AN ABSTRACT OF THE THESIS OF

Qingkai Lu for the degree of Master of Science in Computer Science presented on June 10, 2015.

Title: Offensive Direction Inference in Real-World Football Video

Abstract approved: _____

Alan P. Fern

Automatic analysis of American football videos can help teams develop strategies and extract patterns with less human effort. In this work, we focus on the problem of automatically determining which team is on offense/defense, which is an important subproblem for higher-level analysis. While seemingly mundane, this problem is quite challenging when the source of football video is relatively unconstrained, which is the situation we face. Our football videos are collected from a web-service used by more than $13,000$ high school, college, and professional football teams. The videos display huge variation in camera viewpoint, lighting conditions, football field properties, camera work, among many other factors. These factors taken together make standard off-the-shelf computer vision algorithms ineffective. The main contribution of this thesis is to design and evaluate two novel approaches for the offense/defense classification problem from raw video, which make minimal assumptions about the video properties. Our empirical evaluation on approximately 1200 videos of football plays from 10 diverse games validate the effectiveness of the approaches and highlight their differences.

Offensive Direction Inference in Real-World Football Video

by

Qingkai Lu

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 10, 2015
Commencement June 2015

Master of Science thesis of Qingkai Lu presented on June 10, 2015.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Qingkai Lu, Author

ACKNOWLEDGEMENTS

Firstly, I would like to express my appreciation to my advisor Prof. Alan Fern, who guided me through my master studies with great support and help. I would like to thank you for giving me the opportunity to work on this interesting research project and providing me precious suggestions to learn and improve. I also would like to appreciate Prof Todorovic's help in my research project.

Secondly, I would like to thank the rest of my thesis committee members: Prof. Borello, Prof. Bailey and Prof. Tadepalli. I appreciate your support and time.

Thirdly, I would like to thank our group members(especially Sheng), who helped me a lot in this project.

Last but not the least, I would like to thank my family and friends, especially my parents and two sisters, who has been supporting and encouraging me all the time.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

1.1 Motivation

Activity recognition is an important research area in the field of computer vision, which aims at automatically analysing ongoing events and their contexts from video data [29]. Various state-of-the-art techniques have been developed for these prevalent activity recognition applications, such as security surveillance, patient monitoring and human-computer interactions. However, the activity recognition in sports domain is still under-served.

American football is the most popular sport in the United States. The game planning of American football teams needs a lot of annotation and analysis of videos of their own and opponent games. However, the existing services for football game planning still involve large amount of manual management, annotation, and analysis of videos, since they usually provide only essential user interface functionalities. Even only for one game, video annotation needs to be done repetitively for an average of around 150 plays. Moreover, humans annotations tend to commit errors due to the complex and repetitive nature of football videos. Thus, it will be significant to automate at least part of the annotation process. Automatic analysis and classification of sports play can help sports coaches and analysts to extract patterns and develop strategies from large collections of sports videos. In this work, we aim at automatically classifying the offensive-defensive video plays of football games into one of two types with different offensive directions. Developing techniques for the offensive direction inference will be

helpful for game analysis and planning. Moreover, it can serve as one building block to solve more complex problems, such as the play type recognition of all plays comprising a football game [7].

1.2 Background and Challenges

American football video analysis is conducted around the concept of football plays. A play is a video clip lasting approximately 10 to 30 seconds. One football game usually contains a sequence of approximately 150 plays. One video clip for each play in the game is captured following the temporal order to record a whole football game. Consecutive plays are separated by short time intervals, during which no game action happens and the teams rearrange. Our football videos are recorded by a PTZ camera from an elevated location along the sideline, which captures a sideline view of the football field(e.g. Figure 3.2).

According to the football taxonomy, each play has a distinct type. The two most common play types are *Offense (O)* or *Defense (D)*. The offense(O), the team with control of the football, aims to move the ball forward by running with or passing the ball, while the team without control of the ball, the defense(D), attempts to stop their advance and take control of the ball. The *offensive-defensive(OD)* play refers to the play which involves one team on offense and the other opponent team on defense. Before each OD play starts, two teams lines up facing each other at *the line of scrimmage (LOS)* — an imaginary line parallel to the field lines upon which the ball is placed. An OD play begins at the *moment of snap(MOS)*, when the center throws or hands the ball backward to one of the backs (snap), usually the quarterback, then both teams start moving and executing their own strategies until the play ends. A *formation* of a play is defined as the

Figure 1.1: Mos frames of 6 football plays which show the huge huge diversities of our football videos.

spatial configuration of players at the moment of snap, which includes both the offensive formation and the defensive formation of that particular play.

Considering the huge diversities of our football videos, we believe no off-the-shelf computer vision tools are capable of inferring the offensive direction effectively. Our videos have large variance in camera viewing angles/distances, video shot quality, weather/lighting conditions, the color/sizes/patterns of football field logos/markings, and the scenes around the field, ranging from crowds, to players on the bench, to construction equipments. Also, these videos are typically captured by amateurs, which leads to frequent motion blur, camera jitter and large camera motion. Further, videos of different games have large variations in team strategies, formations, and uniforms color. All these factors make video registration, frame-references, and background subtraction rather challenging, which are critical for existing approaches [7].

1.3 Problem Statement and Overview of Our Approaches

In our work, we focus on classifying OD plays into two types with different offensive directions. To define our objective formally, our input is a sequence of temporally ordered videos comprising all OD plays of football games and the expected output is an accurate labeling of the offensive direction of each play to be left or right.

We have two approaches to achieve our goal. Both our approaches make use of the formation (spatial layout of players) difference, which is the essential information to distinguish the two types of plays with different offensive directions. Our first approach, the Kanade-Lucas-Tomasi (KLT) [32] trajectories analysis, seeks to detect the position of a special type of player—the wide receivers (WRs) to predicate the offensive direction. Our second approach, spatial pyramid matching method, estimates the difference between formations of plays with different offensive directions, in terms of players spatial layout.

Chapter 2: Related Work

2.1 Activity Recognition

There is a large body of existing work on action recognition. Most of this work focuses on detection of human activities for a single person [14] [12] [22]. There is also some work that aims to recognize the actions of groups in more complex contexts, such as movies [24] or sport videos [36]. Although there are various approaches to do action recognition, a majority of them use the same framework which includes three main steps: extracting local features from videos, constructing a representation of the video in terms of these local features and finally, classification [34]. Both our methods, including the KLT trajectories analysis and spatial pyramid matching, fit in this framework. In the first step, different local features such as histograms of oriented gradients(HOG) [33] or trajectories [41] are extracted from videos. In the second step, each video is represented by a certain form of its extracted features, such as a histogram [33] [41] or a semantic model [40], to capture activities. Finally, supervised learning models such as SVM [3] or random forest [30] can be trained using the computed representations of the labeled training set.

2.2 American Football Video Analysis

Sports video analysis is one of these applications of activity recognition that focuses on sports domain. In our work, we mainly focus on the video analysis of one specific sport-

American football. Most of the work on American football analysis focuses on either classifying different types of plays or recognizing football formations. In [3], a framework for automatic recognition of offensive team formations in American football plays is proposed. As one stage of this framework, a method based on the spatial distribution of players is used to infer the offensive direction(see more details in Section 3.3). Siddiquie et al. [33] come up with a learning based method under discriminative feature selection framework for recognizing plays in American football games. In their feature selection framework, the spatial pyramid matching is used to capture information of the spatial and temporal distribution of the local features, which is similar to our spatial pyramid method, but has different types and dimensions of local features compared with our method. Using the players' trajectories as features, a probabilistic generative model is introduced to recognize American football plays in [27]. Swears and Anthony [37] propose a non-stationary kernel HMM approach which processes player trajectories to recognize American football play types. In [19], a mixture of pictorial-structure model is used to recognize football formations by locating and identifying players. To classify American football plays into different camera view types, a top-down HMM-based video representation model is developed in [11]. Due to the challenges of our data-set (Section 1.2), we believe it is beyond the capabilities of these existing football video analysis approaches with restricted assumptions to be successful for our data-set. Specifically, these assumptions can be overhead view of football field, video registration, background subtraction (e.g. [3]), video stabilization (e.g. [37]), access of certain video features such as players' trajectories (e.g. [27] [37]) or spatial-temporal interest points (e.g. [33]).

Chapter 3: Approaches

We will describe three different approaches in this section. The KLT trajectories analysis seeks to detect the wide receivers to predicate the offensive direction. It first detects the initial motion of one or more WRs in a football play, then we know that the offense is on the side opposite to the direction of this motion. The spatial pyramid matching method estimates the players' spatial layout difference between plays to distinguish their offensive directions. It applies the spatial pyramid to extract formation features from the players' foreground football plays, then train a classifier to predict the offensive direction using the extracted features and offensive direction labels of our training set. The method based on the spatial distribution of players infers the offensive direction by applying the insight that offensive formations tend to have smaller variance of players' spatial distribution than defensive formations. The player distribution of the offensive or defensive team of a play is modelled by a spatial pmf, and the offensive team is determined as the team with a smaller pmf variance.

Both the KLT tracks and foreground aim to represent the players in different indirect ways, which aim to provide us the necessary information of players spatial configuration of a play to infer its offensive direction. A player detector is a more direct and popular way to detect players, why don't we use a player detector? With a robust player detector, we can easily and explicitly analyze the spatial configuration of players of a play to infer its offensive direction. However, it turns out to be rather difficult to have a robust player detector for our football videos. For example, we trained a deformable part based model (DPM) [16] [13] for players detection, which is one of the most successful model for

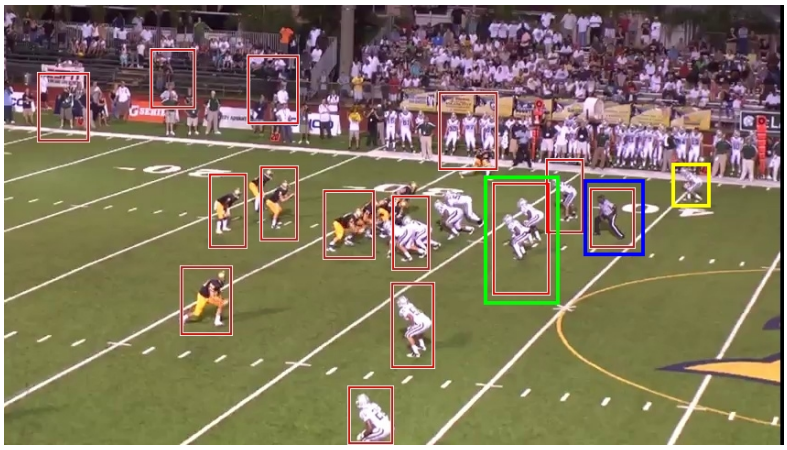

Figure 3.1: Red bounding boxes represent DPM player detections. There are three problems here: the player in the yellow box is missing, the referee in the blue box is detected as a player, and two close players in the green box are detected as one player.

object detection recently. The testing results show that a DPM model is not capable to detect all players correctly (e.g. Fig. 3.1) for our videos, mainly due to the limitation of the model itself and the challenges of our data-set. Thus, we resort to features that are more feasible to generate compared to player detections, such as KLT trajectories and players foreground. KLT trajectories and foreground can be noisy, but they can be exploited to infer the offensive direction with reasonable analysis.

3.1 KLT Trajectories Analysis

3.1.1 Motivation and Overview

We seek to detect a certain type of offensive players called, *wide receivers (WRs)*, whose motion is predictive of which side of the LOS the offense is on. As shown in Fig. 3.2, WRs are players that usually line up at the ends of the field, and are isolated from the other offensive players. A majority of videos of OD plays show at least one WR. After a

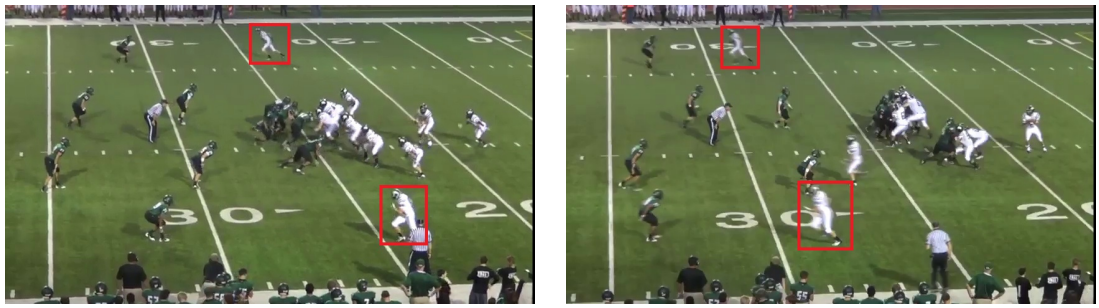

Figure 3.2: Two frames showing two wide receivers (in red boxes) running from right to left.

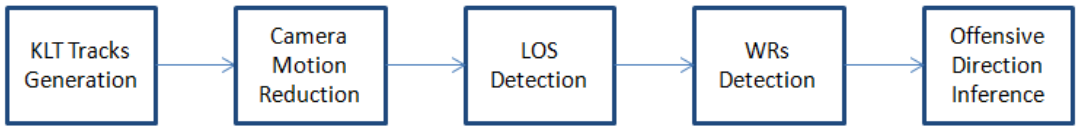

Figure 3.3: Pipeline of KLT tracks analysis method.

play starts, the WRs will almost always immediately run in the direction of the defense. Thus, our approach for offensive direction inference is to detect the initial motion of one or more WRs, then we know that the offense is on the side opposite to the direction of the motion. Note that there are rare cases of OD plays where there are no WRs on the field. In these cases, our approach will produce arbitrary results. The pipeline of our KLT track analysis method can be seen from Fig 3.3. Each step of the pipeline will be described in detail in the rest of this section.

3.1.2 KLT Trajectories Generation and Pre-processing

As an efficient and simple feature extraction approach, the Kanade-Lucas-Tomasi (KLT) makes use of the spatial intensity information to search the best match position in the image [32]. To detect WRs, we extract and analyze KLT trajectories in order to infer

which ones are likely due to WRs, and then infer the WR motion direction based on those trajectories. We run the KLT tracker on a sequence of 45 video frames following the predicted MOS. The extracted point trajectories typically correspond to player movement, non-player foreground motion, and background features under camera motion. When a WR is visible, the KLT trajectories are capable of extracting relatively long trajectories due to the WR's characteristic swift and straight-line motion. To help remove KLT trajectories caused by camera motion, we use the field lines extracted with the method described in [7] to pre-process the KLT trajectories, by measuring the relative motion of each KLT trajectory to its closest field line. Ideally, this relative motion is small when a KLT trajectory corresponds to a stationary background feature. This allows removing all KLT trajectories whose relative motion falls below a threshold. The specific threshold choice is not critical, since the KLT trajectories of WRs generally have very large relative motion.

3.1.3 LOS and Wide Receivers Detection

Given the remaining KLT trajectories, the key to inferring which ones belong to WRs is to use our knowledge that: i) WRs are typically located at the far ends of the LOS, and ii) WRs are usually isolated from the majority of players lined up at the LOS. This requires first inferring the LOS. We know that the LOS is supposed to have relatively large amount of players' movements when the play starts, because the LOS is usually in the region of highest players intensity. The background features such as the logo of the field might also have large amount of movements due to camera motion, which is taken care of by our camera motion reduction method mentioned above. Further, we know that the LOS is always parallel to the field lines. We use a sliding window to scan

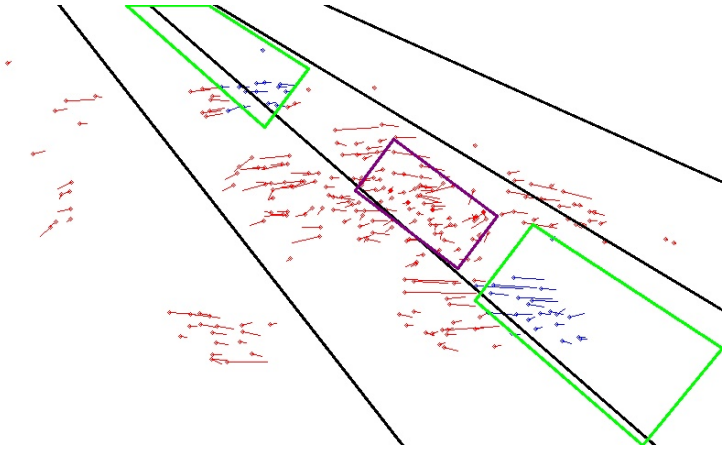

Figure 3.4: Plot of one video's KLT trajectories with LOS bounding box (purple rectangle) and ranges for WRs (green rectangles). Red (blue) vectors are the KLT trajectories outside (inside) the range of the WRs.

across the frame for the region of highest KLT trajectories intensity, where the medial axis of the window is enforced to pass through the estimated vanishing point of the field lines (i.e., to be "parallel" under perspective to the field lines). As we scan, the window size is adapted to the distance between the detected field lines. The sliding window with maximum number of KLT trajectories gives a good estimate of the location of the offensive and defensive lines of players. We use two other windows on each side of this maximum-gradient window to identify likely areas where WRs are located. Figure 3.4 shows an example of extracted KLT trajectories, with the maximum KLT trajectories intensity window, and the two WR windows.

3.1.4 Offensive Direction Inference

Finally, we estimate the motion direction of WRs using a weighted average of the KLT trajectories in the two WR windows. In particular, we linearly weight each KLT track according to its distance from the center of the maximum-gradient window. This assigns

higher weights to KLT trajectories that are more isolated, and hence more likely to be due to WR's motion. We use the direction of the vector resulting from the weighted average, relative to the field lines, as the estimated direction of the WR. Note that in cases where either the upper or lower WR is not present in the video, this combination will tend to be biased in favor of the visible WR, since there will typically be few highly weighted KLT trajectories in the WR window not containing a WR [7].

3.2 Spatial Pyramid Matching

3.2.1 Introduction of Spatial Pyramid Match Kernels

To understand our spatial pyramid matching method, we will introduce the definition and methodology of the pyramid match kernel and the spatial pyramid match kernel in this section. By treating these two types of plays with different offensive directions as two types of scenes, the offensive direction inference can be solved as a scene recognition problem. Recognizing the semantic scene category of an image is a significant problem in computer vision. *Bag of features* [15] [39] represents an image by an orderless collection of local features, which has achieved impressive performance to recognize the semantic category of images. However, bag of features disregard the spatial layout of features, which leads to limited ability to describe objects or scenes. Moreover, it is not able to localize the position of objects or capture the shape of objects without using the geometrical information. To take advantage of the global geometrical correspondence of different parts of objects, such as human or football formations, the spatial pyramid matching [25] [26] method is proposed. Spatial pyramid matching repeatedly subdivides images into increasingly finer sub-regions and computes the histogram of local features inside

each sub-region. As a kernel-based method, it computes the geometric correspondence on a global scale by applying an efficient approximation method based on the pyramid matching scheme of Grauman and Darrell [18]. The experiments in [25] [26] show that the spatial pyramid method significantly outperforms the bag-of-features representation.

Pyramid matching measures the similarity between two sets of feature vectors. It repeatedly subdivides the feature space into a sequence of increasingly finer grids and computes the weighted sum of the number of the matched features inside each grid at all resolution levels. At a certain resolution level, two feature points in the feature space are said to be a match if they fall into the same grid. Let X and Y be two sets of feature vectors in a d dimensional feature space. The feature space is repeatedly subdivided to resolution levels $0, ..., L$. Each dimension of the feature space in the lth level is split into 2^l cells. In total, lth level has $D = 2^{dl}$ grids. Let H_X^l and H_Y^l be the histogram of the number of features of all grids at lth resolution level for X and Y. $H_X^l(i)$ and $H_Y^l(i)$ represent the number of features falling into the ith grid at lth resolution level of X and Y respectively. The histogram intersection function [35] of Equation 3.1 computes the number of matches between X and Y at lth level.

$$I(H_X^l, H_Y^l) = \sum_{i=1}^{D} min(H_X^l(i), H_Y^l(i)) \tag{3.1}$$

To be convenient, we will use I^l to represent $I(H_X^l, H_Y^l)$. Since each grid at level l can be decomposed into multiple finer grids at level $l+1$, the matches at level l include all the matches at level $l+1$. The number of new matches at level l is $I^l - I^{l+1}$. Intuitively, the matches in coarser resolution levels contain less important correspondence information because coarser levels involve increasingly dissimilar features. Therefore, the weight of level l is set to $\frac{1}{2^{L-l}}$, which is inversely proportional to the grid size. The *pyramid kernel*

is defined by equation 3.2 as a Mercer kernel [18].

$$k^L(X,Y) = I^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}}(I^l - I^{l+1}) \tag{3.2}$$

Pyramid matching performs matching of bag-of-features in appearance(feature) space, but totally ignores the spatial information. While, spatial pyramid matching performs matching of features in 2-dimensional image space and clustering of features in feature space. Assume the features extracted from one image are clustered into M discrete types. Each type of features has two sets of 2-dimensional vectors X_m and Y_m, which represent the image coordinates of features of the image X and Y respectively. The *spatial pyramid kernel* is computed by summing the kernels from all M types:

$$K^L_{X,Y} = \sum_{m=1}^{M} k^L(X_m, Y_m) \tag{3.3}$$

Spatial pyramid kernel is actually a weighted sum of histogram intersections of X_m and Y_m for all types of features, which is shown by Fig 3.5. To keep the total number of features in all images to be consistent, each histogram is normalized by the total weight of all features in the images.

The order of histogram is important to estimate the spatial layout correspondence between different images using spatial pyramid matching. For a set of images with the same scene(e.g. highway, mountain, forest, etc), the ith grid at the lth resolution level is supposed to represent approximately the same part of that particular scene in these different images with the same scene. In [25], it assumes different images of the same scene have the approximate same scale and occupies the approximate same area to guarantee the order. Note that the spatial pyramid kernel not only takes advantage of

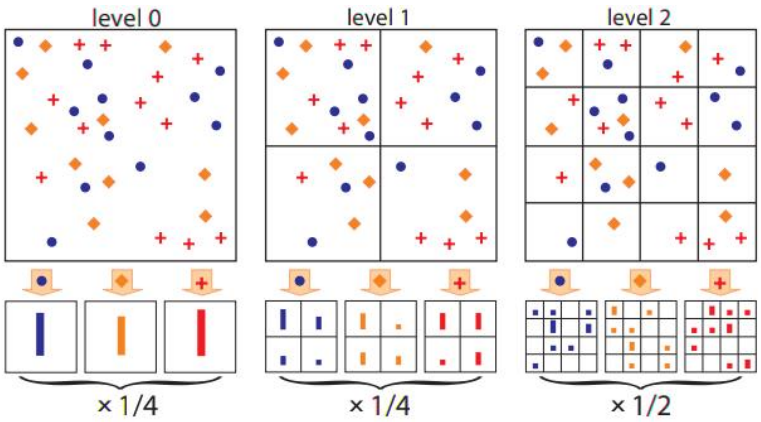

Figure 3.5: A simple example of a three level pyramid[18]. Three types of features are represented by circles, crosses and diamonds respectively. Firstly, the image is subdivided into 3 levels. Secondly, count the number of features inside each grid to construct the histogram. Finally, compute the weight of different resolution levels using equation 3.2

the spatial layout information of features, but also maintains a rich visual vocabulary by clustering features. According to the experiments in [25], spatial pyramid matching has achieved promising results on multiple challenging datasets, which is significantly better the orderless bag-of-features method.

3.2.2 Motivation and Overview

The spatial pyramid makes use of the spatial layout and the visual vocabulary of features to recognize scenes. Both the offensive and defensive team formations have their own pattern of players spatial configuration, which are totally different from each other. Thus, the formation of the two types of plays with different offensive directions will have vital difference in terms of the players spatial layout. By treating these two types of plays as two different scenes, it is natural to apply spatial pyramid matching to infer the offensive direction. The input of each play is the registered foreground of its MOS frame. The

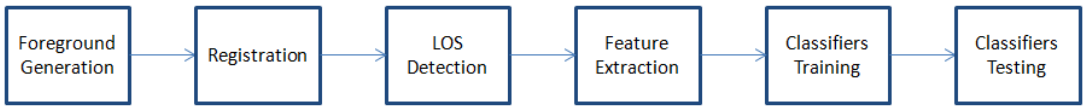

Figure 3.6: Pipeline of spatial pyramid matching method.

players' foreground pixel is the local feature. Our spatial pyramid matching method follows the pipeline shown by Fig 3.6. All stages of the pipeline will be introduced one by one in rest of this section.

3.2.3 players' foreground Generation

Different techniques can be applied to generate players' foreground for football videos. For example,recent work in [3] firstly performs video registration and constructs a background model, then generates plays foreground via background subtraction. As mentioned in section 1.2, our videos have large variance in terms of camera viewing angles/distances, video shot quality, weather/lighting conditions, the color/sizes/patterns of football field logos/markings and the scenes around the field, it is impractical to construct a background model for each football play of our data-set. But scene segmentation is feasible to detect players' foreground of our football videos. As one of the relatively successful methods in scene segmentation, conditional random field [38] is applied to segment players' foreground for the MOS video frame of each OD play in our data-set. The MOS frame of an OD play includes all its formation information, which contains significant patterns to infer the offensive direction. By focusing on the MOS frame, it not only saves the effort of processing and analyzing multiple video frames, but also gets rid of the effect of camera motion. The visual quality of one of our foreground results can be seen from Fig 3.7. As we can see, the foreground results could have some noise(e.g.

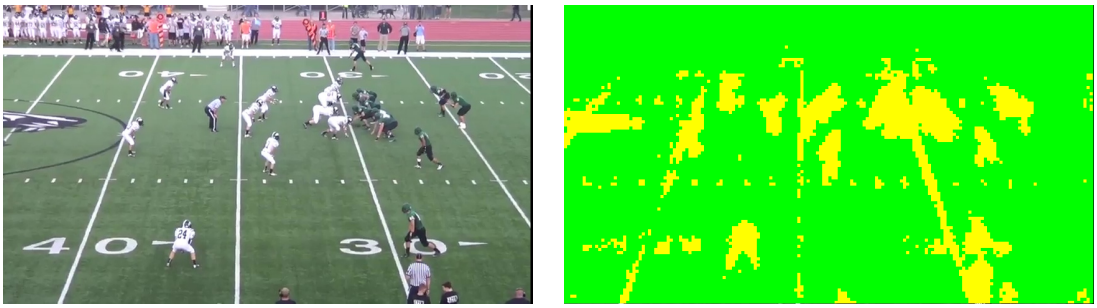

Figure 3.7: The left image is the MOS frame of a play. The right image is the foreground of the left image. In the right image, the yellow pixels represent the players' foreground and the green pixels represent the background.

generated by yard lines and field logos), but their quality satisfies the need to infer the offensive direction, which can be shown by the experiments.

3.2.4 Registration

Image registration refers to the process of transforming a set of images into the same coordinate system [5] [17]. Due to different camera views, the size of players and the distance between players become different for different football plays in the image space, which should be the same in the football field space. Even for the same play, the same problem exists because of the perspective. For example, the size of players and the distance among players will be shrinked at these positions further away from the camera in the image space. To make both the players' size and the distance between players well normalized both within the same play and across different plays, it is necessary to register the MOS frame of all plays of one game to the same football field coordinate. The registration of foreground is done by applying the perspective transform to the image with the homography [1] estimated from the point correspondences labelled manually. To compute the homography, we need at least 4 corresponding pairs of points between

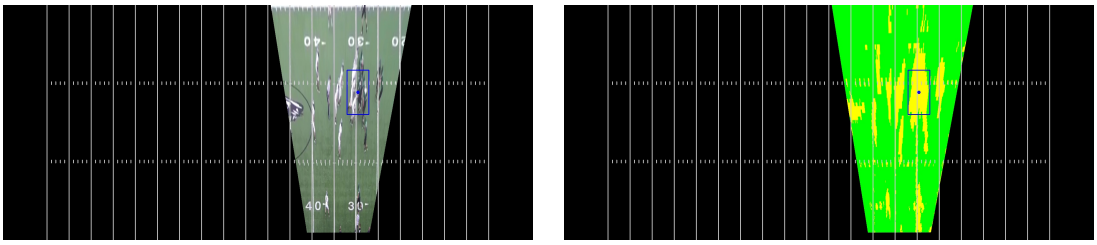

Figure 3.8: The left image is the registration result of the MOS frame of Fig 3.7. The right image is the registration result of the foreground of Fig 3.7. The blue bounding boxes represent the LOS detection result in both images. The blue dots inside the LOS bounding box are the detected LOS center.

the MOS frame and the football field model. These corresponding pairs of points can be annotated efficiently by making use of these intersection points between the hash lines and the yard lines/field boundaries. One registration example of foreground and the MOS frame can be seen from Fig 3.8. As can be seen from this example, both the players' size and the distance between players are well normalized after the registration.

3.2.5 LOS Detection

As mentioned in Section 3.2.1, the order of features matters for spatial pyramid matching. To guarantee the ith spatial pyramid grids of these different plays with the same offensive direction represent approximately the same part of the formation, we use the LOS center as an anchor point to construct spatial pyramid for each football play.

We don't use the same LOS detection result of the KLT tracks analysis here, because it is necessary to keep the spatial pyramid matching independent with the KLT tracks analysis. Player foreground intensity is supposed to be highest at the LOS region compared to other areas in the rectified foreground of MOS frame. A sliding window method is used to find the region with highest player foreground intensity as the LOS region.

However, since our players' foreground results are generated from scene segmentation, the field logos are possible to be labelled as foreground, because their textures are different from the other parts of the football field. Moreover, the noise of a field logo can not be ignored since its area can be as large as or even larger than the area of LOS region. For example, the logo region becomes a false positive result of the LOS detection in the middle image of Figure 3.9 by finding the area with the highest players' foreground intensity. To solve this problem, we make use of the information that the uniform color of the offensive players on one side of the LOS is usually hugely different from the uniform color of the defensive players on the other side of LOS. In contrast, a field logo usually has consistent color from one side to the other. Thus, the LOS region can be detected by using these two properties: i) highest player intensity property around LOS region and ii) the color difference between two sides of LOS.

Let $fgScore(x, y)$ represent the number of foreground pixels inside the sliding window centered at position (x, y). Let $clrScore(x, y)$ be the absolute value of the average color difference between the left half and right half of the sliding window centered at position (x, y). The LOS score of the sliding widow centered at position (x, y) can be computed by summing the normalized foreground score and the normalized color score, which can be seen from equation 3.4. In equation 3.5, the position (x, y) whose sliding window has the best LOS score will be detected as the LOS center and the corresponding sliding window will be detected as the LOS bounding box. The right image of Figure 3.9 shows that the correct LOS detection is achieved by incorporating the team color difference for the same play of the middle image in this Figure. To be specific about the size of the sliding window in our experiment, let d be the distance between adjacent yard lines(i.e. 5 yard on the football field or 75 pixels in our football field model), the width of the sliding window is d and the length is $2d$.

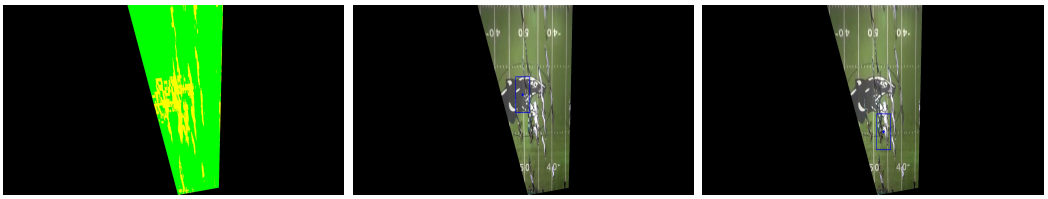

Figure 3.9: The left image is the registered foreground of the play p. The LOS detection of p using foreground can be seen from the middle image. The LOS detection of p using both foreground and color is shown in the right image. The blue rectangle represents the LOS bounding box in both the middle and right image.

$$score(x,y) = fgScore(x,y)/max_{(x,y)}fgScore(x,y) + clrScore(x,y)/max_{(x,y)}clrScore(x,y) \tag{3.4}$$

$$\text{LOS center} = argmax_{(x,y)}score(x,y) \tag{3.5}$$

3.2.6 Feature Extraction

After having the LOS center, the spatial pyramid is applied to extract the formation feature vector from the registered foreground of each play. Our spatial pyramid has four resolution levels. The ith level has 2^{2l+2} spatial pyramid cells, which is different from the definition of the spatial pyramid introduced in section 3.2.1. In section 3.2.1, the spatial pyramid has exactly 1 grid at the first level, while it has 4 in our case. There are $M = 340$ grids at all four resolution levels in total. Still, let d be the distance between adjacent yard lines. Each level of the same football play covers the same $16d$ by $16d$ square region centered at the LOS center in the registered foreground, which represents both the offensive and defensive team region. The local appearance feature

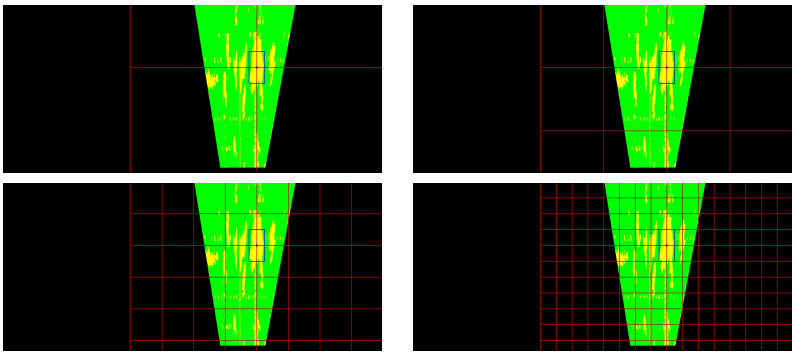

Figure 3.10: An example of our four-level spatial pyramid. Each level is represented by one image. Red rectangles represent the spatial pyramid grids. Blue rectangle is the LOS bounding box. Blue dot is the LOS center.

of the spatial pyramid is the foreground pixel, and the histogram of one spatial pyramid grid is computed by counting the number of foreground pixels inside this grid. The total formation feature vector is obtained by concatenating the histogram of all cells at all resolution levels following a specific order relative to the LOS center. An example of our four-level spatial pyramid can be seen from Figure 3.10.

3.2.7 Classifiers Training

Given the formation feature vectors and offensive direction labels of the OD football plays in the training set, our goal is to train a classifier to predict the offensive direction for football plays. We use three kinds of supervised classifier: SVM, random forest and KNN. All these trained classifiers are tested with leave-one-play-out validation, whose results can be seen from the experiments.

SVM is an efficient model for classification. SVM represents each training example as one point in the feature space and attempts to make the gap as wide as possible to separate training examples of different categories [10] [6]. Testing examples are mapped

into the same feature space and classified to a category according to which side of the gap they fall into. Considering the complexity and variance of the spatial layout of different formations, it is mostly possible that the formation features are not linear separable in the original feature space, which makes linear SVM not suffice. Fortunately, SVM is able to perform classification for non-linear separable data-sets by using the kernel trick[20] [21], which maps the training data points into an implicit high-dimensional feature space by applying kernel functions. The kernel function is equivalent to an inner product in the implicit transformed feature space, which makes the computation more efficient than the explicit computation of the coordinates. We already defined the spatial pyramid matching kernel in equation 3.3 for our problem, which measures the similarity between formations. Therefore, it is natural to train a SVM classifier with the spatial pyramid kernel to infer offensive directions for football plays.

Random forest is an ensemble classifier, which constructs multiple decision trees to perform training and predicts the output as the majority vote of the individual decision trees [8] [4]. By training a set of individual decision trees, random forest overcomes the overfitting problem of the decision tree to the training set. A decision tree divides the feature space into axis-parallel rectangle regions and labels each rectangle region into one of the K classes [31], which makes random forest a reasonable classifier for the offensive team direction inference. Intuitively, the random forest will divide the feature space into different regions based on the spatial layout of all players to predict the offensive direction.

The k-Nearest Neighbours (kNN for short) is a non-parametric classifier [2]. The testing example's k closest training examples in the feature space are picked as its neighbours, and the testing example is classified by a majority vote of the classes of these k closest neighbours [9]. k is usually a small positive number. kNN is also a reasonable

classifier for the offensive inference, because these OD plays with the same offensive direction tend to have similar formations.

3.3 A Method Based on the Spatial Distribution of Players

Both the offensive and defensive formation have repetitive spatial structures. The difference between the spatial layout of the offensive and defensive formation can be exploited to distinguish them. The positions of the offensive team players are usually more compact on the football field, especially around the LOS or along the horizontal(y axis) direction of the field, while the players in defensive formation are usually more spread in order to increase the defense area, which can be seen from the formation of Figure 3.11 and Figure 3.12. Based on this difference between the offensive and defensive formation, in [3], the offensive direction is determined as the side of the LOS whose spatial distribution of players' foreground has smaller variance. The player distribution on one side of LOS is modelled by a spatial pmf as shown in Equation 3.6. The probability that a player exists at position (x, y) in the registered foreground is computed as a function of the foreground value $fg(x, y)$ at (x, y). The side d which has smaller variance is determined to be the offensive direction, as shown in Equation 3.7.

$$p(x, y|d) = \frac{fg(x, y)}{\sum_{(x,y)\in d} fg(x, y)}; [d \in \{left, right\}] \tag{3.6}$$

$$\text{offense team direction} = \underset{d\in\{left,right\}}{\text{argmax}} \ \sigma(p(x, y|d)) \tag{3.7}$$

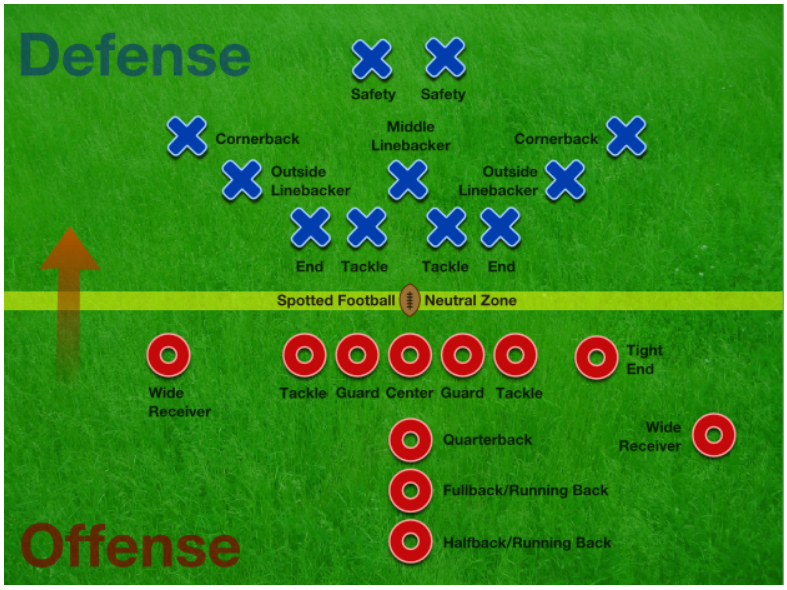

Figure 3.11: An example of football formation from Wikipedia (*http* : *//en.wikipedia.org/wiki/American*$_f$*ootball*$_r$*ules*). Red O symbols represent offensive players, blue X symbols represent defensive players. In this diagram, the offense is in the I formation and the defense is in the 4-3 formation, which are both common formations.

Figure 3.12: The MOS frame of one football play in our dataset. The left yellow team is on defense and the right white team is on offense.

Chapter 4: Experiments and Results

In this section, we will first describe the data-set for our experiments. Then we will show the overall quantitative results for these three approaches: the KLT trajectories analysis, spatial pyramid matching with different classifiers, and the method based on players spatial distribution. To know how our LOS detector works, we will also evaluate the LOS detection accuracy based on the ground truth of the LOS. Then experiments with different parameters are performed on spatial pyramid matching in order to analyze it in further detail. Specifically, we will discuss the SVM with different kernels and the KNN with different K. Finally, we will discuss the running time of each method.

4.1 Data-set

Our data-set of study is provided by a large company which offers the web service of football videos for over 13,000 high school, college, and professional teams. Specifically, our data-set includes 10 diverse, real-world football games, which were selected by the web-service company from their database in an attempt to cover the wide diversity of football videos [7]. The football video diversity includes camera viewing angles/distances, video shot quality, weather/lighting conditions, the color/sizes/patterns of football field logos/markings, and the scenes around the field. These selected 10 games have 1450 plays in total, among which 1190 plays are on OD. The true MOS and team direction of each play are annotated to allow for more refined evaluations. The MOS can also be predicted with relatively good accuracy on average using our approach in [28]. The

automatic registration (e.g. the automatic registration method in [3]) doesn't have good performance for our challenging data-set, but we need the registration for both the spatial pyramid matching and the player distribution method. Thus, we manually annotated the corresponding pairs(at least 4) of points between the original MOS frame and the football field model to perform registration for all OD plays of 4 games(436 OD plays). Moreover, we did the annotations of the LOS center for all OD plays of 1 game(101 OD plays) to validate our LOS detection methods. Since it takes a lot of manual effort to do the annotation, especially for the corresponding points of registration, we didn't annotate all 10 games.

4.2 Quantitative Results

4.2.1 Accuracy of Different Methods

The accuracy of different methods for both true MOS and predicted MOS can be seen from Table 4.1. The KLT trajectories analysis is tested on all 10 games(1119 plays). The spatial pyramid matching method and the method based on the players spatial distributions are tested on the 4 games(436 plays) with registration annotation, since they need the registered foreground. To compare the accuracy of the KLT trajectories analysis with other methods, the KLT trajectories analysis method is also tested on these 4 games(436 plays) with registration annotation. Leave-one-play-out is used to test the spatial pyramid matching with different classifiers(SVM, random forest and KNN) in order to access how the model will generalize to an independent data set [23], whose accuracy can be seen from Table 4.1.

Four conclusions can be made from the accuracy of all methods. Firstly, the accuracy

of the predicted MOS is close to the accuracy of the true MOS, which shows that our predicted MOS is reliable enough. Secondly, spatial pyramid matching method has best performance and the method based on players spatial distributions has worst performance. The most important information of a formation is its spatial layout of players, it is not surprising that these methods which take better advantage of this information will have better performance. The spatial pyramid models the spatial layout of all players in the formation of a play. The KLT trajectories analysis uses the spatial layout information of a certain type of players-wide receivers. The player spatial distribution method relies on the conclusion that the offensive team players tend to have smaller distribution variance than defensive team players, but the variance difference can be relatively small for a lot of football plays, and it can easily be affected by foreground noise(e.g. yard lines) compared to these two other methods. Thirdly, different classifiers(SVM, random forest and KNN) have similar accuracy for the spatial pyramid matching. Fourthly, both the accuracy of KLT tracks analysis and spatial pyramid matching are above 85% for either the predicted MOS or true MOS, which shows the effectiveness of these two methods.

Method	True MOS	Predicted MOS	Number of Plays
KLT Trajectories Analysis	0.87	0.86	1119
KLT Trajectories Analysis	0.88	0.86	436
Spatial Pyramid with SVM	0.93	0.90	436
Spatial Pyramid with Random Forest	0.92	0.91	436
Spatial Pyramid with KNN	0.90	0.90	436
Players Spatial Distribution	0.62	0.61	436

Table 4.1: Accuracy of different methods

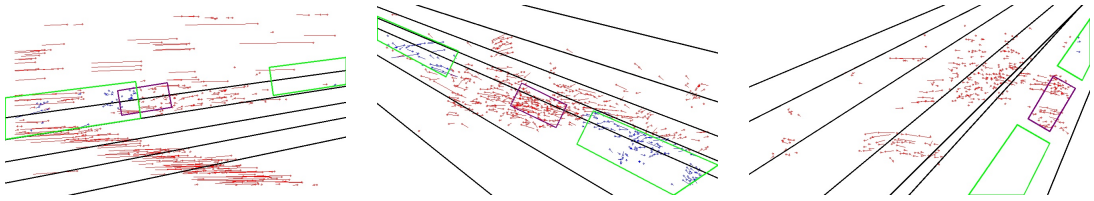

Figure 4.1: Three error cases of KLT trajectories analysis. For each image of a certain play, purple rectangle is the LOS bounding box, green rectangles are the ranges for WRs, red (blue) vectors are the KLT trajectories outside (inside) the range of the WRs. From left to right, the main error sources for these 3 cases are: wrong MOS detection and large camera motions, inexact low WR range detection, and LOS detection failure, respectively.

4.2.2 Error Sources Analysis

There are different error sources which lead to the failure cases of different methods. Wrong LOS/MOS detections, missing of WRs, and inaccurate WRs' estimation of areas(e.g. caused by the unnormalized distance between players) could lead to the failure of KLT tracks analysis, which can be seen from these three failure cases in Figure 4.1. The error sources of spatial pyramid matching could be: wrong MOS/LOS detection, noisy foreground, and missing players, which can be seen from Figure 4.2. Except the wrong LOS and MOS detection, the failure reasons for the method based on players spatial distribution could be: noisy foreground and relative small variance difference between the spatial players distribution of the offensive and defensive team. One failure case of the method based on players spatial distribution is shown in Figure 4.3. The offensive direction of the same play in Figure 4.3 can be correctly predicted by the spatial pyramid matching using the same foreground, which shows that spatial pyramid matching are more robust to the noisy foreground.

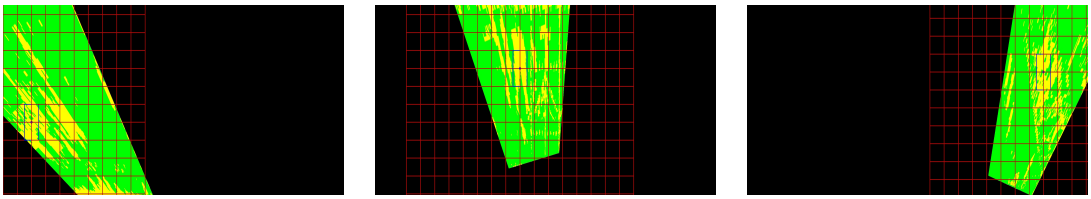

Figure 4.2: Three error cases of spatial pyramid matching. In each image, red rectangles represent the spatial pyramid cells, blue rectangle is the LOS bounding box, blue dot is the LOS center. From left to right, the main error sources for these 3 cases are: wrong LOS detection caused by both the field logo and wrong MOS detection, missing of players and false positive foreground generated by the field logo, and wrong MOS detection, respectively.

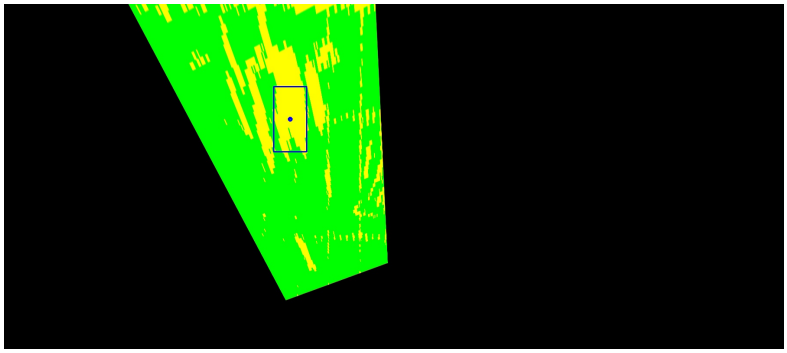

Figure 4.3: One error case of the players spatial distribution method, which is due to: i) the difference of the variance between the players spatial distribution of offensive and defensive team is relatively small, ii)false positive player foreground generated by field logos/boundaries/markings.

4.2.3 LOS Detection Evaluation

KLT trajectories analysis requires LOS detection in order to infer the region of WRs. LOS detection is also important to extract features from a play using spatial pyramid. Moreover, the players spatial distribution method also relies on the LOS detection to distinguish the offensive and defensive team. In Figure 4.4 and 4.5, the LOS center detection accuracy are measured with different offset pixel distance relative to the LOS center ground truth for 101 OD plays of a game for both KLT tracks analysis and spatial

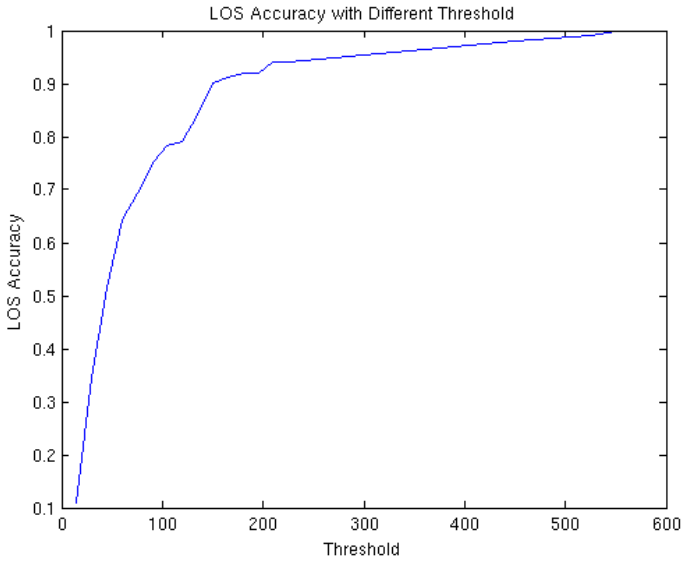

Figure 4.4: LOS center detection accuracy for KLT tracks analysis with different offset pixel distance.

pyramid matching, respectively. As can be seen, both methods achieve an above 90% accuracy when the offset distance reaches 150 pixels, which shows the effectiveness our LOS detection methods. The LOS center detection for spatial pyramid matching method performs better than KLT tracks analysis, for example, when the offset distance is 100 pixels, the accuracy of spatial pyramid matching achieves 100%, which outperforms the approximate 80% accuracy of KLT tracks analysis.

4.2.4 SVM with Different Kernels

To compare the performance of SVM with different kernels, we performed the leave-one-play-out test on these 436 plays with the registration annotation for SVM with different kernels, including spatial pyramid matching, linear, polynomial, radial basis function and sigmoid kernel. The results are shown in Table 4.2. The accuracy of spatial

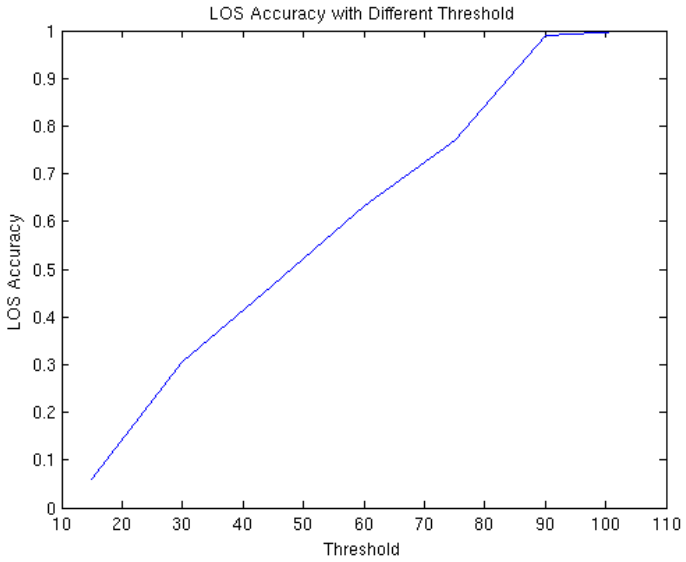

Figure 4.5: LOS center detection accuracy for spatial pyraimd matching with different offset pixel distance.

pyramid matching kernel is significantly better than these 4 other types of kernels, which shows that the spatial pyramid matching is able to capture the essential spatial structure information of formations and estimate the similarity between formations.

SVM Kernel	True MOS	Predicted MOS
Spatial Pyramid	0.93	0.90
Linear	0.58	0.55
Polynomial	0.55	0.55
Radial Basis Function	0.55	0.55
Sigmoid	0.55	0.55

Table 4.2: Leave-one-play-out accuracy of SVM with different kernels.

4.2.5 KNN

To see the effect of different K value, the leave-one-play-out test for KNN is performed on these 436 OD plays with registration annotations with different Ks, whose accuracy can be seen from Table 4.3. As can be seen from the result, smaller Ks(e.g. from 1 to 5) have better performance than bigger Ks. This is reasonable because the bigger the K is, the more dissimilar cases will be contained in the selected nearest neighbours.

K	True MOS	Predicted MOS
1	0.90	0.90
2	0.88	0.86
3	0.88	0.89
4	0.90	0.90
5	0.90	0.91
10	0.88	0.90
50	0.84	0.85
100	0.83	0.83
200	0.81	0.82

Table 4.3: Leave-one-out accuracy of of KNN with different K

4.3 Running Time Analysis of Different Methods

We have already discussed the performance of different methods in term of accuracy. Considering the real-world football video data-set usually has huge size, it is also important to show that all these methods have reasonable running time. Our code is implemented in C++ using OpenCV library and tested on a Red Hat Enterprise 64 bit, 3.20 GHZ machine. The generation of KLT trajectories for 45 video frames following the MOS takes $10-15$ seconds per play. The KLT trajectories analysis for offensive direction inference takes $1-2$ milliseconds per play. The players' foreground generation

takes approximately 1s per MOS frame. For the spatial pyramid matching method, it takes $1-2$ seconds to extract features from the foreground of the MOS frame of a play. The leave-one-play-out test of one play among 436 plays(including training on 435 plays and testing one play) takes approximately $0.5-1$ second for SVM(no matter the type of kernel), $10-15$ seconds for random forest and $1-10$ milliseconds for KNN. Finally, the running time of the players spatial distribution is $0.5-1$ second per play. Overall, all methods are fast enough for practical use. The main time consuming part for KLT trajectories analysis is the generation of KLT trajectories. The training takes most of the time for spatial pyramid method, but the training only needs to be done once in real application with a representative training set.

Chapter 5: Conclusion

We have described three different methods for offensive direction inference of American football videos. The KLT trajectories analysis method seeks to detect the wide receivers to predict the offensive direction. The spatial pyramid matching method estimates the spatial layout correspondence of the players' foreground between the two types of plays with different offensive directions to distinguish them. The player spatial distribution method infers the offensive direction based on the conclusion that the offensive formation tend to have smaller player spatial distribution variance than the defensive formation. All these methods have been evaluated on a large data-set of real-world football videos with wide variations of video conditions and formations. The evaluation shows that both our KLT trajectories analysis and spatial pyramid matching methods achieve promising results(above 85% accuracy), which is close to being usable for real-world applications.

The most essential information of a formation is its spatial layout of players, which can be shown by the convincing performance of the spatial pyramid matching. Future work such as offensive formation recognition can also resort to the spatial layout of players. But we probably need more complex models to capture the spatial structure of different types of offensive formations in more details, because the difference between different offensive formations will not be as obvious as the difference between offensive and defensive formations.

Bibliography

[1] Anubhav Agarwal, CV Jawahar, and PJ Narayanan. A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.

[2] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[3] Indriyati Atmosukarto, Bernard Ghanem, Shaunak Ahuja, Karthik Muthuswamy, and Narendra Ahuja. Automatic recognition of offensive team formation in american football plays. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 991–998. IEEE, 2013.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[6] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[7] Sheng Chen, Zhongyuan Feng, Qingkai Lu, Behrooz Mahasseni, Trevor Fiez, Alan Fern, and Sinisa Todorovic. Play type recognition in real-world football video. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 652–659. IEEE, 2014.

[8] Xue-Wen Chen and Mei Liu. Prediction of protein–protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

[9] Danny Coomans and Désiré Luc Massart. Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136:15–27, 1982.

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[11] Yi Ding and Guoliang Fan. Camera view-based american football video analysis. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pages 317–322. IEEE, 2006.

[12] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[13] Charles Dubout and François Fleuret. Exact acceleration of linear object detectors. In *Computer Vision–ECCV 2012*, pages 301–311. Springer, 2012.

[14] Claudio Fanti, Lihi Zelnik-Manor, and Pietro Perona. Hybrid models for human motion recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 1166–1173. IEEE, 2005.

[15] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[16] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[17] A Ardeshir Goshtasby. *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.

[18] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[19] Robin Hess, Alan Fern, and Eric Mortensen. Mixture-of-parts pictorial structures for objects with variable part sets. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[20] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[21] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[22] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio. A biologically inspired system for action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. Ieee, 2007.

[23] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[24] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[25] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[26] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, et al. Spatial pyramid matching. *Object Categorization: Computer and Human Vision Perspectives*, 3:4, 2009.

[27] Ruonan Li and Rama Chellappa. Recognizing offensive strategies from football videos. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4585–4588. IEEE, 2010.

[28] Behrooz Mahasseni, Sheng Chen, Alan Fern, and Sinisa Todorovic. Detecting the moment of snap in real-world football videos. In *IAAI*, 2013.

[29] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001.

[30] Stephen O'Hara and Bruce A Draper. Scalable action recognition with a subspace forest. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1210–1217. IEEE, 2012.

[31] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[32] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[33] Behjat Siddiquie, Yaser Yacoob, and Larry S Davis. Recognizing plays in american football videos. Technical report, Technical report, University of Maryland, 2009.

[34] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014.

[35] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[36] Eran Swears and Anthony Hoogs. Learning and recognizing american football plays. *preservation*, 1(4):5–10, 2009.

[37] Eran Swears and Anthony Hoogs. Learning and recognizing complex multi-agent activities with applications to american football plays. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 409–416. IEEE, 2012.

[38] Jakob Verbeek and William Triggs. Scene segmentation with crfs learned from partially labeled images. 2007.

[39] Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features: the kernel recipe. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 257–264. IEEE, 2003.

[40] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 915–922. IEEE, 2013.

[41] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.