

Unit Detection in American Football TV Broadcasts Using Average Energy of Audio Track

Mei-Ling Shyu, Guy Ravitz

*Department of Electrical & Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
{shyu, ravitz}@miami.edu*

Shu-Ching Chen

*Distributed Multimedia Information System
Laboratory
School of Computer Science
Florida International University
Miami, FL 33199, USA
chens@cs.fiu.edu*

Abstract

In this paper, we explore the domain of American Football TV broadcasting with respect to interesting event detection. Most of the existing methods achieve the goal of event detection in sports video using some type of shot or scene based approaches. Many problems arise when it comes to work in the shot or scene level in the American Football domain. The big amount of camera movement, such as tilting, zooming, panning, and the large number of different angles used by different broadcasting networks make it very difficult to work in the shot/scene level. We will show how in any of these two levels, over segmentation is almost inevitable. We also propose a novel concept that answers most of the problems of the domain we work on. This is the concept of "unit." A unit is defined as good workable data, which is not necessary to be very accurate. The main purpose of the unit detection process we present is to locate these potentially interesting segments in the video, and separate them from the rest of the redundant data.

1. Introduction

Sports have always captured the interest of many people. As computers being developed, many sports applications came to birth. The huge amount of data that is produced by digitizing sports videos demands a process of data filtration and reduction. The large number of sport TV broadcasts also creates a need among sports fans to have the ability of seeing interesting parts of all these broadcasts, instead of watching all of them in their entirety. These needs were answered by the applications such as video summarization and highlight/interesting events extraction. In the literature, much of the related work achieves these goals by using the concept of shot and scene detection, and then extracts the features in

either the shot or scene level. The detection and feature extraction are done by using either uni- or multi- feature extraction models.

In [5], Ekin and Tekalp introduced an application which used shot type classification to detect play and break segments in basketball games, and goal detection in soccer videos. The Dominant Color Region feature was used to detect shot boundaries. Next, the shots were classified, and slow motion segments were detected. The results were the potentially interesting segments from which they performed a goal detection and summarization for soccer videos, and play-break detection for basketball videos. In [10], the authors used semantic shot classification to detect events mainly in tennis videos. In their work, a fusion scheme of visual and auditory modalities was used. In their paper, the domain knowledge of the game of tennis was used to learn the rules for shot class identification. First, the low-level features were extracted, namely motion vector field, texture, and color. Then some mid-level features were created such as dominant object motion, camera motion patterns, and homogeneous regions. Next, a fusion of the valid mid level features was made at the shot level, and finally the conclusion was made using the decision rules classifier. In our earlier work [4], the detection of goal events in soccer game videos was achieved using a framework that consists of three stages, namely feature extraction, data cleaning, and data mining. During the feature extraction stage, the shot boundary and shot-level multidimensional features were extracted. Due to the fact that the ratio between goal event shots and the rest of the shots in soccer is very small, the data was cleaned to fix this ratio before it was passed to the data mining phase. The decision tree method was used to achieve hierarchical data mining. In this paper, we will discuss our motives behind working with American Football broadcast videos, and will show why the existing shot and scene based approaches are not suitable for the purpose of finding the

units which are potentially interesting in American Football videos.

This paper is organized as follow. In Section 2, we discuss the domain of American football videos, show why the existing shot/scene based approaches are not suitable to detect potentially interesting segments, and propose a solution to this problem. Section 3 presents our proposed algorithm, and explains how it works. This will be followed by a discussion of our experiments and results in Section 4. We will conclude this paper with a conclusion, and a short discussion of our future work in section 5.

2. Why is the Unit Needed?

We have divided this section to three subsections. In the first subsection, we will discuss the American Football broadcast domain. In the second subsection, we will explore some existing applications using the shot/scene detection approaches, and show why these shots/scenes are not suitable in the American Football domain. We will finally propose our solution to this issue in the third subsection.

2.1. American Football Broadcast Domain

According to CNN, Super Bowl 38, which was the most recent one, was watched, in full or partially, by 140 million people in America. This number did not include many more viewers around the world. According to the official web site of the National Football League (www.nfl.com), the total paid attendance of the 2003 regular season was of 16,913,584 spectators, and the average was of 66,328 spectators per game. Such statistics were only for the US. In the spring of 1991, NFL Europe was founded. This brought this great game closer to Europe and the rest of the world. As you can see, American football has become a very popular sport in the United States, as well as the rest of the world.

An American football game is officially sixty minutes long. It consists of four fifteen minute quarters. An American football broadcast on the other hand usually lasts about three to four hours. During this broadcast there are many breaks. Some of them are commercial breaks, and some of them are actual game breaks. In general, we can observe a sequence of events in American Football game broadcasts. The broadcast usually begins with a short discussion of the commentators regarding the upcoming game. At this time, the commentators would probably be on camera. As the game begins, a sequence of plays and breaks begin. As defined in [1], during a play segment, the ball is within the boundaries of the field and is in motion and so are the players. This is the part of

the game that most viewers find to be interesting. Also defined in [1], a break is a segment where the players are preparing for a play (in a huddle, or in the formation); the ball is at rest, either the players or the crowd celebrates a score, a penalty marker has been thrown to the field, and more. Other breaks could be actual breaks in the broadcast, i.e., those are usually commercial breaks. All these breaks are redundant to the average viewer. Most of the viewers are interested in the actual plays, and even more interested in the successful plays (highlights). To these people, highlight/interesting event extraction and game summaries are very appealing. According to our research of the literature, we found that American football was very rarely explored, when it came to multimedia analysis in the field of sports. In [4][5][9][10], the main sports in focus is soccer. In [5], their work on basketball videos was also presented. In [10], Xu et al. mentioned that other than tennis, their application was also capable of handling basketball and volley ball. Could it be a result of not realizing how popular American Football is, or could it be due to the difficulties the format of an American Football broadcast domain presents when we try to model it?

All the reasons mentioned above in this subsection led us believe that the American Football domain ought to be explored. In the next subsection, we will address the existing work in the area of shot/scene detection.

2.2. Why is the Existing Work Not Suitable in the American Football Broadcast Domain?

As mentioned earlier, a lot of the existing work use shot-level and/or scene-level feature extraction to locate the interesting segments in sports video. This approach has proven to be very successful when handling soccer or tennis videos. In this subsection, we will discuss the existing work that deal with shot and scene detection in the areas other than sports, and show why these methods are not suitable for highlight/interesting event extraction and game summaries in American football sports. Then in the next subsection, we will present our proposed new “unit” concept that we believe is more beneficial to use when working with American Football broadcast material.

A framework that detects scenes in Hollywood movies and TV shows was proposed in [7]. In that paper, the authors provided us with a definition of what a scene is. A scene is defined as one of the subdivisions of a play in which the setting is fixed, or when it presents continuous actions in one place [7]. Following this definition, the authors proposed a two pass framework to detect scenes in produced movies. The first step was to detect shots. This was done by calculating the color

histograms of consecutive frames, and measuring the distance between them. If the distance was lower than a specific threshold, the frames were considered as belonging to the same shot, and if not, they were considered as members of different shots. This was done under the notion that in produced movies, those frames from the same shot will have the same color characteristics, since the scenery in a shot usually does not change. The second step was to calculate the motion content and shot length. To detect the scene, a two pass scene boundary detection algorithm was used. The first pass (Pass One) tested color similarity between shots, and the second pass checked the dynamics of the scenes detected in Pass One. This was done due to the fact that many action scenes that have many shot changes in them were over-segmented in Pass One. In this pass, consecutive scenes with high dynamics (high motion) will be merged to one scene.

In [3], Alatan et al. proposed a framework to detect dialog scenes in movies using the audio visual information and Hidden Markov Models. Their proposed framework splits the material to audio and video. The video part is passed through a shot-boundary detection phase, and the audio is classified to be either music speech or silence. The result from the shot-boundary detection phase is passed through two processes, namely face/no-face detection, and location change detection. All these are used to create tokens that are fed into a Hidden Markov Model system and generate the result. The face detection was performed using skin color detection, and the location change is done using color histogram comparison of consecutive frames. The fusion of the results is used to conclude whether each shot detected is of dialog or not. If consecutive dialog shots are found, they are considered as a dialog scene. Two other applications that used scene detection were discussed in [2][6]. The scene detection was achieved in a way similarly to the rest of the work we have already discussed. [2] proposed a scene-based traffic modeling and queuing analysis of MPEG video sequences; while [6] presented a system that constructed a city video database based on automatic scene detection. Though the results of all these existing approaches were reported to be very good, unfortunately, they are all not suitable for our purpose of detecting the highlights/interesting events in American Football games.

For examples, in [5], Dominant Color was used to achieve shot boundary detection. If this approach is applied in the football domain, the over-segmenting problem may occur since in American Football games, one play segment can contain more than one shot, or a changing shot. The reason is that a play can start as a wide angle shot, and as the play progresses, it will turn into a more zoomed shot. This will result in over-

segmenting our potentially interesting segment, since the dominant color will eventually change before the segment we are interested in is over (as shown in Figure 1). Similarly, we will run into the same problem if other methods, like histogram comparison [4][6][7][8], is used for shot detection. Due to zooming and tilting, the intraframe color histogram can change before a play is over. For example let us consider a play that begins in an area of the field, where the grass is all green, and then passes by mid field, or the end-zone. Both the mid field and the end-zone, will have different color characteristics, this will again over-segment the play segment. Due to the fact that at the shot level, we will end up with over-segmenting potentially interesting segments, it is impossible for us to use the shot-based approach [4][5][10]. Therefore, the next step is to check the possibility of working in the scene level.










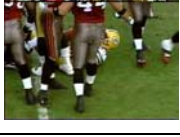

		
17	128	180
		
215	258	297
		
379	447	509
		
578	610	

Figure 1. Key frame sequence of a unit

If we go back to the definition provided in [7], a scene will be detected when there are changes in the location, environment, background, etc. However, in the football domain, there is no change of location. All the action is done on the football field. It will be very difficult if not impossible to detect scene boundaries when there is no location change. Most of the scene detection algorithms find the scene boundary by looking for abrupt changes. In the American Football Domain, we

encounter abrupt changes only in the cases of commercial breaks, crowd shots, zoom in, etc., but that will not enable us to separate potentially interesting segments from the rest of the video.

To better illustrate the problems of working in the scene and/or shot level in the American Football domain, a sequence of frames which are a part of a play segment from one of the games we digitized is given in Figure 1. These are key frames that represent this segment. The unit in which the potentially interesting event we are interested starts at Frame 128, and ends at Frame 578. From the previous experience and by observing this sequence of frames, it can be shown why the shot or scene level detection cannot serve the purpose of extracting highlights or interesting events. First, we can observe that we have 5 shot changes in this sequence. (Frames 128–180, 297–379, 379–447, 447–509, and 509–578). If working at the shot level, we would have probably over segment this play segment to at least 4 different segments. At the scene level, we still have a few problems. There is a scenery change between Frames 258 and 297 (the crowd disappears), and an abrupt change between Frames 447 and 509 (much less grass in the frame). Therefore in the scene level, we are still expecting over segmentation of at least two segments. Our algorithm actually detects a unit that starts at Frame 17, and ends at Frame 610. As mentioned earlier, we are not interested in the accurate segments. As long as the unit contains the entire play segment or more of it, our goal is achieved. When the unit only contains an incomplete part of the play, we consider that unit inappropriate. We will spend more time discussing this issue when we present our experiments and results in Section 4. These problems that we have just mentioned exist due to the fact that in the American Football domain, there is a lot of camera movement, namely tilting, zooming, and panning. On top of that, many different camera angles are used, by different networks that broadcast the different games. Many of these problems could be solved by utilizing our proposed unit concept (to be discussed in the next subsection).

2.3. The Unit

Our main purpose was to find a proper method to locate the potentially interesting event segments in an American Football broadcast, and separate them from the rest of the broadcast. This is considered as a preprocessing stage to a system that will be later implemented, and that will eventually be able to extract highlights from American Football TV broadcasts. Being a pre processing stage, it has 2 main purposes:

- Reduce the amount of data by ridding off the non-relevant data, meaning any type of breaks mentioned in Section 2.2.
- The result should be the workable data which we can use to find highlights/interesting events.

Both of these purposes can be answered by our proposed unit detection process. We define our unit to be a segment of consecutive frames. Such a unit can consist of one shot or several shots that represent a segment of a play potentially interesting. Unlike our work in [1], accurate segments are not required in our proposed unit detection process, which means that the exact beginning frame and ending frame of each play are not very critical. Actually, the focus is to keep the data from the breaks before the play begins and a little after it ends so that the nature of each unit can be learned. In addition, since this is considered as a preprocessing stage, the least computation required is more desirable. Due to these reasons, in our current approach, the audio feature namely *average energy* is used for unit detection, which serves our purpose. The motivations for using this feature are: (1) during interesting events, the crowd presence in the audio track is relatively high, which results in high audio energy; and (2) *average energy* is fairly simple to calculate, and does not require much computation. Figure 2 gives a plot of the samples' amplitude of a part of a clip's audio track. By observing this plot, it is fairly easy to see that a play segment should have a higher average energy than the break segments.

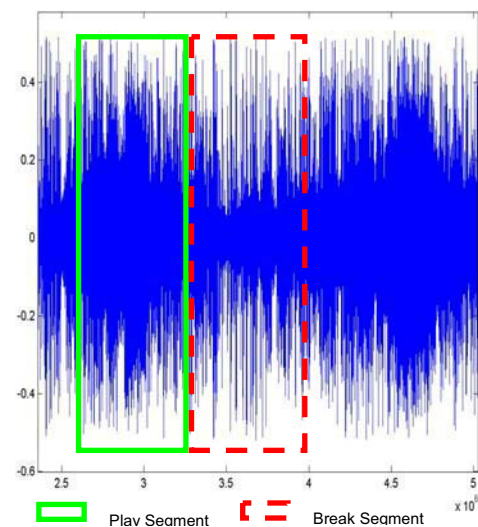


Figure 2. Plot of the amplitude vs. samples of a part of an audio track of a clip

The unit detection process produces a solution to all the problems we have previously addressed, and also

provides us with a way to reduce the amount of data being processed that is one of the desired purposes when it comes to data preprocessing and cleaning. In the next section, we will get into a detailed discussion of our algorithm, and how it is used to produce our unit.

3. Our Proposed Unit Detection Algorithm

Our proposed unit detection algorithm uses the following formula (shown in Figure 3) to calculate the average energy of the audio track.

$$E = \sum_{n=1}^{L-1} \frac{x^2(n)}{L}$$

where $x(n)$ is the value of the n^{th} sample of x , and L is the total number of samples that the average energy is calculated.

Figure 3. Average energy formula

Before any calculation is being made, each audio clip is first normalized using the “Normalize Mean to Zero” method. The reason for normalization is that the clips we have experimented with were recorded from different sources, different TV networks, and using different media, and hence different clips may have different volumes (amplitudes of the samples). By normalization, their levels can be adjusted to be relatively similar to each other. In our early experimentations, we have noticed that a difference in the level of the audio clips harms the performance of our algorithm. The main reason behind it was that the most important threshold in our algorithm was an amplitude threshold, and when each clip had a different level, this threshold would not be calculated correctly. The normalization process is simply achieved by calculating the mean of an audio clip, and adjusting the levels of all the samples in order to bring the mean closer to zero. As we will mention later, we work with stereo audio files. This means that each audio file has two channels, namely left and right channel. Hence during normalization, we calculate the mean for each channel and adjust each channel’s levels appropriately. This means we normalize the left channel mean to zero by adjusting the values of the left channel’s samples using the left channel mean value, and the right channel’s mean to zero by adjusting the right channel samples values using the right channel mean value.

When extracting the features from the audio tracks, it is common to work in frames/bins, as opposed to work on the entire data at once, to get more accurate results. Many different bin sizes have been proposed in the literature.

Based on our experimental experience, the bin size of 50 milliseconds is selected, which will be discussed more in the next section. In addition, the value of L (in Figure 3) will be the number of samples equivalent to 50 ms. The average energy for each 50 ms bin is calculated and all these results are stored in a vector called Average Energy Vector (AEV). This AEV vector represents our entire clip.

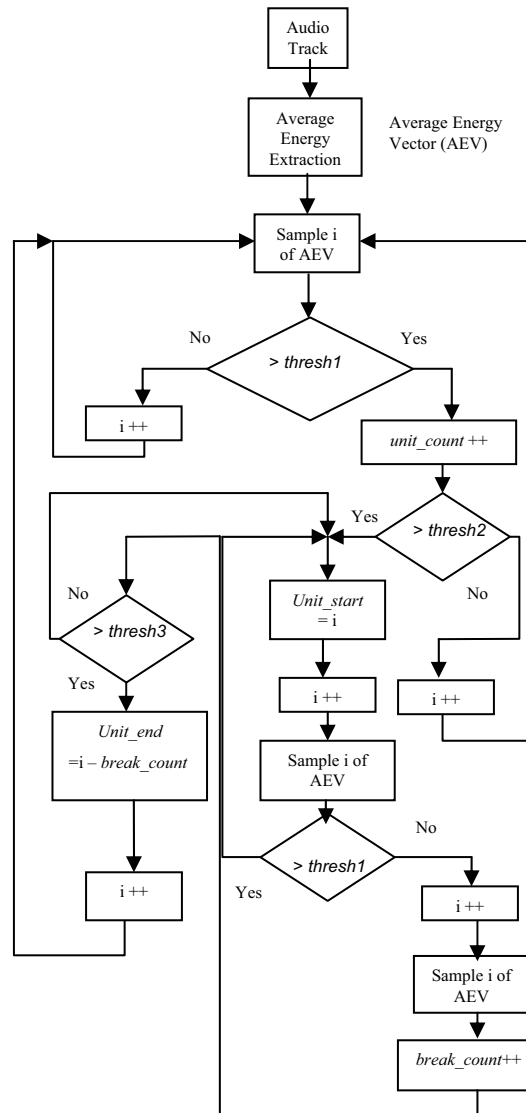


Figure 4. Flowchart of the proposed unit detection algorithm

The next step is to use this data to conclude which segments are the desired units, and which are redundant. For this purpose, three different thresholds are defined, where $thresh1$ is an amplitude threshold and $thresh2$ and $thresh3$ are duration thresholds. The value of $thresh1$ is

derived from the data (as shown in Equation 1) and the other two are determined via domain knowledge. The flowchart of our proposed unit detection algorithm is presented in Figure 4.

$$thresh1 = 0.53 * median(AEV) \quad (1)$$

As shown in Figure 4, each bin is first checked. If its average energy is more than *thresh1*, it would count as a potential member of a unit by incrementing *unit_count* by one. If not, the *unit_count* will be zero. If *unit_count* becomes greater than *thresh2*, we conclude that we have found a unit. The next step is to find out when the unit ends. In order to find the end of the unit, we continue to check the bin values of AEV. As long as a bin value is greater than *thresh1*, it will be considered as a member of the current unit by incrementing *unit_count* by one. If the next bin's value is less than *thresh1*, then a new count called *break_count* is started. This new count is incremented as long as the bin value is less than *thresh1*, and will be zero if the next bin's value is greater than *thresh1*. When *break_count* reaches the value of *thresh3*, we calculate the end of the unit by subtracting the value of *break_count* from the value of the current index (the current location in AEV). This algorithm continues until it reaches the end of AEV, which represents the end of the clip. The variables "*Unit_start*" and "*Unit_end*" are the beginning and end locations of each unit detected.

Now, let us take a quick look at how the threshold values being determined. The first threshold, *thresh1*, is an amplitude test (given in Equation 1). Any bin's value that is greater than this value is considered as a potential member of a unit. We will only consider the current sample to be a part of a unit if it is a part of *thresh2* or bigger consecutive samples that are greater than *thresh1*. Hence it can be seen that *thresh2* is a duration test, and its value is determined using domain knowledge. That is, an important segment (play) cannot be shorter than 5 seconds. The third threshold, *thresh3*, is used to find the end of the unit, and its value is determined using domain knowledge as well. This threshold exists to avoid over-segmenting a unit. We have observed that if a sequence is a unit, it can have up to 500 ms where the value of the bins is less than *thresh1* at any point, and still be considered as a member of the unit. If the value of *break_count* exceeds *thresh3*, then the end of a unit is found.

4. Experimentations and Results

In order to test our proposed unit detection algorithm, we recorded about 120 minutes of three different American Football games that were broadcasted in three

different major networks and commentated by three different teams of commentators. These games were recorded using a Samsung hi-fi stereo head VCR, off an analog cables network. These videos were later digitized using an ATI All-In Wonder 9600 XT video card, which was installed in a Pentium4 computer with Windows 2000. The video was divided into several clips. Each clip was between five to ten minutes long. The video was digitized to an MPEG-1 format. VirtualDub was used to separate the audio files from the video files. Each audio file was sampled using a sample rate of 44.1 KHz, 16 bits, and in stereo mode. The output was the stereo audio files of the different clips we have collected. These files were then processed using Matlab.

The first part of the experiments was to test the algorithm with various bin sizes of 20, 30, 40, 50, 100 ms, and to select the best performance from these bins, which is the bins of 50ms. We then made some experiments and observations to determine the two duration thresholds. As mentioned earlier, we concluded from these set of experimentations that a unit cannot be shorter than 5 seconds, and that within a unit we can have segments with amplitude less than *thresh1* and at most 500 ms long. After determining these threshold values, we tested the performance of the proposed algorithm in both the number of correctly detected units in a clip, and how the beginning and ending of each segment were detected. It needs to be noted that, as we discussed earlier, in this study, we are not focusing on accuracy when it comes to the detection of the beginning and ending of the segments. On the other hand, we are more concerned about the ability of the algorithm to extract potentially interesting units.

All the calculations were done in the sample domain. Since we sampled at 44.1 KHz, which meant that we had 44,100 samples per channel per second. In other words, 44 samples represent one millisecond. Hence, each bin (50 ms) was represented by 2,200 samples. In order to verify our results, we had to synchronize the units with the video clips to view the clips and check the validity of the results. The beginning or ending of the units are in terms of the indices of AEV, where each index represents 50 ms. In order to locate the appropriate location in the video that a specific bin is related to, we simply calculate the time in seconds it represents. This was done by taking the value we got (e.g., the beginning of a unit) and multiplying it by fifty. This gave us the answer in milliseconds then it is divided by one thousand, which gives us the location of this bin in seconds. Each unit's beginning and ending points are compared to a pre-list that has been prepared manually, by watching the different clips and recording the times of the play segments of each clip.

The experimental results are shown in Table 1. We had tested the clips from 3 different games, namely mia_jax, bucs_gb, and ne_car. We used two quarters from mia_jax and ne_car, and one quarter from bucs_gb. Each quarter was separated to clips, each between 5 and 10 minutes long. Each clip named units_x where the x stands for the chronological order of each clip, within its game and quarter. During the test of each clip, we have recorded the starting and ending location of each segment within the clip. We then compared the results to the pre-list. Success was recorded each time a segment's starting location was either less than or equal to the pre-recorded beginning location, and it's ending location was equal to or larger than the pre-recorded ending location. If a segment was concluded to be shorter, then it was considered an unsuccessful result. When a segment started too late or ended too early, we tolerated it only if it was off by at most 1 second at the beginning and/or at most 1 second at the end. As a reminder, we are interested in those potentially interesting events. In American football, there are many types of interesting events, for examples, a touchdown, a field goal, an interception, to name a few. When a segment that was recorded as a play but not detected as a unit, we still considered it a success if the content was of a non-interesting event. Out of the 122 play segments, 26 were not detected as units (missed segments). This was mainly in the cases of Kick-offs or punts, where the crowd does not get excited until the middle of the play. Among the 26 missed segments, 3 of them were detected as units but not the complete ones.

This is mostly due to the fact that the crowd did not get excited until the middle of the play, or lost excitement before the end of the play. We also had 20 cases that the algorithm did not detect as units, but since they were all play segments that contained data that is not potentially interesting, those units were ignored. They were not counted as play segments, and also not counted as missed segments, since we considered the fact that they were missed as a success. After all, one of our goals is to eliminate the non-interesting material. Finally, we observed 6 cases where the segments were declared as units but actually they were not (mis-detected units), which occurred due to the sound effects or short promos inserted to the broadcast. We are very confident that the numbers of missed segments and mis-detected units can be reduced when we introduce visual features to our work. As can be seen from Table 1, our proposed unit detection algorithm achieves **94%** in precision and **79%** in recall. It is worth mentioning that our proposed algorithm is promising since it achieves satisfactory performance even by using data with poor quality. The poor quality is due to the use of a low level home VCR (using analog VHS tapes), the loss of generation during video capturing process, and another loss during the process of separating the audio files from the video files. We believe that the solutions to that problem are possible by using other feature(s), either from audio or from other modality, and improve the quality of the audio and video data. All these improvements are our on-going research directions.

Table 1. Experimental results in precision and recall

Game	Quarter	Clip	Total play segments	Detected units	Missed segments	Mis-Detected units	Precision	Recall
jax_mia	1	units_1	7	6	1	0	1	0.86
jax_mia	1	units_2	7	5	2	0	1	0.71
jax_mia	1	units_3	7	5	2	0	1	0.71
jax_mia	1	units_4	7	6	1	0	1	0.86
jax_mia	2	units_1	6	6	0	0	1	1
jax_mia	2	units_2	7	5	2	0	1	0.71
jax_mia	2	units_3	7	4	3	2	0.67	0.57
jax_mia	2	units_4	6	5	1	0	1	0.83
jax_mia	2	units_5	8	5	3	0	1	0.63
bucs_gb	1	units_1	6	5	1	0	1	0.83
bucs_gb	1	units_2	7	6	1	2	0.75	0.86
ne_car	3	units_1	5	4	1	0	1	0.80
ne_car	3	units_3	6	5	1	0	1	0.83
ne_car	3	units_4	4	3	1	2	0.6	0.75
ne_car	3	units_5	6	5	1	0	1	0.83
ne_car	4	units_1	4	3	1	0	1	0.75
ne_car	4	units_2	6	4	2	0	1	0.67
ne_car	4	units_3	7	6	1	0	1	0.86
ne_car	4	units_4	5	4	1	0	1	0.8
ne_car	4	units_5	4	4	0	0	1	1
Overall			122	96	26	6	0.94	0.79

5. Conclusions

In this paper, a new “unit” concept that provides a solution to many problems that arise in the American football domain is presented. We define the units as the segments of the video which are potentially interesting. The process of unit extraction also acts as a data preprocessing and data cleaning process. We have shown that the proposed unit detection algorithm achieves satisfactory performance under the use of data with poor quality. We believe that this result can be improved. Our main purpose in this study was to introduce this new concept of the unit. This proposed “unit” concept helps us get around the problems that the shot-based and scene-based processing introduce when working in the American football domain. In the later stages of our system that will be soon implemented, the features will be extracted in the unit level, and using a data mining technique, we will discover which of these units are highlights, and also what types of highlights they are. We also plan to utilize more audio features and visual features to improve the ‘unit’ detection process.

6. Acknowledgement

For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Shu-Ching Chen, this research was supported in part by NSF EIA-0220562 and NSF HRD-0317692.

References

- [1] Abdel-Mottaleb, M. and Ravitz, G., “Detection of Plays and Breaks in Football Games Using Audiovisual Features and HMM,” *Proceedings of the 9th International Conference on Distributed Multimedia Systems*, September 24-26, 2003, Miami, Florida, USA.
- [2] Akrivas, G., Doulamis, N. D., Doulamis, A. D., and Kolias, S. D., “Scene Detection Methods for MPEG – Encoded Video Signals,” *IEEE Proceedings of 10th Mediterranean Electrotechnical Conference*, vol. 2, pp. 667-680, 2000.
- [3] Alatan, A. A., Akansu, A. N., and Wolf, W., “Multi-Modal Dialog Scene Detection Using Hidden Markov Models for Content-Based Multimedia Indexing,” *Multimedia Tools and Applications*, Vol. 14, pp. 137-151, 2001.
- [4] Chen, S.-C., Shyu, M.-L., Chen, M., and Zhang, C., “A Decision Tree-based Multimodal Data Mining Framework for Soccer Goal Detection,” *IEEE International Conference on Multimedia and Expo (ICME 2004)*, June 27-June 30, 2004, Taipei, Taiwan, R.O.C.
- [5] Ekin, A. and Tekalp, M., “Shot Type Classification by Dominant Color for Sports Video Segmentation and Summarization,” *IEEE Proceedings of International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Vol. III, pp. 173-176, 2003.
- [6] Jin, H., Yoshitomo, Y., and Sakauchi, M., “Construction of City Video Database Based on Automatic Scene Detection and Recognition,” *IEEE proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 963-968, Venice, Italy, September 1999.
- [7] Rasheed, Z. and Shah, M., “Scene Detection in Hollywood Movies and TV Shows,” *IEEE Proceedings of the computer Society Conference on Computer Vision and Pattern Recognition (CVPR’03)*, 2003.
- [8] Sundaram, H. and Chang, S.-F., “Computable Scenes and Structures in Films,” *IEEE Transactions on Multimedia*, Vol. 4, No. 4, pp. 482-491, December 2002.
- [9] Xie, L., Chang, S.-F., Divakaran, A., and Sun, H., “Structure Analysis of Soccer Video with Hidden Markov Models,” *Proceedings of International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Orlando, FL, 2002.
- [10] Xu, M., Duan, L.-Y., Xu, C.-S., and Tian, Q., “A Fusion of Visual And Auditory Modalities for Event Detection in Sports Video,” *IEEE Proceedings of International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Vol. III, pp. 189-192, 2003.