Detecting the Moment of Snap in Real-World Football Videos

Behrooz Mahasseni and **Sheng Chen** and **Alan Fern** and **Sinisa Todorovic**
School of Electrical Engineering and Computer Science
Oregon State University

Abstract

In recent years, there has been a great increase in the use of web services for the storage, annotation, and sharing of sports video by athletic teams. Most of these web services, however, do not provide enhanced functionalities to their users that would enable, e.g., faster access to certain video moments, or reduce manual labor in video annotation. One such web service specializes in American football videos, supporting over 13,000 high school and college teams. Its users often need to fast-forward the video to certain moments of snap when the corresponding plays of the football game start. To our knowledge, this paper describes the first effort toward automating this enhanced functionality. Under a very tight running-time budget, our approach reliably detects the start of a play in an arbitrary football video with minimal assumptions about the scene, viewpoint, video resolution and shot quality. We face many challenges that are rarely addressed by a typical computer vision system, such as, e.g., a wide range of camera viewing angles and distances, and poor resolution and lighting conditions. Extensive empirical evaluation shows that our approach is very close to being usable in a real-world setting.

1 Introduction

American football teams put many resources into the collection, annotation, and analysis of game video of both their own games and those of their opponents, for the purposes of game planning. In recent years, companies have begun offering web services to facilitate these video-related activities. Such web services currently do not perform any type of automated analysis of the game videos, but provide only basic functionalities to their users. This makes human computer interaction cumbersome, and requires a significant amount of human labor when using the web service. For example, cutting non-useful parts of the video (and thus saving the purchased storage space) has to be done manually. Also, accessing a certain video part involves time-consuming watching of irrelevant parts, before observing the desired moment. Therefore, there is a growing demand for automated analysis of football videos, which would enable enhanced functionalities.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Designing such a video analysis system, however, is highly non-trivial, and beyond the capabilities of off-the-shelf computer vision tools. The key challenge is a huge diversity of football videos, which the web services typically host. The videos vary widely in terms of camera viewing angles and distances, resolution and shot quality, and weather and lighting conditions. The videos are often taken by amateurs, and thus exhibit motion blur and jittery camera motions, which may not be correlated with the football play. All this requires relaxing the restrictive assumptions about viewpoints, scales, and video shot quality, commonly made in the computer vision literature.

This paper presents, to the best of our knowledge, the first computer vision system that is capable of addressing a large diversity of football videos. Given a raw football video, our approach is aimed at estimating the moment when a football play begins, also known as the *moment of snap*. Our approach has a number of applications, including automatic video cutting, initializing the start frame for viewing, and providing a seed frame for further automated analysis. Since we cannot make assumptions about the players' layout in the scene and video quality, our primary goal is achieving robustness in the face of the wide variability, while also maintaining a reasonable runtime. This is made feasible by our new representation of motion in a video, called Variable Threshold Image.

We conduct this study in close collaboration with one of the largest companies dealing with football video[1], having a client base of over 13,000 high school, college, and professional teams.

In what follows, we first describe our application problem. Next, we describe our approach. Finally, we provide a detailed evaluation and sensitivity analysis of our approach on a select set of 500 very diverse real-world videos. This empirical evaluation indicates that our current approach is close to being ready for use in upcoming product releases.

2 Background and Problem Statement

In this section, we first give an overview of the web service our work is targeted toward, and the characteristics of the

[1]This company has chosen to remain unnamed for competitive reasons, at this time, but this information can be provided to the program chairs, with proper disclosures.

football video that we will be dealing with. We then discuss some of the challenges involved with automated video analysis, and state the specific analysis problem addressed in this paper. Finally, we review related work.

Web Services for Football Video. The web-service company that we work with provides services to over 13,000 high school, college, and professional football teams. It provides the functionalities for uploading game video, which can then be manually annotated, and shared with other users. Typically, the teams will upload video of each of their own games, and also get access to opponent video via a secure video-trade feature.

Game video is, for the most part, captured with one or more panning, tilt, and zooming (PTZ) cameras. In most cases, one camera captures a sideline view from an elevated location along the sideline. The sideline view generally provides the best overall view of a game. Figures 1 and 5 show typical examples of sideline views. These are the views that our work will focus on.

American football video is shot and organized around the concept of football plays. Each game involves a sequence of plays, separated by short time intervals where no game action occurs, and the teams regroup. Before each play begins (with minor exceptions), the offensive and defensive teams line up facing one another at the line of scrimmage — the line where the ball is located at the time. The play starts when the ball is "snapped" (or passed) from a player called the center to a player called the quarterback, and both teams begin moving and executing their chosen strategies. Each play lasts from roughly 5 to 30 seconds, and ends under various conditions (scoring, halting forward progress, etc). The cameras are operated so that they begin recording a play sometime before the moment of snap (MOS), and end recording at the termination of each play. Thus, at the end of a game, a camera has a sequence of files, one for each play in the game. These files are then uploaded to the web-service for storage and manipulation via a user interface.

The recording of each play, however, does not generally begin at the exact MOS. Rather, in many cases, there is a significant amount of time that elapses between the start of a video and the MOS. This prefix of the video is not useful to viewers, costing them waiting time. It also wastes server space, costing the web-service company dollars. Thus, automated MOS estimation could save both of these costs. First, the play viewer could be initialized to start at the estimated MOS, or a small number of frames before the estimated MOS. Second, the pre-MOS prefix of a video could be cut in order to save server space. Thus, a solution to automated MOS estimation has an immediate and high product value.

Challenges. Automated analysis of football videos, hosted by the aforementioned web service, is challenging due to their enormous variability. In particular, the videos are shot by camera-persons of varying skill and style, on fields with different textures and markings, under different weather and lighting conditions, from different viewpoints, and cameras of varying quality. Further, the scenes around the field can vary significantly, ranging from crowds, to players on the bench, to construction equipment. Figures 1 and 5 show some examples the video variability encountered on the web service.

Moment of Snap Estimation. In light of the aforementioned video variability, we have worked with the company to identify an analysis problem that would both have immediate product value, while also appearing approachable in the near term. The problem that has resulted is to estimate the frame number where a play starts in a video. We refer to this problem as *moment of snap (MOS) estimation*, since each play starts with the snap of the ball. More precisely, our input for MOS estimation will be a video of a single football play, and the output will be a frame number. The quality of the output is based on how close the frame numbers are to the actual moment of snap. In addition, the runtime of the solution is very important, because any computational overhead will cost money in terms of server time, and possibly delays upon a first viewing.

Related Work. While the computer vision literature presents a number of approaches to analyzing football (and other team sports) videos, it is unlikely that they would be successful on our videos. This is, for the most part, due to the restrictive assumptions made by these approaches. For example, inferring player formations in a football video, presented in (Hess, Fern, and Mortensen 2007), could be used to identify the line of scrimmage, and thus facilitate MOS estimation. Similarly, tracking football players, presented in (Intille and Bobick 1995; Hess and Fern 2009), and the 3D registration of a visible part of the football field, presented in (Hess and Fern 2007), seem as useful approaches that could be directly employed in MOS estimation. However, all of these methods make the assumptions that the videos are taken under fairly uniform conditions — namely, on the same football field, and from the same camera viewpoint and zoom — and thus cannot be applied in our setting. In addition, the approaches presented in (Liu, Ma, and Zhang 2005; Ding and Fan 2006; L. and Sezan 2001) perform foreground-background estimation, yard-line detection, and camera motion estimation for the purposes of activity recognition. These approaches require high-quality videos, a fixed scale at which the players may appear in the video, and prior knowledge of the field model. Consequently, these approaches cannot be used for MOS estimation in our videos. Remarkably, the reported accuracies of the above approaches are often not high, despite their restrictive settings, indicating fundamental challenges.

3 Overview of Our MOS Estimation

Typically, there is relatively little movement on the football field before the snap, followed by substantial movement by the players after the snap. Therefore, searching for the video frame that has the maximum difference of some measure of movement in the video before and after the frame seems a good approach. However, as our results will demonstrate later, such an approach is not effective for a variety of reasons. First, common measures of movement in the video — such as, e.g., optical flow, Kanade-Lucas-Tomasi (KLT) point-feature tracker, or tangent distance — typically estimate pixel displacements from one frame to another. All these motion measures are directly affected by a particular camera zoom and viewpoint, because object motions in close-up views correspond to larger pixel displacements than those in zoomed-out views, and, similarly, objects moving perpendicular to the camera viewing angle correspond to larger pixel displacements than those in other views. Since we cannot make strong assumptions about the camera zoom and viewpoint, the aforementioned naive approach could easily confuse small pixel displacements with a pre-snap period when they actually correspond to very large player motions on the field. Second, the camera may pan and zoom arbitrarily, at any time, which registers as pixel displacements, even when no foreground objects (here, football players) are moving. Since we cannot assume any type of calibration information between the camera and field, which otherwise could be used to subtract camera motion, the above approach is likely to confuse large camera motions with MOS. Third, one could try to separate video foreground (i.e., players) from background, and conduct MOS estimation based on the displacements of foreground pixels. However, since we cannot make strong assumptions about video resolution, field markings, and background, it is very difficult to reliably detect and track players.

Given the above challenges, we developed an approach for MOS estimation that has two main stages. The first stage, *field boundary extraction*, computes for each frame in a video an approximate top and bottom boundary of the field. This information can be used to spatially focus later processing on parts of the video that most likely correspond to the actual playing field. The second stage, *active cell analysis*, computes a novel representation of the video based on the concept of active cells, called Variable Threshold Image (VTI). The VTI represents coarse changes in the motion profile of a video. The VTI is then used to estimate MOS in a way that is more resilient to the indicated challenges compared to the aforementioned naive approach. The next two sections describe each of these stages in further detail.

4 Stage 1: Field Boundary Extraction

We make the assumption that each video frame shows a sideline view of a part of the football field. This assumption is reasonable for the intended application. However, the exact location of the football field relative to the coordinates of each frame can vary substantially from one video to another. To focus processing on the field rather than other frame parts (e.g. crowd), we seek to efficiently and robustly extract approximate field boundaries in each frame.

More formally, given a frame, depicting a sideline view of some part of a football field, the frame can be viewed as consisting of three parts: 1) The top part above the playing field in image coordinates, which often contains the crowd, or football players on the sidelines; 2) The middle part, which contains the field; and 3) The bottom part below the field in image coordinates, which often contains the crowd, or players on the sidelines. Our goal is to identify two boundaries, the *top boundary* between the top and middle part, and the *bottom boundary* between the middle and bottom part, as illustrated in Figure 1. The frame area between these two boundaries will roughly correspond to the football field, and is where further processing will be focused. It is important to note that in some cases (e.g. close-up shots), the middle/field part will extend all the way to the top or bottom of the frame, and hence the top and/or bottom parts may not be present. Thus, our approach must handle such situations.

To compute the field boundaries, we draw upon a recent dynamic programming approach for computing "tiered labelings" in images (Felzenszwalb and Veksler 2010). The tiered labeling in our case is defined as follows. Let I be the image frame with n rows and m columns. A tiered labeling of I is a sequence of pairs $s_k = (i_k, j_k)$, one for every column, k, such that $0 \leq i_k \leq j_k \leq n-1$. Given such a labeling, the top boundary is defined by the sequence of i_k values across the columns, and the bottom boundary is defined by the sequence of j_k values across the columns. Our solution will favor continuous boundaries.

Our goal is to find a labeling, f, that minimizes an energy function, $E(f)$, which measures the goodness of f for the particular application. We specify $E(f)$ such that it becomes smaller for labelings which are more likely to be good field boundaries, as

$$E(f) = \sum_{k=0}^{m-1} U(s_k) + \sum_{k=0}^{m-2} H(s_k, s_{k+1}), \qquad (1)$$

where U encodes the local goodness of the pair s_k for column k, and H encodes the horizontal contiguity of the boundaries selected for consecutive columns k and $k+1$. The definitions of these two functions are the same as those used in (Felzenszwalb and Veksler 2010). $U(s_k)$ assigns a lower energy (lower is preferred) to values of s_k where the corresponding pixels are estimated to belong to the football field part of the frame. The coarse football field localization is conducted by a simple clustering of the pixel colors, and selecting the most dominant cluster to represent the field color. $H(s_k, s_{k+1})$ penalizes pairs s_k and s_{k+1} to a degree that increases as their corresponding boundaries differ in location and pixel values. This component helps smooth out the extracted boundaries, which could be arbitrarily jagged if only $U(s_k)$ were used to optimize labelings.

We use the standard dynamic programming to minimize $E(f)$. Note that this approach can return solutions where one or both of the boundaries are not visible by assigning the corresponding boundaries close to either row 0, or row $n-1$. In practice, since we just need a coarse boundary estimation, the tiered labeling is efficiently done every 10 columns, instead of every column. As shown in the experimental section, the algorithm runs very quickly on all frames, and is not a time bottleneck of the current system. Two results of the algorithm are shown in Figure 1.

5 Stage 2: Active Cell Analysis

This sections describes our novel representation of motion changes in a video as Variable Threshold Image. It is based on quantization of motion in a video, and robust accumulation of spatial and temporal statistics of motion changes.

Given approximate field boundaries from stage 1, finding the MOS amounts to identify a frame where there is little prior motion followed by much motion on the field. As a measure of motion, we use the popular Lucas-Kanade dense optimal flow, which estimates for each pixel in a video frame the magnitude and direction of its displacement in the next

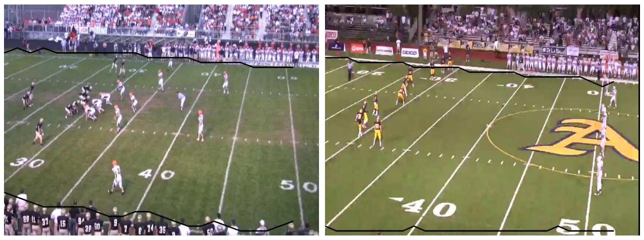

Figure 1: Results of our coarse field boundary detection. The red lines mark the extracted boundaries of the field.

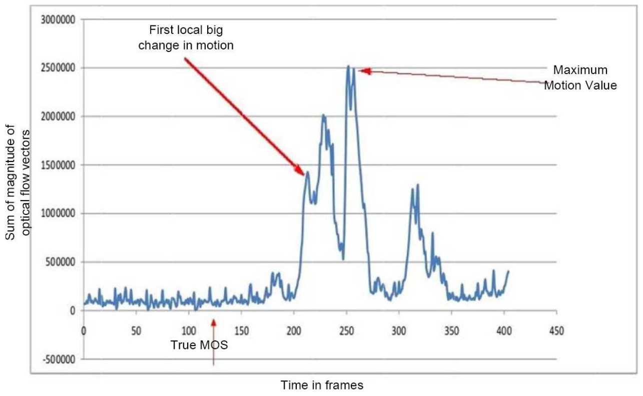

Figure 2: Sum of magnitudes of optical flow signal in time for an example video (the horizontal axis shows frames).

frame. While optical flow may be noisy, it can be computed efficiently compared to many other motion measures.

Our first attempt at MOS estimation based on optical flow, first, computes the sum of magnitudes (SOM) of optical flow vectors in the field portion of each frame. This provides a one-dimensional signal in time that roughly measures the motion across the video, as illustrated in Figure 2. Various statistics of this temporal signal can be used for selecting a particular frame as the estimated MOS, including: change points, local maximum, and various combinations and smoothed versions of these. However, empirically, these naive approaches frequently fail even in simple videos which have no camera motion. In the case of camera motion, the performance becomes much worse. As can be seen in Figure 2, the various statistics of the SOM of optical flow that one may consider do not always play out in practice.

This suggests that a more sophisticated analysis of changes of optical flow is needed for our problem. In response, we further investigate a quantization approach, which leads to the concept of an *active cell*. We divide each frame into $N \times N$ regular cells, where each cell within the field boundary is assigned a value equal to the SOM of the optimal flow vectors in that cell. Given a threshold, t, a cell is called active if its SOM value is above t. This provides a more robust estimate of whether there is motion in a particular area of the field versus more dispersed optimal flow. We then use the number of active cells in a frame as a measure of motion, rather than the overall SOM of a frames optical flow. This results in a new temporal signal of changes of active cell numbers per frame that we analyze. Specifically, we scan a window of length $2L$ across the video, and compute for each frame the difference between the number of active cells in the L following frames frames and the L previous frames. The frame that maximizes the difference is interpreted as the MOS.

The aforementioned difference depends on two input parameters — namely, the threshold t, and the window length $2L$. We experimented with a variety of choices and normalizations of t and L to identify their optimal values for MOS estimation. However, we were unable to find combinations that worked well across most videos. This suggests that an adaptive estimation of t would be more appropriate, for which we develop a new video representation called Variable Threshold Image.

Variable Threshold Image. For robust estimation of changes in the number of active cells across the video, we use a *variable threshold image (VTI)* as a representation of the motion in a video. We first discretize the nontrivial range of possible thresholds t into M evenly spaced values $\{t_1, \ldots, t_m, \ldots, t_M\}$. The VTI representation of a video with $n = 1, ..., N$ frames is then an $M \times N$ image, whose every pixel at location (m, n) encodes the difference in the total number of active cells detected at threshold $t = t_m$ in frames $\{n - L, n - L + 1, ..., n\}$ and frames $\{n + 1, n + 2, ..., n + L\}$. Figure 4 shows a contour plot of the VTI for a typical play that includes some periods of camera motion. The VTI provides a more complete view of the overall motion of the video than the aforementioned 1-D temporal signal (see Fig. 2). In particular, the local optima in the VTI tend to correspond to actual large changes in motion on the football field, as illustrated by labels of the time intervals of different events in the football play in Figure 4.

To understand why such local optima occur, consider an event that causes an increase in the amount of motion starting at frame n. For some threshold t_m, VTI(m, n) will be large. As we increase the threshold, $t_{m'} > t_m$, the difference in active cell numbers will tend to decrease, VTI$(m, n) >$ VTI(m', n), since for larger thresholds there will be overall fewer active cells (even with motion). Further, as we move away from frame n to frame n', where $n' < n$ or $n' > n$, and keep the threshold fixed at t_m, VTI$(m, n') <$ VTI(m, n), since for a frame n' we will have similar numbers of active cells before and after n. Thus, motion events will tend to register as peaks in the VTI.

MOS Classification. The VTI optima may correspond to several possible types of motion events on a football field, including the MOS, player motion before the MOS, and camera pans and zooms. As a result the problem of finding the MOS using the VTI amounts to selecting the correct local optima. To do this, we performed an exploratory analysis of various easily computable properties of local maxima across a variety of videos with different characteristics. Such properties included, absolute and normalized values of the maxima, area of the maxima, the absolute and normalized optical flow values before and after the maxima, etc. Given these features, we pursued a machine learning approach to classifying optima as the MOS using different classifiers, in-

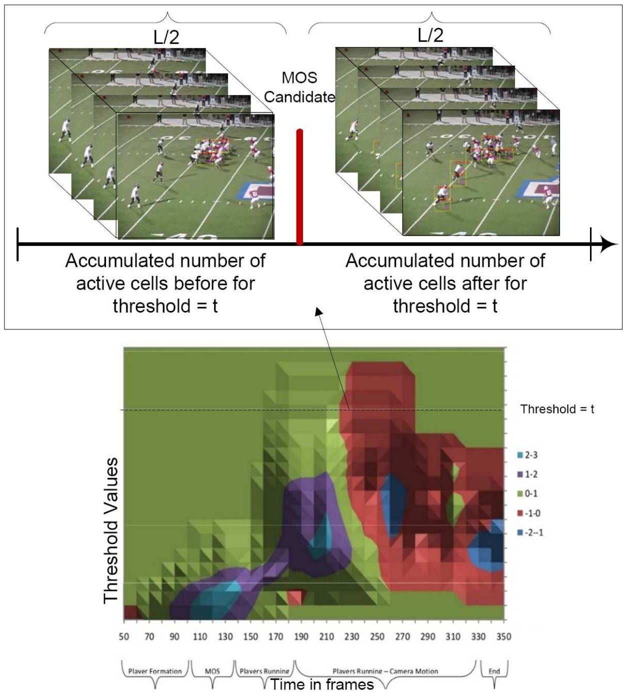

Figure 3: Contour plot of variable threshold image for a football play. The x-axis shows frame numbers, and the y-axis shows threshold t values of active cells.

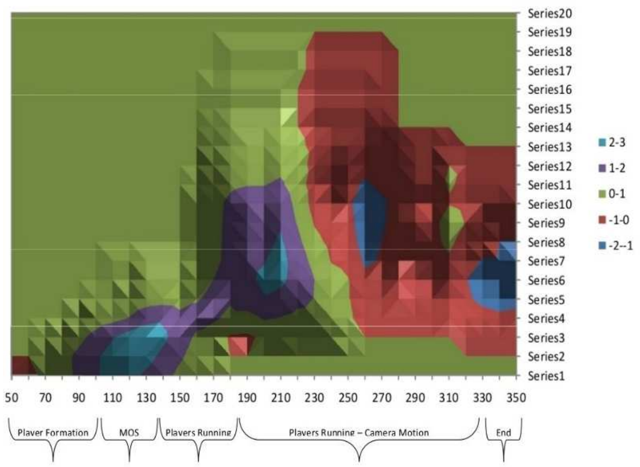

Figure 4: Contour plot of variable threshold image for a football play. The x-axis shows frame numbers, and the y-axis shows threshold t values of active cells.

Figure 5: Sample Videos - Video1 (top), Video2 (bottom)

cluding linear SVM, RBF-SVM, and decision trees. However, none of these classifiers gave satisfactory results, due to the mentioned huge variations in training and test video sets. Therefore, we resorted to our domain knowledge, and hand-coded the following classification rule for selecting an optimum of the VTI as our MOS estimate.

We first collect the top local optima that have a value within 50% of the best local optima. We find that this set of optima almost always contains the optimum corresponding to the true MOS. We then select the optimum from that set that has the minimum amount of raw optical flow occurring in the L frames before it. The intuition behind this rule is that it is generally the case that the true MOS produces a local optimal with the best value or very close to the best value. Further, the time before the MOS is generally fairly free of significant motion, even camera motion. This is because most players will be standing still and the camera is generally focused waiting for the action to begin. There are cases when the camera is moving or zooming during the MOS (generally considered bad camera work). But our rule often works in those cases as well.

6 Experiments

We evaluate our moment of snap detector on a set of 500 videos of high school football plays from the company's web-service database. Each video is hand-labeled by the frame number of the MOS for evaluation purposes. The videos are selected by the company to be representative of the video diversity they obtain from their customers, and is constrained only to include sideline view videos. The videos vary widely in viewpoint, number of players in the video, presence of a crowd, resolution, duration, scale, field color, and camera work. This makes the dataset very unconstrained. Figures 1 and 5 shows snapshots of sample videos.

Parameter Sensitivity and Selection. Our input parameters are the scanning window size L described in Section 5, and the "frame gap" used when computing optical flow. The frame gap of v indicates that optical flow is computed at frames that are multiples of v. Larger values of v lead to faster computations of, but less accurate optical flows. We begin by considering the impact of L on MOS accuracy. Table 1 shows quantitative results for different windows sizes using a fixed frame gap of 2. For each window size we show the percent of videos that have a predicted MOS within a specific number of frames of the true MOS. We see that the best results occur for values of L ranging from 100 to 150

Error(frames) \ W-Size (frames)	50	100	150	200
$[-5, +5]$	38	39	35	24
$[-15, +15]$	30	30	30	28
$[-30, +30]$	5	15	17	10
$\geq$ 1 second	27	16	18	38

Table 1: Percent of videos in different error ranges for different values of the window size L. $[-\delta, \delta]$ corresponds to videos where the predicted MOS is within δ frames of the true MOS. The final row is for videos whose predictions are greater than 30 frames (1 sec) away from the true MOS. $[-\delta_i, \delta_i]$ does not include videos in $[-\delta_j, \delta_j]$ where $j < i$.

Error(frames) \ Gap(frames)	2	3	5
$[-5, +5]$	39	32	28
$[-15, +15]$	30	34	31
$[-30, +30]$	15	15	17
$\geq$ 1 second	16	19	24

Table 2: Error when applying our algorithm with different gaps and window size = 100. Accuracy in [%]

frames. When using small windows, we are more susceptible to noise, while larger windows smooth out the signal too much for good localization. Based on these results we use a value of $L = 50$ for the remainder of our experiments.

Table 2 shows quantitative results for different values of the frame gap v when using $L = 100$. After discussions with the company, it was decided that the maximum runtime permissible by our approach was approximately 4x to 5x of real-time. Given this constraint, the minimum frame gap that we can consider is $v = 2$. From the table we see that indeed a gap of $v = 2$ provides the most accurate results, and thus we use this value for the remainder of the paper.

Comparison to Baselines. As described in Section 5, we considered a variety of baseline approaches early in our development that computed simple statistics of the raw optical flow changes in time. Here we compare two of the best baselines of this type: 1) *Max Change*, which measures the difference in total optical flow between successive frames, and returns the frame preceding the maximum difference; and 2) *First Big*, which selects the frame preceding the first "big" change in optical flow, where big is relative to the set of changes observed in the video. Note that the baselines only consider optical flow within the extracted field boundaries, which make them more comparable to our active-cell approach. Table 3 shows the results of two baselines, and our approach for $L = 100$ and $v = 2$. We see that the baselines do not perform very well, and commit a large percentage of errors over 1 second. Rather, our approach has a much smaller percentage (16%) of 1 second errors. A large fraction of the active cell results are extremely accurate, with 69% having an error less than 15 frames or 0.5 seconds. As we will show later, these levels of error appear to be at a level that can be useful for video initialization and cutting.

Error(frames) \ Method	Max Change	First Big	Ours
$[-5, +5]$	2	3	39
$[-15, +15]$	3	7	30
$[-30, +30]$	6	12	15
$\geq$ 1 second	89	78	16

Table 3: Comparison with baselines. Accuracy in [%]

Running Time. The average runtime of our code, implemented in C, per frame, used by each computational step is as follows: 1) Field boundary extraction 1ms, 2) Optimal flow calculation 105ms, and 3) Active cell analysis 49ms. The optical flow consumes about 2/3 of the total runtime.

Error Analysis. We carefully examined videos where our current approach makes errors of more than 1s. The errors can be grouped into two categories: 1) The MOS occurs at or very close to the first video frame, and 2) A local optimum corresponding to a non-MOS event has a significantly higher value than that of the MOS. The reason for the first case is obvious. Our method ignores the first and last $\frac{L}{2}$ frames of the video since the sliding window of length L is centered at each analyzed frame. The second error case is more complex, and is related to arguably poor camera work. In some videos, there are one or more extremely jerky camera movements. Those movement can lead to large local optima due to apparent movement of background objects on the field (e.g. numbers, lines, logos) and/or non-moving players. One way to avoid the second type of error is to explicitly estimate and subtract camera motion from the optical flow. However, existing approaches to camera motion estimation cannot deal will our video diversity.

Video Cutting Evaluation. An important application of our MOS estimator will be to cut unnecessary pre-MOS parts of the video. We say that the estimated cut point is a *bad cut* if it occurs after the MOS. To avoid bad cuts, the company plans to propose cut points not exactly at our estimated MOS, but rather at some number of frames Δ before our our MOS estimate. We considered three values of Δ and measured the percentage of bad cuts across our data set for each: 1) $\Delta = 0$: 63% bad cuts, 2) $\Delta = 30$: 11% bad cuts, and 3) $\Delta = 60$: 8% bad cuts. These results show that Δ need not be large to arrive at reasonably small bad cut rates. The majority of these remaining bad cuts are due to videos with very early moments of snap, which as we discussed above, our method does not properly handle, yet.

7 Road to Deployment

Considering the size and diversity of our dataset, the above results show that the current system can have utility in real software. The current plan is to begin integrating the MOS estimator into the highlight viewer and editor functionality provided by the company in 2013. The MOS detector will be used for smart initialization of video and safe cutting.

There is interest in improving our current approach, both in terms of runtime and accuracy/reliability. Regarding computation time, we will explore alternative optical flow calculations and video sampling strategies. We will also evaluate the speedups attainable via GPU implementations.

Regarding improving the accuracy and reliability, we are currently pursuing two directions. First, in terms of reliability, we are interested in providing the company not only a MOS estimate, but also a confidence associated with our estimate. When our system indicates high confidence the accuracy should almost always be high. Such a confidence estimate would be quite valuable to the company, since they could choose to only act on highly confident predictions.

Our second effort toward accuracy improvement is to address the two main failure modes observed in our experiments. First, to address the issue of camera motion, we are currently developing approaches for estimating the camera motion that are tailored to our data. In particular, we are developing estimation techniques based on tracking the lines on the football field. The other major error mode was for videos where the MOS occurs very close to the start. We plan to work on determining whether there was any significant player motion at the start of the video.

Acknowledgement

This research has been sponsored in part by NSF IIS 1018490.

References

Ding, Y., and Fan, G. 2006. Camera view-based american football video analysis. In *IEEE ISM*.

Felzenszwalb, P. F., and Veksler, O. 2010. Tiered scene labeling with dynamic programming. In *CVPR*.

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

Hess, R., and Fern, A. 2009. Discriminatively trained particle filters for complex multi-object tracking. In *CVPR*.

Hess, R.; Fern, A.; and Mortensen, E. 2007. Mixture-of-parts pictorial structures for objects with variable part sets. In *ICCV*.

Intille, S., and Bobick, A. 1995. Closed-world tracking. In *ICCV*.

L., B., and Sezan, M. I. 2001. Event detection and summarization in sports video. In *CBAIVL*.

Liu, T.-Y.; Ma, W.-Y.; and Zhang, H.-J. 2005. Effective feature extraction for play detection in american football video. In *MMM*.