sequence classification method [72] on most of the events.

	Corner	Foul	Goal	Gameplay	Subst.
TDD	21.05	12.82	77.45	77.08	25.80
HCRF	7.6	15.5	6.6	77.13	**86.44**
Ours	**75.1**	**57.4**	**84.6**	**94.6**	62.4

Table 4.7 Comparison of our method with TDD [72] and HCRF [57].

4.3.3 Spatial segmentation

In this section, we demonstrate the usefulness of camera label information with an application of spatial segmentation. Given an individual frame, the task is to assign each pixel with an unique class label. We consider three different classes i.e players; crowd and playing field. The experiments are performed on a dataset of 50 images with top-zoom out camera labels and 50 images with zoom-in camera labels (sampled both from ground zoom-in and top zoom-in cameras). The images are equally

sampled from 10 different matches to cover different challenges in segmentation like shadows, different color player jerseys etc. The ground truth labels for all the 100 images are created manually.

We investigate two different segmentation approaches. First we follow the class based image segmentation approach of Ladicky et al. [38] which casts the segmentation problem as graph cut based inference on conditional random fields. Second we propose a segmentation approach specific to top zoom-out views. Our main insight is that knowing the context can help to design minimal segmentation algorithms bringing significant improvement in terms of both time and labelling accuracy. Knowing that the camera label is a top zoom-out, we can assume that the large part of the frame will be covered by playing field which can be efficiently segmented based on color. We use a variation of Heckbert's [26] median cut algorithm to estimate dominant color and segment out the playing field in normalized rgb space. The segmentation is performed using a threshold on euclidean distance from the dominant color. The holes in the playing field are then labelled as players and the rest is labelled as crowd.

We then trained three instances of automatic labelling environment (ALE) [38], first for the top zoom-out camera labels, second for the zoom-in camera labels and third for the combined set. Half of the total images were used for training in each case. The ALE segmentation results for top zoom-out and zoom-in are illustrated in Table 4.8. We can observe that ALE fails to segments players accurately in images with top zoom-out camera labels. Interestingly when we trained ALE by taking frames with from both zoom-out and zoom-in viewpoints we obtained nearly same results. This suggests that ALE is not taking full advantage of the context. We then performed segmentation using the second context

	ALE top zoom-out	ALE zoom-in
Players	44.4	80.8
Field	99.31	93.3
Crowd	99.8	91.5

Table 4.8 Results of average per class recall measure, defined as $\frac{TruePositives}{TruePositives+FalsePositives}$ on ALE [38]. The recall measure for class *Players* is low in top zoom-out viewpoints.

aware approach (based on dominant color) and the recall for the class *Players* improved to $\mathbf{64.2\%}$ maintaining nearly the same recall for *Field* and *Crowd* classes. The results from the dominant color based approach were obtained in real time compared to 30 hours training (for 50 images with image resolution of $720p$) and 3 hours testing (4-5 minutes per frame) in case of ALE. This clearly shows that knowing context (like camera labels) can bring significant improvements for the task of spatial segmentation both in terms of performance and recall.

4.4 Summary

In this chapter we have investigated the problem of automatic analysis of football broadcast videos. We have shown that this problem can be partitioned into two smaller problems, namely, camera viewpoint estimation and event tagging. We have demonstrated that since the input videos are already edited,

the camera viewpoint information provides a natural context which could be exploited to improve the other task of event tagging.

Based on thorough quantitative analysis on variety of tasks in 10 football matches, we have justified our claims. Our method obtains an overall accuracy of 92.2% for camera viewpoint estimation and 85.5% accuracy for event classification. We have also demonstrated that the contextual approach can outperform state of the art deep learning based action recognition approaches. We further demonstrate that the accuracy of tasks like spatial semantic segmentation can be improved by as much as 20% using the context.

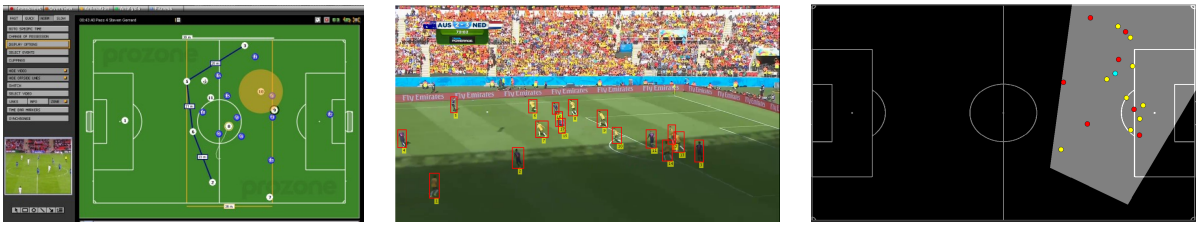
Chapter 5

Automated Top View Registration of Broadcast Football Videos

Advent of tracking systems by companies like Prozone [2] and Tracab [3] has revolutionized the area of football analytics. Such systems stitch the feed from six to ten elevated cameras to record the entire football field, which is then manually labelled with player positions and identity to obtain the top view data over a static model as shown in Figure 5. Majority of recent research efforts [23, 46, 45, 10] and commercial systems for football analytics have been based on such top view data (according to prozone website, more than 350 professional clubs now use their system). There are three major issues with such commercial tracking systems and associated data. First, it is highly labour and time intensive to collect such a data. Second, it is not freely available and has a large price associated with it. Third, such a data can not be obtained for analysing matches where the customized camera installations were not used. It is also difficult for most research groups to collect their own data due to the challenges of installing and maintaining such systems and the need of specific collaborations with the clubs/stadiums.

All the above problems can be addressed, if we can obtain such data using the readily available broadcast videos. However, this is a non trivial task since the available broadcast videos are already edited and only show the match from a particular viewpoint/angle at a given time. Hence, obtaining the top view data first requires the registration of the given viewpoint with the static model of the playing surface. This registration problem is challenging because of the movement of players and the camera; zoom variations; textureless field; symmetries and highly similar regions. Due to these reasons, this problem has interested several computer vision researchers over the past [52, 28, 44], however most of the existing solutions are based on computation of point correspondences or/and require some form of manual initialization. Not just that the manual initialization for each video sequence is a impractical task (as shot changes occur quite frequently), such approaches are also not applicable in the presented scenario due to absence of good point correspondences (the football playing surface is almost textureless in contrast to the cases like American football [28]).

Motivated by the above reasons, we take an alternate approach based on edge based features and formulate the problem as nearest neighbour search to the closest edge map in a precomputed dictionary with known projective transforms. Since, manual labelling of a sufficiently large dictionary of edge maps with known correspondences is an extremely difficult and tedious task, we employ a semi su-



(a) (b)

Figure 5.1 (a) A snapshot from Prozone tracking system. (b) An example result from the proposed method, which takes as input a broadcast image and outputs its registration over the static top view model with the corresponding player positions. The yellow, red and cyan circles denote the players from different teams and referee respectively.

pervised approach, where a large 'camera-view edge maps to projective transform pairs' are simulated from a small set of manually annotated examples (the process is illustrated in Figure 5.3). The simulated dictionary generation allows us to cover edge maps corresponding to 'every inch and degree' movement of the camera from different viewpoints (which is an infeasible task manually). More importantly, this idea reduces the accurate homography estimation problem to a minimal dictionary search using the edge based features computed over the query image. The tracking data can then be simply obtained by projecting the player detections performed over broadcast video frames, using the same projective transform. An example of our approach over a frame from Australia vs Netherlands world cup match is illustrated in Figure 5.

Since the camera follows most of the relevant events happening in the game, it can be fairly assumed that the partial tracking data (only considering the players visible in the current camera view) obtained using the proposed approach is applicable to most of the work on football play style analytics [23]. Furthermore, the knowledge of camera position and movement can work as an additional cue for applications like summarization and event detection (goals. corners etc.), as the camera movement and editing is highly correlated with the events happening in the game. It is also useful for content retrieval applications, for instance it can allow queries like "give me all the counter attack shots" or "give me all the events occurring on the top left corner" etc. The proposed approach can also be beneficial in several other interesting research topics like motion fields for predicting the evolution of the game [33], social saliency for optimized camera selection [65] or automated commentary generation [5].

More formally our work makes following contributions:

1. We propose a novel framework to obtain the registration of soccer broadcast videos with a static model. We demonstrate that the proposed nearest neighbour search based approach makes it possible to robustly compute the homography in challenging cases, where even manually labelling the minimum four point based correspondences is difficult.

2. We thoroughly compare three different approaches based on HOG features, chamfer matching and convolution neural net (CNN) based features to exploit the suitable edge information from the playing field.

3. We propose a semi-supervised approach to synthetically generate a dictionary of 'camera-view to projective transform pairs' and present a novel dataset with over a hundred and sixty thousand pairs.

4. We propose a mechanism to further enhance the results by combining the results of different methods using a Markov Random Field (MRF) optimization and removing camera jitter using a convex optimization framework.

5. We present extensive qualitative and quantitative results on a simulated and a real test dataset, to demonstrate the effectiveness of the proposed approach.

The proceeding section briefly explains the related work. The semi-supervised dictionary learning approach is described in Section 5.2.1, followed by the explanation of the proposed matching algorithms. Section 5.2.3 covers the optimization techniques followed by the experimental results and concluding discussion.

5.1 Related work

Top view data for sports analytics has been extensively used in previous works. Bialkowski et al. [9] uses 8 fixed high-definition (HD) cameras to detect the players in field hockey matches. They demonstrated that event recognition (goal, penalty corner etc.) can be performed robustly even with noisy player tracks. Lucey et al. [45] used the same setup to highlight that a role based assignment of players can eliminate the need of actual player identities in several applications. In basketball, a fixed set of six small cameras are now used for player tracking as a standard in all NBA matches, and the data has been used for extensive analytics [20]. Football certainly has gained the most attention [23] and the commercially available data has been utilized for variety of applications from estimating the likelihood of a shot to be a goal [46] or to learn a team's defensive weaknesses and strengths [10].

The idea of obtaining top view data from broadcast videos has also been explored in previous works, Okuma et al. [52] used KLT [61] tracks on manually annotated interest points (with known correspondences) and used them in RANSAC [19] based approach to obtain the homographies in presence of camera pan/tilt/zoom in NHL hockey games. Gupta et al. [24] showed improvement over this work by using SIFT features [42] augmented with line and ellipse information. Similar idea of manually annotating initial frame and then propagating the matches has also been explored in [44]. Li and Chellapa [41] projected player tracking data from small broadcast clips of American football in top view form to segment group motion patterns. The homographies in their work were also obtained using manually annotated landmarks.

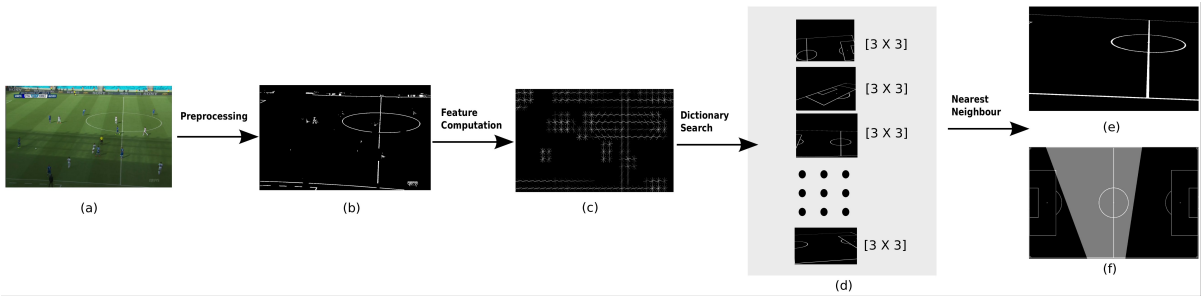


Figure 5.2 Overview of the proposed approach. The input to the system is a broadcast image (a) and the output is the registration over the static model (f). The image (e) shows the corresponding nearest neighbour edge map from the synthetic dictionary.

Hess and Fern [28] build upon [52] to eliminate the need of manual initialization of correspondences and proposed an automated method based on SIFT correspondences. Although their approach proposes an improved matching procedure, it may not apply in case of normal football games due to lack of ground visual features. Due to this reason, instead of relying on interest point matches, we move toward a more robust edge based approach. Moreover, we use stroke width transforms(SWT) [16] instead of usual edge detectors for filtering out the desired edges. Another drawback of the work in [28] is that the static reference image in their case is manually created, and the process needs to be repeated for each match again. On the other hand, our method is applicable in more generic scenario and we have tested it on data from 16 different matches. More recently, the work by Agarwal et al. [4] posed the camera transformation prediction between pair of images as a classification problem by binning possible camera movements, assuming that there is a reasonable overlap between the two input images. However, such an approach is not feasible for predicting exact projective transforms.

Our work is also related to camera stabilization method of Grudmann et al. [22] which demonstrate that the stabilized camera motion can be represented as combination of distinct constant, linear and parabolic segments. We extend their idea for smoothing the computed homographies over a video. We also benefit from the work of Muja and Lowe [49] for computationally efficient nearest neighbour search.

5.2 Method

The aim of our method is to register a video sequence with a predefined top view static model. The overall framework of our approach is illustrated in Figure 5.2. The input image is first pre-processed to remove undesired areas such as crowd and extract visible field lines and obtain a binary edge map. The computed features over this edge map are then used for knn search in pre-built dictionary of images with synthetic edge maps and corresponding homographies. Two different stages of smoothing are then performed to improve the video results. We now describe, each of these steps with detail:

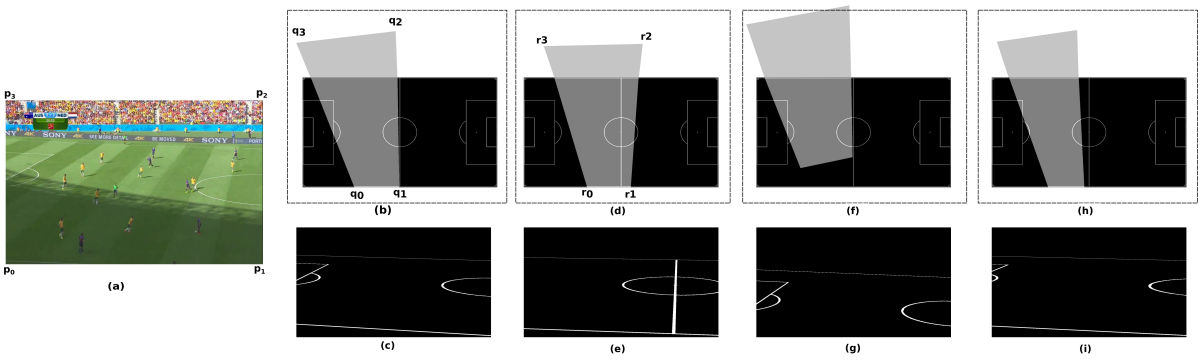


Figure 5.3 Illustration of synthetic dictionary generation. First column shows the input image and second column shows the corresponding registration obtained using manual annotations of point correspondences. The pan, tilt and zoom simulation process is illustrated in third, fourth and fifth column respectively.

5.2.1 Semi supervised dictionary generation

Two images of the same planar surface in space are related by a homography ($\mathbf{H}$). In our case, this relates a given arbitrary image from the soccer broadcast to the static model of the playing surface. Given a point $x = (u, v, 1)$ in one image and the corresponding point $x' = (u', v', 1)$, the homography is a 3×3 matrix, which relates these pixel coordinates $x' = \mathbf{H}x$. The homography matrix has eight degrees of freedom and can ideally be estimated using 4 pair of perfect correspondences (giving eight equations). In practice, it is estimated using a RANSAC based approach on a large number of partially noisy point correspondences.

However, finding a sufficient set of suitable non-collinear candidate point correspondences is difficult in the case of football fields. And manual labelling each frame is not just tedious it is also challenging task in several images (for example consider the query image in Figure 5.2). Due to these reasons, we take an alternate approach: we first hand label the four correspondences in small set of images and then use them to simulate a large dictionary of 'field line images (synthetic edge maps) and related homography pairs'. An example of the process is illustrated in Figure 5.3. Given a training image (Figure 5.3(a)), we manually label four points to compute homography ($\mathbf{H_1}$) and register it with the static top view of the ground (Figure 5.3(b)). We can observe that after applying homography to entire image and warping, the boundary coordinates (p_0, p_1, p_2, p_3) gets projected to points (q_0, q_1, q_2, q_3) respectively. We can now use this to obtain the simulated field edge map (Figure 5.3(c)) by applying $(\mathbf{H_1^{-1}})$ on the static model (top view). This simulated edge map paired with $\mathbf{H_1}$ forms an entry in the dictionary.

We simulate pan by rotating the quadrilateral (q_0, q_1, q_2, q_3) around the point of convergence of lines q_0q_3 and q_1q_2 to obtain the modified quadrilateral (r_0, r_1, r_2, r_3), as illustrated in Figure 5.3(d). Using (r_0, r_1, r_2, r_3) and (p_0, p_1, p_2, p_3) as respective point correspondences, we can compute the inverse transform $(\mathbf{H_2^{-1}})$ to obtain Figure 5.3(e). This simulated image along with $\mathbf{H_2}$ forms another entry

in the dictionary. Similarly, we simulate tilt by moving the points q_0q_3 and q_1q_2 along their direction and we simulate zoom by reducing the angle between the lines q_0q_3 and q_1q_2. Now, by using different permutations of pan, tilt and zoom over a set of 80 manually annotated images, we learn a large dictionary $D = \{I_j, H_j\}$ where I_j is the simulated edge map, H_j is corresponding homography and $j \in [0 : N-1]$. The manually annotated images and the permutations were chosen carefully to comprehensively cover the different field of views.

5.2.2 Nearest neighbour search algorithms

We pose the homography estimation problem as the nearest neighbour search over the synthetic edge map dictionary. Given a preprocessed input image and its edge map x, we find the best matching edge map I_j (or k best matching edge maps) from the dictionary and output the corresponding homography H_j (or set of k homographies). In this section, we present three different methods for computing the nearest neighbours studied in this work. We specifically choose an image gradient based approach (HOG), a direct contour matching approach (chamfer matching) and an approach learning abstract mid level features (CNN's).

5.2.2.1 Chamfer matching based approach

The first method we propose is based on chamfer matching [8], which is a popular technique to find the best alignment between two edge maps. Although proposed decades ago, it remains a preferred method for several reasons like speed and accuracy, as discussed in [67]. Given two edge maps x and I_j, the chamfer distance quantifies the matching between them. The chamfer distance is the mean of the distances between each edge pixel in x and its closest edge pixel in I_j. It can be efficiently computed using the distance transform function $T(.)$, which takes a binary edge image as input and assigns to each pixel in the image the distance to its nearest edge pixel. The chamfer matching then reduces to a simple multiplication of the distance transform on one image with the other binary edge image. The process is illustrated in Figure 5.4.

We use the chamfer distance for the nearest neighbour search. Given an input image x and its distance transform $T(x)$ we search for index j^* in the dictionary, such that

$$j^* = \underset{j}{\operatorname{argmin}} \frac{T(x).I_j}{\|I_j\|_1} \,, \tag{5.1}$$

where $\|\|_1$ is the ℓ_1 norm and the index j^* gives the index of the true nearest neighbour. Given an epsilon $\epsilon > 0$, the approximate nearest neighbours are given by list of indices j, such that $\frac{T(x).I_j}{\|I_j\|_1} \leq (1+\epsilon)\frac{T(x).I_{j^*}}{\|I_{j^*}\|_1}$.

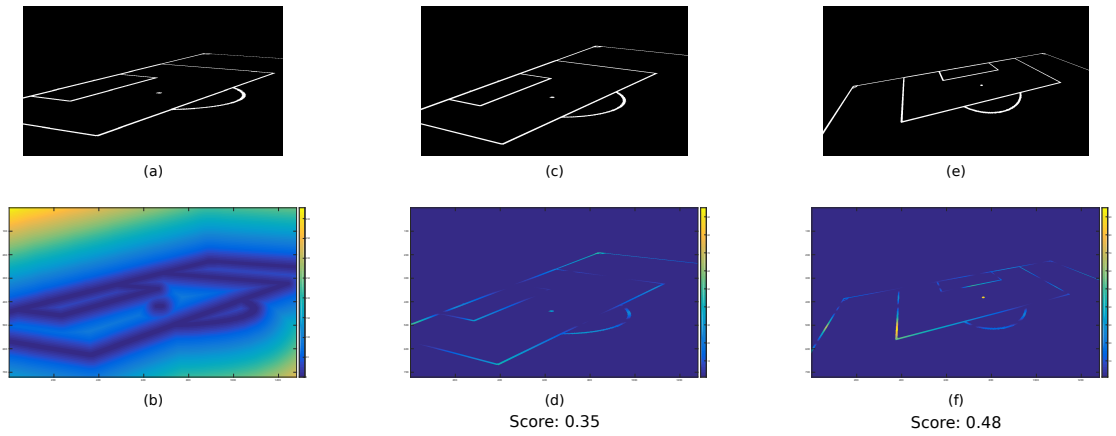


Figure 5.4 Illustration of chamfer matching. The first column shows the input image x and its distance transform $T(x)$. The second and third column show two different edge maps and their multiplication with $T(x)$. We can observe that image (c) is a closer match and gives a lower chamfer distance.

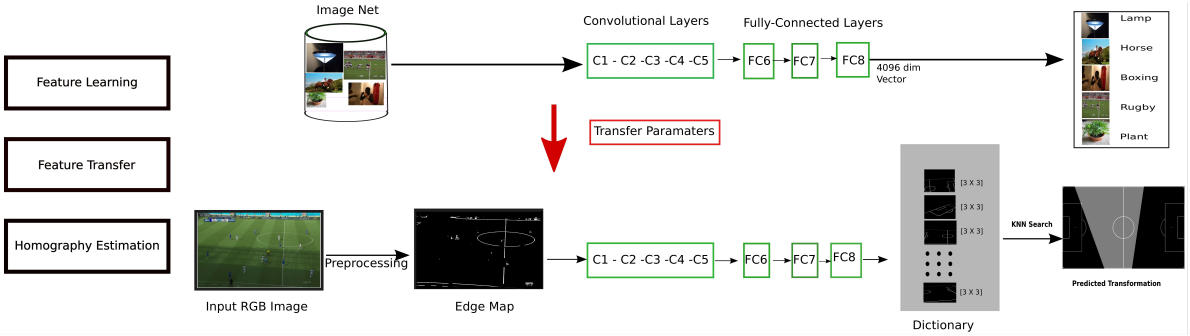


Figure 5.5 Illustration of CNN pipeline. The parameters learnt over the ImageNet classification task are transferred for the task of nearest neighbour search.

5.2.2.2 HOG based approach

The second method is based on HOG features [13], where the nearest neighbour search is performed using the euclidean distance on the HOG features computed over both the dictionary edge maps and the input edge map. So, given the input edge map x and its corresponding HOG features $\phi_h(x)$ we search for j^* in the dictionary, such that

$$j^* = \underset{j}{\operatorname{argmin}} \|\phi_h(x) - \phi_h(I_j)\|_2 \,, \tag{5.2}$$

where $\|\|_2$ is the ℓ_2 norm.

5.2.2.3 CNN based approach

The convolution neural networks (CNN's) have shown significant improvement in variety of computer vision tasks, and this success can be attributed to their ability to learn powerful mid level features

instead of using hand-crafted features. It has been shown that CNN features learnt for one task like object classification, can be efficiently used for other tasks like object localization [54]. On the similar lines, we use the mid level features learnt using the network architecture of Krizhevesky et al. [37] on ImageNet [58] for the task of nearest neighbour search. The network in [37] is composed of five successive convolution layers $C_1, C_2, ..., C_5$, followed by three fully connected layers FC_6, FC_7, FC_8 (illustrated in Figure 5.5). We remove the last fully connected layer FC_8 (of dimensionality 4096) and use it as the feature vector for the nearest neighbour search.

So, given the input edge map x and its output at last fully connected layer $\phi_c(x)$ we search for j^* in the dictionary, such that

$$j^* = \underset{j}{\operatorname{argmin}} \|\phi_c(x) - \phi_c(I_j)\|_2 \,, \tag{5.3}$$

where $\|\|_2$ is the ℓ_2 norm.

5.2.3 Smoothing and Stabilization

For a given input video sequence, we compute k homography candidates independently for each frame using the nearest neighbour search algorithms described above. Just taking the true nearest neighbour for each frame independently may not always give the best results due to noise in the pre-processing stage or the absence of a close match in the simulated dictionary. To remove outliers and to obtain a jerk free and stabilized camera projections, we use two different optimization stages. The first stage uses a markov random field (MRF) based optimization, which selects one of the k predicted homographies for each frame to remove the outliers and jerks. The second stage further optimizes these discrete choices, to obtain a more smooth and stabilized camera motion.

5.2.3.1 MRF optimization

The algorithm takes as input the k predicted homographies for each frame with their corresponding nearest neighbour distances and outputs a sequence of $\xi = \{s_t\}$ states $s_t \in [1 : k]$, for all frames $t = [1 : N]$. It minimizes the following global cost function:

$$E(\xi) = \sum_{t=1}^{N} E_d(s_t) + \sum_{t=2}^{N} E_s(s_{t-1}, s_t). \tag{5.4}$$

The cost function consists of a data term E_d that measures the evidence of the object state using the nearest neighbour distances and a smoothness term E_s which penalizes sudden changes. The data term and the smoothness term are defined as follows:

$$E_d(s_t) = -\log(P(s_t, t)) \,. \tag{5.5}$$

Here, $P(s_t, t)$ is the nearest neighbour distance for state s_t at frame t. And

$$E_s(s_{t-1}, s_t) = \|H_{s_t} - H_{s_{t-1}}\|_2, \tag{5.6}$$

is the Euclidean distance between the two (3×3) homography matrices, normalized so that each of the eight parameters lie in a similar range. Finally, we use dynamic programming (DP) to solve the optimization problem presented in Equation 5.4.

5.2.3.2 Camera stabilization

The MRF optimization removes the outliers and the large jerks, however a small camera jitter still remains because its output is a discrete selection at each frame. We solve this problem using a solution inspired by the previous work on camera stabilization [22]. The idea is to break the camera trajectory into distinct constant (no camera movement), linear (camera moves with constant velocity) and parabolic (camera moves with constant acceleration or deceleration) segments. We found that this idea also correlates with the camera work by professional cinematographers, who tend to keep the camera constant as much as possible, and when the movement is motivated they constantly accelerate, follow the subject (constant velocity) and then decelerate to static state [68]. The work in [22] shows that this can be formalized as a L1-norm optimization problem.

However the idea of [22] cannot be directly applied in our case, as we can not rely on interest point features for the optimization, because we are already in projected top view space. We parametrize the projected polygon (for example the quadrilateral $q_0q_1q_2q_3$ in Figure 5.3) using six parameters, the center of the camera (cx, cy), the pan angle θ, the zoom angle ϕ and two intercepts $(r1, r2)$ (for near clipping plane and far clipping plane respectively). Given a video of N frames, we formulate the stabilization as convex optimization over the projected plane $P_t = \{cx_t, cy_t, \theta_t, \phi_t, r1_t, r2_t\}$ at each frame $t \in [0 : N-1]$. We solve for P_t^* which minimizes the following energy function:

$$E_c = \sum_{t=1}^{N}(P_t^* - P_t)^2 + \lambda_1 \sum_{t=1}^{N-1} \|P_{t+1}^* - P_t^*\|_1 1 + \lambda_2 \sum_{t=1}^{N-2} \|P_{t+2}^* - 2P_{t+1}^* + P_t^*\|_1 + \lambda_3 \sum_{t=1}^{N-3} \|P_{t+3}^* - 3P_{t+2}^* + 3P_{t+1}^* - P_t^*\|_1. \tag{5.7}$$

The energy function E_c comprises of a data term and three L1-norm terms over the first order, second order and the third order derivatives and λ_1, λ_2 and λ_3 are parameters. As E_c is convex, it can be efficiently solved using any off the shelf solver, we use cvx [21].

5.3 Experimental Results

We perform experiments on broadcast video sequences sampled from 16 matches of football world cup 2014. We evaluate our work using three different experiments. The first experiment compares the three matching approaches (chamfer, HOG and CNN based) over a large simulated test dataset. The second experiment draws similar comparison over actual broadcast images from different matches with different teams in varying conditions. The third experiment showcases the results over broadcast video sequences, highlighting the benefits of the camera smoothing and stabilization.

Synthetic Dataset	
NN-Chamfer	89.6
NN-HOG	89.7
NN-CNN	88.4

Broadcast image dataset	
NN-Chamfer	78.5
NN-HOG	81.6
NN-CNN	66.1

Table 5.1 Results over the synthetically generated test dataset (left) and results over the real image dataset (right).

5.3.1 Results over simulated edge maps

Similar to the procedure explained in section 5.2.1, we generate a set of 4500 edge map and homography pairs and use it as a test dataset. We annotated a different set of images for generating this test dataset to keep it distinct with the training set (used for learning the dictionary). Then, we compute the true nearest neighbour using the three approaches explained in section 5.2.2 on each of the test image (edge map) independently. We use the computed homographies to project the given test image over the static model and obtain a polygon P_e. Since, the simulated dataset also contains the corresponding ground truth homography matrix, we then use it to obtain actual ground truth top view estimation, which gives another polygon P_g. To evaluate the accuracy, we use the intersection-over-union measure over the ground truth and the estimated polygons i.e. $\frac{P_e \cap P_g}{P_e \cup P_g}$.

The results are illustrated in Table 5.3.1. Interestingly, there is not much difference in results using different features and all three approaches give nearly ninety percent accuracy. Since, the intersection-over-union measure decreases quite rapidly, a ninety percent accuracy shows that the idea works nearly perfect in absence of noise.

5.3.2 Results over broadcast images

The proposed method can only be practicable applicable if it can broadly replicate the accuracy obtained on synthetic dataset over sampled RGB images from broadcast videos. Since, the nearest neighbour search takes as input the features over edge maps, we need to first pre-process the RGB images to obtain the edge maps (only containing the field lines). Moreover, a football broadcast consists

Figure 5.6 We classify the camera viewpoints from a usual football broadcast into five different categories namely (from left to right) top zoom-out, top zoom-in, ground zoom-out, ground zoom-in and miscellaneous (covering mainly the crowd view).

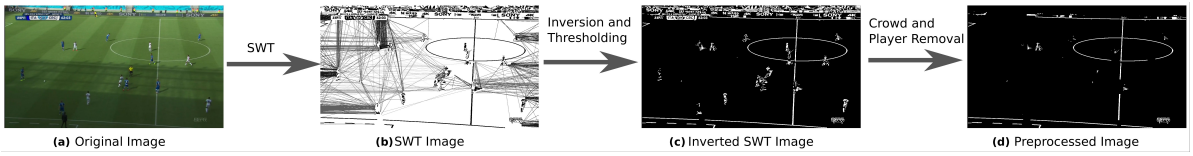


Figure 5.7 Illustration of the pre-processing pipeline. Observe that how SWT is able to filter out the field lines in presence of complex shadows (usual edge detectors will fail in such scenarios).

of different kind of camera viewpoints (illustrated in Figure 5.6). The field lines are only properly visible in the far top zoom-out view (which though covers nearly seventy five percent of the broadcast video frames). Henceforth, we propose a two stage pre-processing algorithm:

5.3.2.1 Pre-processing

The first pre-processing step selects the top zoom-out frames from a given video sequence. We employ the classical Bag of Words (BoW) representation on SIFT features to classify each frame into one of the five classes illustrated in Figure 5.6. We use a linear SVM to perform per frame classification (taking features from a temporal window of 40 frames centred around it), followed by a temporal smoothing. Even using this simple approach, we achieve an accuracy of 98 percent, for the top zoom-out class label (trained over 45 minutes of video and tested over 45 minutes of video from another match).

Now, given the top zoom-out images from the video, the second pre-processing step extracts the edge map with field lines. The entire procedure is illustrated in Figure 5.7. First we compute the stroke width transform (SWT) over the input images and filter out the strokes of size more than 10 pixels (preserving the field lines which comprise of small consistent stroke widths). The benefit of using SWT over usual methods like canny edge detection is that it is more robust to noise like shadows, field stripes (light-dark stripes of green colors) etc. We further remove crowd (using color based segmentation of field) and players (using a HOG based player detector) to obtain the edge map, primarily containing only the field lines with partial noise (Figure 5.7(d)).

5.3.2.2 Quantitative evaluation

We selected 200 RGB images from the set of top zoom-out images predicted by the pre-processing algorithm and manually labelled four point correspondences to register it with the static model for quantitative evaluation. The images were selected from different video segments across multiple matches of football World cup 2014. The images were chosen from varying lighting conditions with prominent shadows, motion blur, varying angles and zooms covering different areas of the playing field to properly test the robustness of the proposed approach. We then evaluate the three proposed nearest neighbour search algorithms over these images and compared them with the corresponding ground truth projections (computed using manually labelled point correspondences). The intersection-over-union measure on the estimated and ground truth projections are given in Table 5.3.1.

We can observe that the HOG features give the best results over the three approaches with an accuracy of around 82 percent. The result only degrade by around eight percent from the noiseless scenario of synthetic dataset and performs extremely well under diverse conditions (as illustrated in Figure 5.8). The chamfer matching based approach also gives reasonable results but we found it to be slightly more sensitive towards noise. Surprisingly, the accuracy of the CNN based approach degraded considerably over the synthetic experiments. This shows that it is not able to adjust to even the partial noise present while testing on actual broadcast images which was not observed during the training over synthetic dataset.

5.3.2.3 Qualitative evaluation

Results over a small set of images using HOG based approach are shown in Figure 5.8. We can observe that the predictions are quite accurate over diverse scenarios and the method works perfectly even in cases where manual annotation of point correspondences is challenging itself (Figure 5.8(d)). As we see in the images (g) and (h), our algorithm is able to handle extreme variations in camera angles, demonstrating the wide suitability of our approach. We can also observe the robustness over the unconstrained challenges like shadows $\{$(a),(b),(e),(f)$\}$ and motions blur $\{$(d)$\}$ showing the practical applicability of our method. The applicability over varying zoom, coverage and angles is also evident. The reader can refer to the supplementary material for more details, where we provide the results over the entire dataset.

5.3.3 Results over broadcast videos

To demonstrate the effectiveness of the temporal smoothing we choose 30 broadcast video sequences sampled across 16 world cup matches containing close to hundred top zoom-out shots. The total length of video sequences covers around one hour of broadcast videos. We compute the results on each frame individually over all these videos and then perform MRF smoothing. For a quick evaluation, we then check every 50th frame over all the 100 shots and mark a prediction correct if it qualitatively matches the respective field of view in the broadcast image near perfectly. And we found out that the prediction of the proposed method were correct on 90.1% of the frames.

5.3.3.0.1 MRF evaluation: For a more detailed quantitative analysis of MRF optimization, we manually annotated two of the shots of total length 3600 frames, at equal interval of approximately 10 frames. Each frame in this case, was annotated with four point correspondences. We then compute the homography using nearest neighbour search for each frame independently and compare it with the MRF based approach (which selects among the five nearest neighbours at each frame). We found that the results (intersection-over-union measure) improved from 84% to 88% by employing the MRF based optimization over the per frame results.

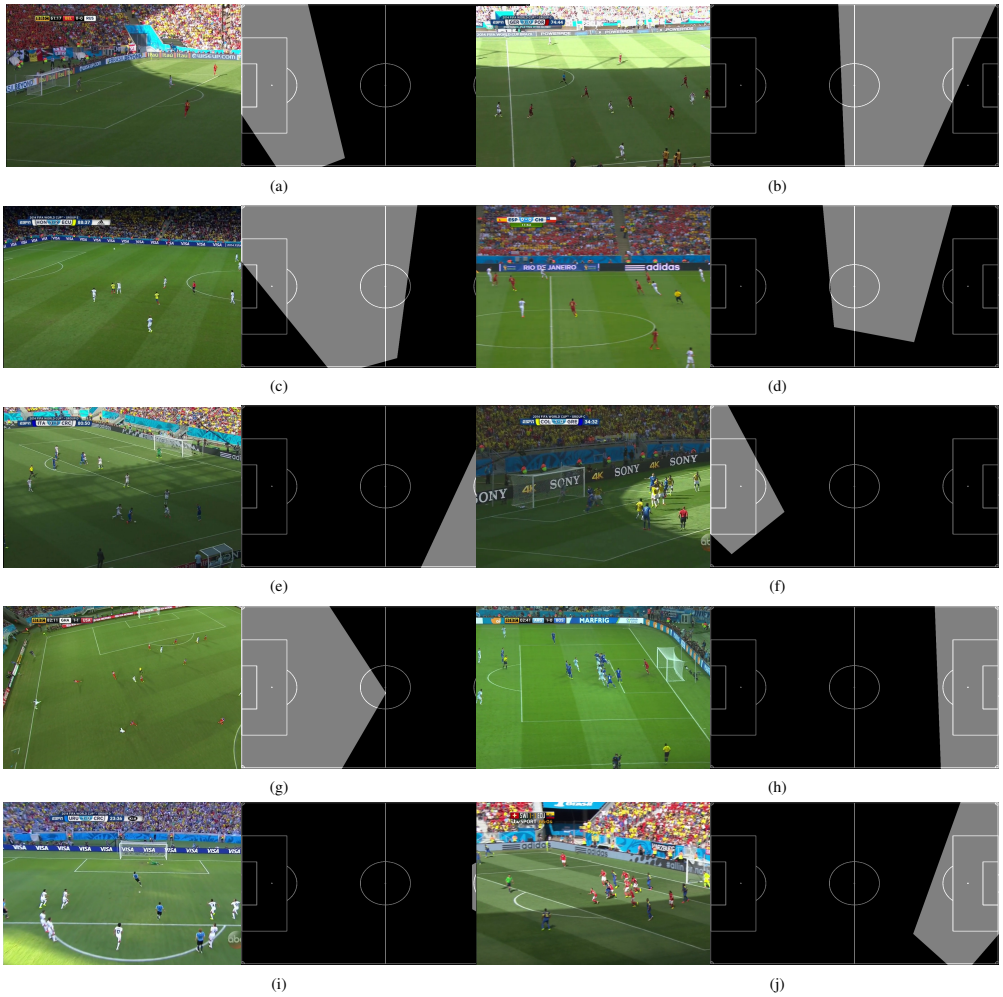


Figure 5.8 Original images and registered static model pairs computed using the HOG based approach. Covering shadows {(a),(b), (e),(f)}, motion blur {(d)}, varying zoom {(a),(c)}, varying camera viewpoints {(g),(h)}, varying positions {(e),(f)} etc.

5.3.3.0.2 Convex optimization evaluation: Qualitative results of the camera stabilization are shown in Figure 5.9 over a video sequence from Chile vs Australia world cup match. The video starts at midfield, pans to left goal post, stays static for few frames and quickly pans back to midfield following a goalkeeper kick. The figure shows the pan angle trajectory of the per frame predictions with and without camera stabilization. We can observe that the optimization clearly removes jitter and replicates a professional cameraman behaviour. The actual and the stabilized video are provided in the supplementary material.

5.3.3.0.3 Player tracking application: We perform player detections using HOG based player detector (trained on broadcast video data), project them over the static model using predicted homogra-

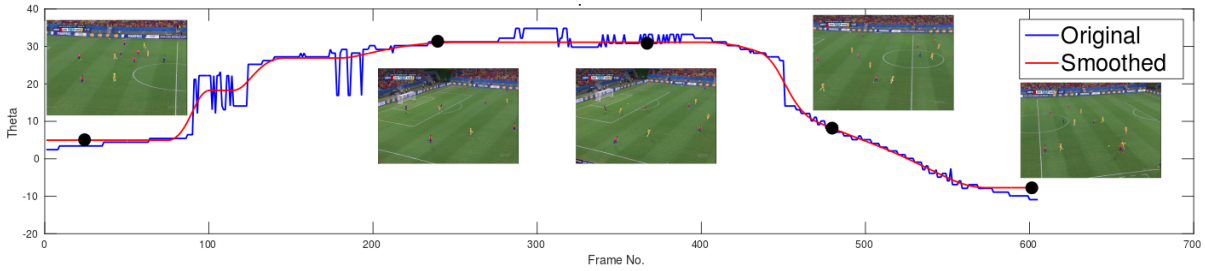


Figure 5.9 Illustration of stabilization using convex optimization. The blue curve shows the pan angle predicted by the proposed approach on each frame individually. The red curve shows the stabilized pan angle after the convex optimization. We can observe the the smoothed pan angle composes of distinct static, linear and quadratic segments. The black dots denote the frames at respective locations.

phies and then perform tracking in the projected space using a Hungarian algorithm [29]. Example video results are provided in supplementary material.

5.4 Summary

We have presented a method to compute projective transforms as a nearest neighbour search and have shown that the presented approach gives highly accurate results (over 85% after MRF smoothing) over challenging datasets. Our method is devoid of any manual initialization prevalent in previous approaches [52, 44] and can be parallelized to work in real time (the current Matlab implementation takes less than a second per frame). Once the dictionary is learnt, our method can be easily applied to any standard soccer broadcast and in fact can be easily extended to any sport where such field lines are available (like basketball, ice hockey etc.). Moreover, the semi supervised dictionary generation allows us to adapt the algorithm even if new camera angles are used in future. The proposed method opens up a window for variety of applications which could be realized using the projected data. One limitation of our approach is that is only applicable to top zoom-out views and it would be an interesting problem to register other kind of shots (ground zoom-in, top zoom-in shots) using the predictions over top zoom out views, player tracks and other temporal cues.

Chapter 6

Conclusion

We have introduced a series of novel solutions for different tasks of sports video analytics. Each of our method has been evaluated on challenging datasets so as to show the effectiveness of our approach across varying conditions.

In Chapter 3 we proposed a solution for fine grain temporal alignment of Cricket Video and Textual commentaries. We also presented a preview of web based demo that demonstrates the usefulness of the approach. Our method can be used to obtain a large set of annotated video clips that can be used to learn fine grain activity classifiers. As part of future work, we are trying to train our system on complete IPL 2017 Video dataset and use this data for the task of video caption/commentary generation using deep learning based approaches.

In Chapter 4 we show that use of contextual information can enhance the action recognition and can beat even the state of the art deep learning methods (which requires lots of data) with a simple SVM classifier. Our approach of extracting camera information has shown promising results on task of spatial segmentation and person detection. This approach can be also be used in other sports such as Basketball, Baseball etc.

The algorithm proposed in Chapter 5 enables us to do sports analysis on broadcast videos. The algorithms works in real time and can be easily parallelized which is essential if we want to build real life systems. Our algorithm is devoid of any manual initialization and doesn't require any additional sensors data. Large applicability of our algorithm on any broadcast video and semi-supervised dictionary generation are two novel contributions of our work. Our algorithm can also be easily extended to any other sports such as Basket Ball, Baseball etc with only a few set of manual initializations.

6.1 Applications

The proposed methods for cricket and soccer videos are generic and can be used for variety of applications. Some of the explored applications are discussed below.

Figure 6.1 Example of Cricket Commentary based search system

6.1.1 Search and Retrieval

Using algorithm proposed in Chapter 3 we can build a text based search system to search over complete Cricket match. Our system will accept text query as input and will search through textual commentary and retrieve corresponding video instance from the match. For e.g say user query for "Dhoni cover drive" would result in a collection of all instances from a match where Dhoni is playing a *Cover Drive* shot.

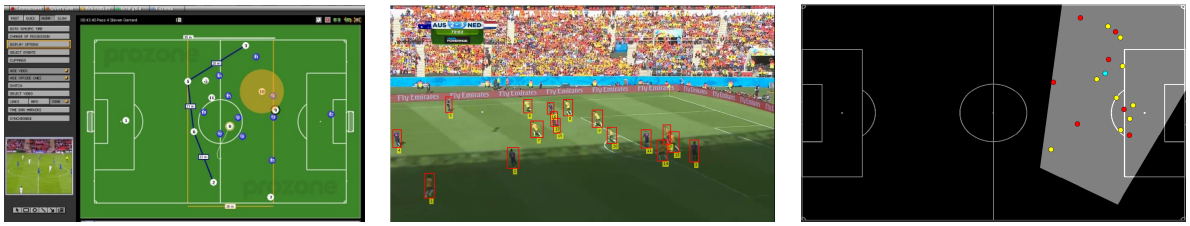
Using our algorithm proposed in Chapter 4 we can retrieve all important events such as goal, foul etc from a given soccer video. This will enable us to build a large collection of extracted event clips that can be further utilized to improve the accuracy of the system.

6.1.2 Summarization

Our proposed algorithm enables us to get per frame description for a complete sports video. This description is commentary for a cricket match and event information for a soccer match. Using these descriptions we can automatically summarize a match. For e.g. in case of cricket we can use commentaries to select important events. A simple naive solution is that if commentary contains more number of adjectives it denotes an important event. So we can only select a subset of commentaries based on adjectives and display only that information to the user. Instead of using only adjectives, many advanced NLP techniques can also be applied to process importance of commentaries. We can also use similar technique to generate highlight clips of a Soccer match.

6.1.3 Strategy Analysis

Using our proposed approach in Chapter 5 we can get top-view registration for every frame of a video. This information can easily be used to extract various high level soccer analytics from a soccer



(a) (b)

Figure 6.2 (a) A snapshot from Prozone tracking system. (b) An example result from the proposed method, which takes as input a broadcast image and outputs its registration over the static top view model with the corresponding player positions. The yellow, red and cyan circles denote the players from different teams and referee.

match. For e.g. we can now do player detection in Broadcast frames and project them over the static top-view model. We can now classify the projected bounding boxes into three categories i.e team1, team2 and referee. By tracking these player positions of separate teams across a match we can get an insight into the formation of the teams, tactics used by them, defending strategies etc. Using our approach on a large scale we can also analyze these tactics across various matches for a team.

Related Publications

1. Rahul Anand Sharma, Pramod Sankar K & C.V. Jawahar : Fine Grain Annotation of Cricket Videos, Asian Conference on Pattern Recognition (ACPR), 03-06 Nov 2015, Kuala Lumpur, Malaysia

2. Rahul Anand Sharma, Vineet Gandhi, Visesh Chari & C.V. Jawahar : Automatic analysis of broadcast football videos using contextual priors, Springer Journal on Signal, Image and Video Processing(SIVP), Volume 10, Issue 5, July 2016.

3. Rahul Anand Sharma, Vineet Gandhi & C.V. Jawahar : Automated Top View Registration of Broadcast Football Videos, Conference on Computer Vision and Pattern Recognition (CVPR), 2017 (Under Review)

4. Himangi Saraogi, Rahul Anand Sharma & Vijay Kumar : Event Recognition in Broadcast Soccer Videos, Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), 18-22 Dec 2016, Guwahati, India

Bibliography

[1] Cricinfo at: http://www.cricinfo.com.

[2] Prozone sports. `http://prozonesports.stats.com`.

[3] Tracab optical tracking. `http://chyronhego.com/sports-data/tracab`.

[4] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *CVPR*, 2015.

[5] E. André, K. Binsted, K. Tanaka-Ishii, S. Luke, G. Herzog, and T. Rist. Three robocup simulation league commentator systems. *AI Magazine*, 21(1):57, 2000.

[6] J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *CVIU*, 92:285 – 305, 2003.

[7] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[8] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, DTIC Document, 1977.

[9] A. Bialkowski, P. Lucey, P. Carr, S. Denman, I. Matthews, and S. Sridharan. Recognising team activities from noisy data. In *CVPR Workshops*, 2013.

[10] I. Bojinov and L. Bornn. The pressing game: Optimal defensive disruption in soccer. In *Procedings of MIT Sloan Sports Analytics*, 2016.

[11] C. Chen, O. Wang, S. Heinzle, P. Carr, A. Smolic, and M. Gross. Computational sports broadcasting: Automated director assistance for live sports. In *ICME*, 2013.

[12] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] I. Davidson and S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. SIAM, 2005.

[15] L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, and J. S. Jin. A unified framework for semantic shot classification in sports video. *Multimedia, IEEE Transactions on*, 7(6):1066–1083, 2005.

[16] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, 2010.

[17] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy” – automatic naming of characters in TV video. In *Proc. BMVC*, 2006.

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9):1627–1645, 2010.

[19] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[20] A. Franks, A. Miller, L. Bornn, and K. Goldsberry. Counterpoints: Advanced defensive metrics for nba basketball. MIT Sloan Sports Analytics Conference. Boston, MA, 2015.

[21] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[22] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011.

[23] J. Gudmundsson and M. Horton. Spatio-temporal analysis of team sports–a survey. *arXiv preprint arXiv:1602.06994*, 2016.

[24] A. Gupta, J. J. Little, and R. J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 32–39. IEEE, 2011.

[25] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In *Proc. CVPR*, pages 2012–2019, 2009.

[26] P. Heckbert. Color image quantization for frame buffer display. In *SIGGRAPH*, 1982.

[27] A. Hervieu, P. Bouthemy, and J.-P. L. Cadre. Trajectory-based handball video understanding. In *ICIV*, 2009.

[28] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *CVPR*, 2007.

[29] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*. 2008.

[30] V. Jain, A. Singhal, and J. Luo. Selective hidden random fields: Exploiting domain-specific saliency for event classification. In *CVPR*, 2008.

[31] R. Kapela, K. McGuinness, A. Swietlicka, and N. E. OConnor. Real-time event detection in field sport videos. In *Computer Vision in Sports*, pages 293–316. 2014.

[32] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. CVPR*, 2015.

[33] K. Kim, M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, and I. Essa. Motion fields to predict play evolution in dynamic sport scenes. In *CVPR*, 2010.

[34] R. Kindermann, J. L. Snell, et al. *Markov random fields and their applications*. American Mathematical Society, 1980.

[35] Y. Kong, W. Hu, X. Zhang, H. Wang, and Y. Jia. Learning group activity in soccer videos from local motion. In *ACCV*. 2010.

[36] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[38] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*. 2010.

[39] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.

[40] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[41] R. Li and R. Chellappa. Group motion segmentation using a spatio-temporal driving force model. In *CVPR*, 2010.

[42] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[43] W.-L. Lu, J.-A. Ting, J. Little, and K. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE PAMI*, 35(7):1704–1716, July 2013.

[44] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *TPAMI*, 35(7):1704–1716, 2013.

[45] P. Lucey, A. Bialkowski, P. Carr, S. Morgan, I. Matthews, and Y. Sheikh. Representing and discovering adversarial team behaviors using player roles. In *CVPR*, 2013.

[46] P. Lucey, A. Bialkowski, M. Monfort, P. Carr, and I. Matthews. quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. MIT Sloan Sports Analytics Conference, 2014.

[47] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *ACM Multimedia*, pages 469–478, 2012.

[48] B. Markoski, Z. Ivanković, L. Ratgeber, P. Pecev, and D. Glušac. Application of adaboost algorithm in basketball player detection. 2015.

[49] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP*, 2009.

[50] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*. Morgan Kaufmann Publishers Inc.

[51] N. Nguyen and A. Yoshitaka. Shot type and replay detection for soccer video parsing. In *Multimedia (ISM), 2012 IEEE International Symposium on*, pages 344–347, 2012.

[52] K. Okuma, J. J. Little, and D. G. Lowe. Automatic rectification of long image sequences. In *ACCV*, 2004.

[53] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[54] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[55] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[56] Pramod Sankar K., S. Pandey, and C. V. Jawahar. Text driven temporal segmentation of cricket videos. In *Proc. ICVGIP*, pages 433–444, 2006.

[57] X. Qian, G. Liu, Z. Wang, Z. Li, and H. Wang. Highlight events detection in soccer video using hcrf. In *Proceedings of the Second International Conference on Internet Multimedia Computing and Service*, pages 171–174, 2010.

[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[59] K. P. Sankar, S. Pandey, and C. Jawahar. Text driven temporal segmentation of cricket videos. In *Computer Vision, Graphics and Image Processing*, pages 433–444. Springer, 2006.

[60] W. Shen and L. Wu. A method of billiard objects detection based on snooker game video. In *ICFCC*, 2010.

[61] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[62] H. B. Shitrit, J. Berclaz, F. Fleuret, , and P. Fua. Tracking multiple people under global appearance constraints. 2011.

[63] M.-H. Sigari, H. Soltanian-Zadeh, V. Kiani, and A.-R. Pourreza. Counterattack detection in broadcast soccer videos using camera motion estimation. In *AISP*, pages 101–106, 2015.

[64] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[65] H. Soo Park and J. Shi. Social saliency prediction. In *CVPR*, 2015.

[66] E. Swears, A. Hoogs, Q. Ji, and K. Boyer. Complex activity recognition using Granger Constrained DBN (GCDBN) in sports and surveillance video. In *Proc. CVPR*, pages 788–795, 2014.

[67] A. Thayananthan, B. Stenger, P. H. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, 2003.

[68] R. Thompson and C. Bowen. *Grammar of the Edit*. Focal Press, 2009.

[69] R. Walecki, O. Rudovic, V. Pavlovic, and M. Pantic. Variable-state latent conditional random fields for facial expression recognition and action unit detection. In *Automatic Face and Gesture Recognition*, 2015.

[70] G. Waltner, T. Mauthner, and H. Bischof. Improved sport activity recognition using spatio-temporal context. In *DVS/GSSS*, 2014.

[71] G. Waltner, T. Mauthner, and H. Bischof. Indoor activity detection and recognition for automated sport games analysis. In *AAPR/OAGM*, 2014.

[72] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[73] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Structure analysis of soccer video with hidden markov models. In *ICASSP*, 2002.

[74] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan. Live sports event detection based on broadcast video and web-casting text. In *Proc. ACM Multimedia*, pages 221–230, 2006.

[75] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan. Live sports event detection based on broadcast video and web-casting text. In *ACM Multimedia*, pages 221–230, 2006.

[76] G. Xu, Y.-F. Ma, H.-J. Zhang, and S.-Q. Yang. An hmm-based framework for video semantic analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(11):1422–1433, 2005.

[77] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and system for segmentation and structure analysis in soccer video. In *ICME*, volume 1, pages 928–931, 2001.

[78] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 2003.

[79] Y. Zhang, X. Zhang, C. Xu, and H. Lu. Personalized retrieval of sports video. In *Proc. Intl. Workshop on Multimedia Information Retrieval*, pages 313–322, 2007.

[80] G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao. Event tactic analysis based on broadcast sports video. 2009.

[81] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 2009.