Automatically Extracting Highlights for TV Baseball Programs

Yong Rui, Anoop Gupta, and Alex Acero
Microsoft Research
One Microsoft Way, Redmond, WA 98052
{yongrui, anoop, alexac}@microsoft.com

ABSTRACT

In today's fast-paced world, while the number of channels of television programming available is increasing rapidly, the time available to watch them remains the same or is decreasing. Users desire the capability to watch the programs time-shifted (on-demand) and/or to watch just the highlights to save time. In this paper we explore how to provide for the latter capability, that is the ability to extract highlights automatically, so that viewing time can be reduced**.**

We focus on the sport of baseball as our initial target---it is a very popular sport, the whole game is quite long, and the exciting portions are few. We focus on detecting highlights using audio-track features alone without relying on expensive-to-compute video-track features. We use a combination of generic sports features and baseball-specific features to obtain our results, but believe that many other sports offer the same opportunity and that the techniques presented here will apply to those sports. We present details on relative performance of various learning algorithms, and a probabilistic framework for combining multiple sources of information. We present results comparing output of our algorithms against human-selected highlights for a diverse collection of baseball games with very encouraging results.

Keywords

Highlights, summarization, television, video, audio, baseball.

1. INTRODUCTION

Internet video streaming and set-top devices like WebTV [1], ReplayTV [2], and TiVo [3] are defining a new platform for *interactive* video playback. With videos being in digital form, either stored on local hard disks or streamed from the Internet, many sophisticated TV-viewing experiences can be supported. It has become possible to "pause" a live broadcast program while you answer the doorbell and continue from where you left off. The fact that video is stored on the hard disk (instead of tape) also allows for instant random access to the program content. This allows for rich browsing behavior by users based on additional meta-data associated with the video. For example, indices into TV news programs can permit users to focus only on subset of stories that are of interest to them, thus saving time. Similarly, meta-data indicating action-segments in a sports program can permit viewers to skip the less interesting portions of the game.

The value of such meta-data was explored in a recent study by Li *et. al*., where viewers were provided with metadata (manually generated) and instant random access for a wide variety of video content [4]. The ability to browse video was found to be highly valuable by users, especially for news, sports, and informational videos (e.g., technical presentations, travel documentaries). In addition to saving time watching content, the users appreciated the feeling of being in control of what they watched.

We also note a key difference between two models on how highlights may be made available to viewers. In the traditional TV broadcast model, e.g., CNN sports highlights, when they show a 1-minute highlight of a game, the user has no choice to watch anything more or less. In the new model, with set-top boxes and hard disks, we can make the assumption that the whole 2-hour game is recorded on the local hard disk, and the highlights act only as a guide. If the user does not like a particular selected highlight they can simply skip it with a push of a button on their remote control, and similarly at the push of a button they can watch more details. This new model allows for greater chance of adoption of automatic techniques for highlight extraction, as errors of automation can be compensated by the end-user.

In this paper we explore techniques to automatically generate highlights for sports programs. In particular, we focus on the game of baseball as our initial target---it is a very popular sport, the whole game is quite long, often there are several games being played on the same day so viewer can't watch all of them, and the exciting portions per game are few. We focus on detecting highlights using audio-track features alone without relying on expensive-to-compute video-track features. This way highlight detection can even be done on the local set-top box (our target delivery vehicle) using the limited compute power available.

Our focus on audio-only forces us to address the challenge of dealing with an extremely complex audio track. The track consists of announcer speech, mixed with crowd noise, mixed with remote traffic and music noises, and automatic gain control changing audio levels. To combat this, we develop robust speech endpoint detection techniques in noisy environment and we successfully apply support vector machines to excited speech classification. We use a combination of generic sports features and baseball-specific features to obtain our results, but believe that many other sports offer the same opportunity. For example, we use bat-and-ball impact detection to adjust likelihood of a highlight segment, and the same technology can also be used for other sports like golf. We present details on relative performance of various learning algorithms, and a probabilistic framework for combining multiple sources of information. The probabilistic framework allows us to avoid *ad hoc* heuristics and loss of information at intermediate stages of the algorithm due to premature thresholding.

We present results comparing output of our algorithms against human-selected highlights for a diverse collection of baseball games. The training for our system was done on a half-hour segment of one game, but we test against several totally distinct games covering over 7 hours of play. The results are very encouraging: when our algorithm is asked to generate the same number of highlight segments as marked by human subject, on average, 75% of these are the same as that marked by the human.

The rest of the paper is organized as follows. Section 2 discusses related work from both technology perspectives and video domains. In Section 3, we first examine the advantages and disadvantages of the information sources that we can utilize to perform baseball highlights extraction and then discuss the audio features that will be used in this paper. In Section 4, we present both the algorithm flowchart and the algorithm details that include noisy environment speech endpoint detection, excited speech classification, baseball hit detection and probabilistic fusion. Section 5 presents detailed descriptions of the test set, evaluation framework, experimental results, and observations. Conclusions and future work are presented in Section 6.

2. RELATED WORK

Video-content segmentation and highlight extraction has been an active research area in the past few years [5]. More recently, leading international standard organizations (e.g., MPEG of ISO/IEC [6] and ATVEF [7]) have also started working actively on frameworks for organizing and storing such metadata. Below we focus primarily on technologies used and the types of content addressed by such systems and organizations.

There are primarily three sources of information used by most video segmentation and highlight detection systems. These are analysis of video track, analysis of audio track, and use of close-caption information accompanying some of the programs. Within each of these, the features used to segment the video may be of a general nature (e.g., shot boundaries) or quite domain specific (e.g., knowledge of fact that a news channel segments stories by a triple hash mark "###" in the close caption channel).

When analyzing the video track, a first step is to segment raw video into "shots". Many shot boundary detection techniques have been developed during the past decade. These include pixel-based, histogram-based, feature-based and compressed-domain techniques [8]. However, video shots have low semantic content. To address real-world need, researchers have developed techniques to parse videos at a higher semantic level. In [5], Zhang *et. al*. present techniques to categorize news video into anchorperson shots and news shots and further construct a higher-level video structure based on news items. In [9], Wactlar *et. al*. use face detection to select the frame to present to the user as representative of each shot. In [10], McGee and Dimitrova developed a technique to automatically pick out TV commercials from the rest of the programs based on shot change rate, occurrence of black frames and occurrence of text regions. This allows users to quickly skip through commercials. In [11], Yeung *et. al*. developed scene-transition graphs to illustrate the scene flow of movies. As stated in the introduction, in this paper we do not focus on video-track features for computational reasons.

The audio-track contains immense amounts of useful information and it normally has closer link to semantic event than the video information. Some interesting early work was done by Arons [12] in trying to aggressively speed-up informational talks. He noticed that relative-pitch increases for people when they are emphasizing points. In his Speech Skimmer system, he used that for prioritizing regions within a talk. He *et al* [13] further built upon Aron's work and constructed presentation summaries based on pitch analysis, knowledge of slide transitions in the presentation, and information about previous users' access patterns. The study showed that the automatically generated summaries were of considerable value to the talk viewers. As we will discuss later, we use pitch as one component for emphasis detection in this paper too.

Use of close-caption information (e.g., Informedia project [9]) is a special case of speech track analysis; ideally if speech-to-text conversion were perfect, one would not have to rely on close-caption information. However, we are far from ideal today, and close caption text is a powerful source to classify video segments for indexing and searching. For this paper, as is the case in practice, we assume close caption information is not available for baseball games.

As one moves away from relatively clean speech environments (e.g., news, talks), analysis of audio-track can become trickier. For example, in sports videos, there are several sources of audio—the announcer, the crowd, noises such as horns — are all mixed together. These sound sources need to be separated, if their features are to be used in analysis and segmentation of video. The CueVideo system from IBM [15] presents techniques to separate speech and music in mixed-audio environments. They use a combination of energy, zero-crossing rate, and analysis of harmonics. In [16], Zhang and Kuo developed a heuristic-based approach to classifying audio signals into silence, speech, music, song, and mixtures of the above. While both systems achieve good accuracy, the selection of many hard-coded thresholds prevents them from being used in a more complicated audio environment such as baseball games. As we discuss in later sections, the audio channel in TV baseball programs is very noisy, the sound sources more diverse, and we want to detect special features like baseball bat-and-ball impact that have not been addressed earlier.

Looking at related work in the sports domain, we see that relatively little work has been done on sports video as compared to news video. This is partly due to the fact that the analysis is more difficult for sports, for example, due to lack of regular structure in sports video (in contrast, news often has structured format: anchor person → clip from the field → back to anchor person) and more complex audio. In some early work, Gong *et. al.* [17] targeted at parsing TV soccer programs. By detecting and tracking soccer court, ball, players, and motion vectors, they were able to distinguish nine different positions of the play (e.g. midfield, top-right corner of the court, etc.). While Gong *et al* focused on video track analysis, Chang *et. al*.[18] primarily used audio analysis as an alternative tool for sports parsing. Their goal was to detect football touchdowns. A standard template matching of filter bank energies was used to spot the key words "touchdown" or "fumble". Silence ratio was then used to detect "cheers", with the assumption that little silence is in cheering while much more are in reporter chat. Vision-based line-mark and goal-posts detection were used to verify the results obtained from audio analysis. Our work reported here is similar in spirit though different in detail.

3. INFORMATION SOURCES

As discussed in previous section, the two primary sources of information are video-track and audio-track. Video/visual information captures the play from various camera distances and angles. One can possibly analyze the video track to extract *generic features* such as: high-motion scene or low-motion scene; camera pan, zoom, tilt actions; shot boundaries. Alternatively, as done by Gong *et. al.* and Chang *et. al* for soccer and football, we can detect *sport-specific features*. For baseball, one can imagine detecting situations such as: player at bat, the pitcher curling-up to pitch the ball, player sliding into a base, player racing to catch a ball. Given our goal of determining exciting segments, we believe sport-specific features are more likely to be helpful than the generic features. For example, interesting action usually happens right after the ball is pitched, so detecting the curled-up pitching motion sequences can be very helpful, especially when coupled with the audio-track analysis.

The technology to do such video-analysis while challenging seems within reach. However, we do not use video analysis in this paper. We had two reasons. First, visual information processing is compute intensive, and we wanted to target set-top box class of machines. For example, to compute the dense optical flow field of a 320x240 frame, it needs a few seconds on a high-end PC even using the hierarchical Gaussian pyramid [19]. Second, we wanted to see how well we can do with audio information only. As we discuss below, we thought we could substitute for some of the visual cues with cheaper-to-compute audio cues. For example, instead of detecting beginning of a play with a curled-up pitcher visual sequence, we decided to explore if we could locate it by detecting bat-and-ball impact points from the audio track.

There are four major sources mixed in: 1) announcers' speech, 2) audience ambient speech noise, 3) game-specific sounds (e.g. baseball hits), and 4) other background noise (e.g. vehicle horning, audience clapping, environmental sounds, etc.). A good announcer's speech has tremendous amount of information, both in terms of actual words spoken (if speech-to-text were done) and in terms of prosodic features (e.g., excitement transformed into energy, pitch, and word-rate changes). The audience ambient noise can also be very useful, as audience viscerally react to exciting situations. However, in practice this turns out to be an unreliable source, because automatic gain control (AGC) affects the amount of audience noise picked up by the microphones. It varies quite a bit depending on whether the announcer is speaking or not. Game specific sounds, such as bat-and-ball impact sound, can be a very useful indicator of the game development. However, AGC and the far distance from the microphones make detecting them challenging. Finally, vehicle horning and other environmental sounds happen arbitrarily in the game. They therefore provide almost no useful, if not negative, information to our task.

Based on the above analysis, in this paper, we will use announcers' speech and game specific sound (e.g., baseball hits) as the major information sources and fuse them intelligently to solve our problem at hand. We make the following assumptions in extracting highlights from TV broadcasting baseball programs:

1. Exciting segments are highly correlated with announcers' excited speech;
2. Most of the exciting segments in baseball games occur right after a baseball pitch and hit.

Under the above two assumptions, the *challenges* we face are: develop effective and robust techniques to detect excited announcers' speech and baseball hits from the mixed and very noisy audio signal, and intelligently fuse them to produce final exciting segments of baseball programs. Before we going into full details of the proposed approach in Section 4, we first examine various audio features that will be used in this paper.

3.1. Audio Features Used

3.1.1 Energy Related Features

The simplest feature in this category is the short-time energy, i.e., the average waveform amplitude defined over a specific time window.

When we want to model signal's energy characteristics more accurately, we can use sub-band short-time energies. Considering the perceptual property of human ears, we can divide the entire frequency spectrum into four sub-bands, each of which consists of the same number of critical bands that represent cochlear filters in the human auditory model [14]. These four sub-bands are 0-630hz, 630-1720hz, 1720-4400hz, and 4400hz and above. Let's refer them as E_1, E_2, E_3, and E_4. Because human speech's energy resides mostly in the middle two sub-bands, let's further define $E_{23} = E_2 + E_3$.

3.1.2 Phoneme-level Features

The division of the sub-bands based on human auditory system is not unique. Another widely used sub-band division is the Mel-scale sub-bands [20]. For each tone with an actual frequency, *f*, measured in Hz, a subjective pitch is measure on a so called "Mel-scale". As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 Mels. In plain words, Mel-scale is a gradually warped linear spectrum, with coarser resolution at high frequencies. The Mel-frequency sub-band energy is defined accordingly. For automatic speech recognition, many phoneme-level features have been developed. Mel-frequency Cepstral coefficients (*MFCC*) is one of them [20]. It is the c*osine* transform of the Mel-scale filter bank energy defined above. *MFCC* and its first derivative capture fine details of speech phonemes and have been a very successful feature in speech recognition and speaker identification.

3.1.3 Information Complexity Features

There are quite a few features that are designed for characterizing the information complexity of audio signals, including bandwidth and entropy. Because of entropy's wide use and success in information theory applications, in this paper we will concentrate on entropy (*Etr*). For an *N*-point FFT of the audio signal *s(t)*, let *S(n)* be the *nth* frequency's component. Entropy is defined as:

$$Etr = -\sum_{n=1}^{N} P_n \log P_n$$

$$P_n = |S(n)|^2 / \sum_{n=1}^{N} |S(n)|^2$$

3.1.4 Prosodic Features

The waveform of voiced human speech is a quasi-periodic signal. The period in the signal is called the pitch (*Pch*) of the speech. It has been widely used in human speech emotion analysis and synthesis [21]. Independent of the waveform shape, this period can be shortened or enlarged as a result of the speaker's emotion and excitement level. There are many approaches to pitch estimation, including auto-regressive model and average magnitude difference function [16], etc. The pitch tracker we use in this paper is based on the maximum *a posteriori* (MAP) approach [22]. It creates a time-pitch energy distribution based on predictable energy that improves on the normalized cross-correlation and is one of best pitch estimation algorithms available.

3.1.5 Summary

We have discussed various audio features in this section. They are designed for solving different problems. Specifically, we will use E_{23}, *Etr*, and *MFCC* for human speech endpoint detection. E_{23} to E_4 are used to build a temporal template to detect baseball hits. Statistics based on E_{23} and *Pch* are used to model excited human speech.

4. PROPOSED APPROACH

In this section we will first give an algorithm overview and then discuss each sub-systems in full detail.

4.1 Algorithm Overview

As stated in Section 3, we base our algorithm for highlight detection on a model of baseball where we assume: (i) exciting segments are highly correlated with announcers' excited speech; and (ii) most exciting segments in baseball occur right after a baseball pitch and hit. As a result, we need to develop techniques to reliably detect excited human speech and baseball hits, and then fuse them intelligently to generate the final highlights segments. The following is the flowchart of the algorithm.

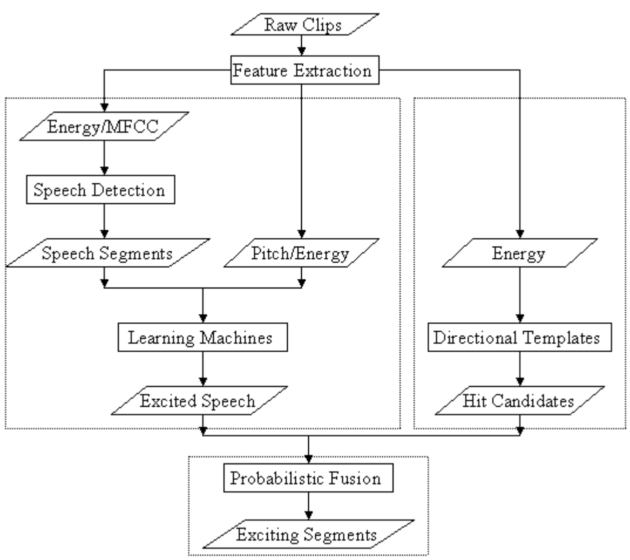


Figure 1. Algorithm Flowchart

The top-left block is the sub-system for excited speech classification, including the pre-processing stage of noisy environment speech endpoint detection. The top-right block is the sub-system for baseball hits detection. The bottom block is the sub-system for probabilistic fusion.

1. *Noisy Environment Speech Endpoint Detection:* In conventional speech endpoint detection, the background noise level is relatively low. An energy-based approach can achieve reasonably good results. Unfortunately, in TV baseball programs, the noise presence can be as strong as the speech signal itself, and we need to explore more sophisticated audio features to distinguish speech from other audio signals.
2. *Classifying Excited Speech Using Learning Machines:* Once speech segments are detected, the energy and pitch statistics are computed for each speech segment. These statistics are then used to train various learning machines, including pure parametric machines (e.g., *Gaussian fitting*), pure non-parametric machines (e.g., *K nearest neighbors*), and semi-parametric machines (e.g., *support vector machines*). After the machines are trained they are capable of classifying excited human speech for other baseball games.
3. *Detecting Baseball Hits Using Directional Templates:* Excited announcers' speech does not correlated *100%* with the baseball game highlights. We should resort to additional cues to support the evidence that we obtained from excited speech detection. Sports-specific events, e.g., baseball hits, provide such additional support. Based on the characteristics of baseball hits' sub-band energy features, we develop a directional template matching approach for detecting baseball hits.
4. *Probabilistic Fusion*: The outputs from Steps 2 and 3 are the probabilities if an audio sequence contains excited human speech and contains a baseball hit, respectively. Each one of two probabilities alone does not provide enough confidence in extracting true exciting highlights. However, when integrated appropriately, they will produce stronger correlations to the true exciting highlights. We will develop and compare various approaches to fuse the outputs from Steps 2 and 3.

Based on the nature of each processing steps, different audio signal *resolutions* are used. All of the original audio features are extracted at the resolution of 10 msec (referred as *frames*). The frame-resolution E_{23} and E_4 are used in *directional template matching* to detect baseball hit candidates. In speech endpoint detection, human speech seldom is less than half a second. We therefore use 0.5 sec resolution (referred as *windows*). The statistics of *Pch* and E_{23} are extracted from each *window* to conduct excited speech recognition.

One thing worth emphasizing is that the whole proposed approach is established on a probabilistic framework. Unlike some of the existing work that uses heuristics to set hard thresholds, we try to avoid thresholding during the intermediate stages. In the thresholding approaches, early misclassifications cannot be remedied at later stages. The probabilistic framework approach will, on the other hand, produce probability values at each intermediate stage not a 0/1 decision. This probabilistic formulation of the problem allows us to avoid *ad hoc* procedures and solve the problem in a principled way.

5. Noisy Environment Speech Detection

Most of the traditional speech endpoint detection techniques make the assumption that the speech is recorded in a quiet room environment. In that case, E_{23} alone can produce reasonably good results. At a baseball stadium, however, human speech is almost always mixed with other background noise, including machinery noise, car horns, background conversations, etc [20].

In this case, E_{23}'s distinguishing power drops significantly, because microphone's AGC amplifies the background noise level when the announcers are not talking. The energy level of non-speech signal can therefore be as strong as that of speech.

In a recent work by Huang and Yang [23], they proposed to use a hybrid feature (product of energy E_{23} and entropy *Etr*) to perform noisy car environment speech endpoint detection. Based on our experiments, even though this approach is effective in car environment, its performance drops significantly in baseball stadium environment that has much more varieties of background interferences.

Inspired by the success of *MFCC* in automatic speech recognition, and the observation that speech exhibits high variations in *MFCC* values, we propose to use first derivatives of *MFCC* (delta *MFCC)* and E_{23} as the audio features. They are complimentary in filtering out non-speech signals: energy E_{23} helps to filter out low energy but high variance background interference (e.g., low volume car horns) and delta *MFCC* helps to filter out low variance but high energy noise (e.g., audience ambient noise when AGC produces large values). In Section 5, we compare the performance of the above three approaches: energy only, energy and entropy, and energy and delta *MFCC*.

5.1 Classifying Excited Human Speech

A good announcer's speech has tremendous amount of information, both in terms of actual words spoken (if speech-to-text were done) and in terms of prosodic features (e.g., excitement transformed into energy and pitch). As speech-to-text is not reliable in noisy environment, in this paper we concentrate on the prosodic features. Excited announcers' speech has good correlations with the exciting baseball game segments. Previous study has shown that excited speech has both raised pitch and increased amount of energy [21]. The features we use in this paper are therefore statistics of pitch *Pch* and energy E_{23} extracted from each 0.5 sec speech *windows*. Specifically, we use six features: maximum pitch, average pitch, pitch dynamic range, maximum energy, average energy, and energy dynamic range of a given speech *window*.

The problem of classification can be formulated as follows. Let C_1 and C_2 be the two classes to be classified (e.g., excited speech vs. non-excited speech). Let X be the observations of the features (e.g., the six audio features described above). Let $P(C_i \mid X), i = 1, 2,$ be the *posterior* probability of a data being in class C_i given the observation X. *Bayes* decision theory tells us that classifying data to the class whose *posterior* probability is the highest minimizes the *probability of error* [24]:

$$\arg\max_i P(C_i \mid X)$$

How to reliably estimate $P(C_i|X)$ is the job for learning machines. We next explore three different approaches.

5.1.1 Parametric Machines

Bayes rule tells us that $P(C_i \mid X)$ can be computed as a product of the *prior* probability and the conditional class density, and then normalized by the data density:

$$P(C_i \mid X) = \frac{P(C_i)\, p(X \mid C_i)}{p(X)}$$

As $p(X)$ is a constant for all the classes and does not contribute to the decision rule, we only need to estimate $P(C_i)$ and $p(X|C_i)$. Priors $P(C_i)$ can easily be estimated from labeled training data (e.g., excited speech and non-excited speech). There are many ways to estimate the conditional class density $p(X|C_i)$. The simplest approach is the *parametric* approach. This approach represents the underlying probability density as a specific functional form with a number of adjustable parameters [24]. The parameters can be optimally adjusted to best fit the training data. The most widely used functional form is *Gaussian* (Normal) distribution $N(\mu,\sigma)$, because of its simple form and many nice analytical properties. The two parameters (mean μ and standard deviation σ) can be optimally adjusted by using the *maximum likelihood estimation* (MLE):

$$\mu = \frac{1}{n}\sum_{k=1}^{n} X_k \,,\; \sigma^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \mu)$$

where n is the number of training samples.

5.1.2 Non-Parametric Machines

Even though easy to implement, parametric machines are too restrictive in data modeling and sometimes result in poor classification results. For example, the pre-assumed function seldom matches the true underlying distribution function and it can only model unimodal distributions [24]. Non-parametric machines were proposed to overcome this difficulty. They do not pre-assume any functional forms, but instead depend on the data itself. There are non-parametric machines that can estimate the *posterior* probability $P(C_i \mid X)$ directly. *K nearest neighbor* is such a technique.

Let V be the volume around observation X and V covers K labeled samples. Let K_i be the number of samples in class C_i. Then the *posterior* probability can be estimated as [24]:

$$P(C_i \mid X) = \frac{K_i / nV}{\sum_i K_i / nV} = \frac{K_i}{K}$$

This estimation matches our intuition very well: the probability that a data sample belongs to class C_i is the fraction of samples in the volume labeled as class C_i.

5.1.3 Semi-Parametric Machines

Pure parametric machines are easy to train and fast to adapt to new training samples, but too restrictive. Non-parametric machines, on the other hand, are much more general but take more time to compute. To combine the advantages and avoid the disadvantages of the above two approaches, *semi-parametric* machines have been proposed [25]. These new set of machines include the *Gaussian mixture models*, n*eural networks* and *support vector machines (SVM)*. Because of its recognized success in pattern classification [26], we will focus on SVM in this paper.

Let R be the actual risk (test error) and Re be the empirical risk (training error). For η, where $0< \eta<1,$ with probability $1 - \eta$, the following bound holds [26]:

$$R \le R_e + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

where n is the number of training samples and η is a non-negative integer called the *Vapnik-Chervonenkis* (VC) dimension of a learning machine [26]. R (test error) represents a learning machine's ability to generalize to unseen data, after it is trained.

In any classification task, we want R to be minimized. It is not always true that R will be minimized when Re is minimized. The second term on the right-hand side determines the "mismatch" between training and testing situations, and it increases as the VC dimension increases. VC dimension characterizes the "capacity" of a learning machine. If the capacity is *too low*, the machine cannot learn and results in a high Re (thus high R). On the other hand, if the capacity is *too high*, even though Re can be arbitrarily small, the machine can be "over fit" and results in a high value of the second term (thus high R). The remarkable characteristic of SVM is that it can automatically find the required "capacity" to learn the training samples without being over trained. In another word, SVM learns in a principled way. SVM has found successes in many applications including face detection, hand writing recognition, and text categorization [26].

Standard SVM does not generate the *posterior* probability directly. In [27], Platt developed a new approach to first train a SVM and then to train an additional *sigmoid* function to map the SVM outputs into *posterior* probabilities. Because of its effectiveness, we adopted this method in our system:

$$P(C_i \mid X) = \frac{1}{1+\exp(AX+B)}$$

where A and B are the parameters of the *sigmoid* function.

In Section 5, we give detailed comparisons between the above three learning machines' performance.

5.1.4 Post Processing

In real world, excited speeches never appear in just one *window* (0.5 sec). Instead, they appear in a much longer unit. Experimentally, we find a *segment* (5 sec) is the minimum length required by a coherent excited speech. Since each *window* contributes equally to a *segment*, we use the average *posterior* probability of the *windows* in the *segment* as the *posterior* probability for the *segment* $P(ES)$:

$$P(ES) = \frac{1}{M}\sum_{m=1}^{M} P(C_1 \mid X_m)$$

where C_1 represents the excited speech class, and M is the number of *windows* in a *segment*.

5.2 Baseball Hit Detection

Even though excited announcers' speech has good correlations with exciting baseball game segments, it is not sufficient or reliable to base the judgment solely on the excited speech. For example, the pitch tracker may perform poorly in noisy speech environment. More importantly, announcers' speech can become excited due to other reasons that are totally irrelevant with the development of the game (e.g., a joke from their partners or a balloon passing the stadium). If we were to use excited speech only, there would have been many false alarms.

In most of the sports, there exist sports-specific events. For example, *player gatherings* indicate the start of new attacks in football, and baseball hits manifest possible exciting segments a few seconds later in the game. These sports-specific events can help reduce the amount of false alarms. In this section, we will describe a *directional template matching* approach to detecting baseball hits.

In the audio signal spectrograms, when we examine a baseball hit in isolation, it is extremely difficult to distinguish it from a strong speech fricative or a stop. However, when we look at it in the context of its surrounding signals, while the task is still difficult we have some hope: fricatives or stops normally are followed by vowels that exhibit high energy in E_{23} but low energy in E_4. To capture this temporal context, we build a baseball-hit template consisting of 25 *frames*, with the hit peak at the 8^{th} *frame*. In this template, using only the absolute values of E_{23} and E_4 is not sufficient. To capture the shape of the energy curves over time, we further use the ratio of E_{23} and E_4 normalized by $E_{23}(8)$:

$$ER_{23}(i) = E_{23}(i) / E_{23}(8)$$
$$ER_4(i) = E_4(i) / E_{23}(8)$$

where $i = 1, \ldots, 25$.

The four 25-element templates are constructed based on labeled training data. Figure 3 shows the four templates (E_{23}, ER_{23}, E_4 and ER_4 in that order) built on 55 training samples.

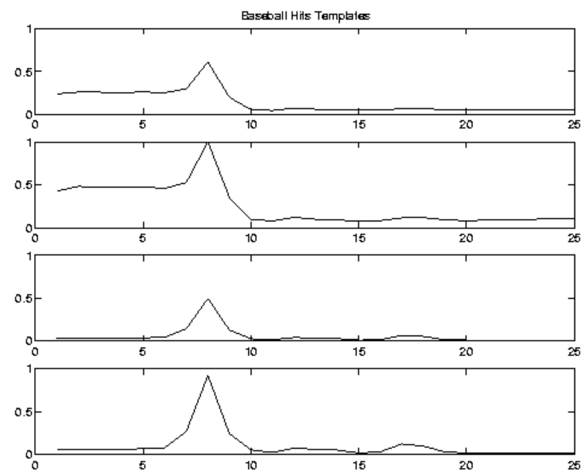


Figure 2. Baseball hit's template

We next discuss how we compute the probability that a data sequence (25 *frames*) contains a baseball hit. Let D be the Mahalanobis distance of a data sequence $\vec{X}$ from the template $\vec{T}$:

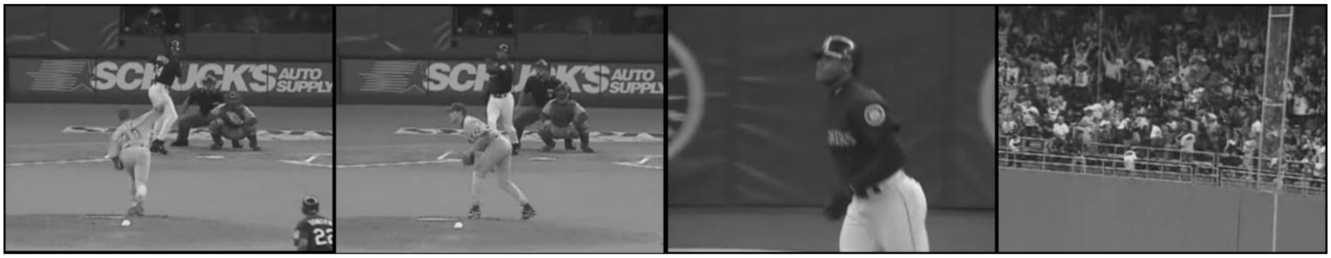
$$D^2 = (\vec{X} - \vec{T})^T \Sigma^{-1} (\vec{X} - \vec{T})$$

where both $\vec{X}$ and $\vec{T}$ are vectors of $4 \times 25 = 100$ elements, and Σ is the covariance matrix of $\vec{T}$. To simplify computation, we restrict Σ to be a diagonal matrix. The distance D can be converted to a probability value as follows [28]:

$$P(HT) = \exp(-\frac{1}{2}D^2) / (C + \exp(-\frac{1}{2}D^2))$$

where C is a suitable constant.

The above conventional (*un-directional*) template matching technique does not incorporate domain knowledge into the computation of D effectively. For example, domain knowledge (Figure 2) tells us that $E_{23}(8)$ should exhibit high value while other $E_{23}(i)$'s exhibit low values. But in the un-directional template matching, an over-mismatch data point of $E_{23}(8)$ is treated the same as an under-mismatch data point of $E_{23}(8)$. In reality, however, an over-mismatch should not only not to be punished, but also be encouraged. The *direction* from which the data point is approaching the template *is* important. We thus propose a *directional* template matching approach:



(a) (b) (c) (d)

Figure 3. A typical presentation of an exciting segment: It starts with the pitcher throwing the ball (a). Then the hitter tries to hit the ball (b). If it is a good hit, then the hitter is running (c). The final part (d) is the audience cheering for the good play.

$$D^2 = (\vec{X} - \vec{T})^T I \times \Sigma^{-1} (\vec{X} - \vec{T})$$

where I is a diagonal indicator matrix. Its elements can be of various values to reflect the domain knowledge. For example, $I(8,8)$ takes a negative value when $E_{23}(8)$ is an over-mismatch to reduce the distance D but a positive value for an under-mismatch to increase the distance D. This new formulation makes template matching much more flexible to incorporate domain knowledge into the distance computation. Specifically, in this paper, when $E_{23}(i)$'s are over-matching, $I = diag[1, \ldots, 1, -1, 1, \ldots, 1]$, where the –1 is at location 8. When $E_{23}(i)$'s are under-matching the templates, $I = diag[-1, \ldots, -1, 1, -1, \ldots, -1]$, where the 1 is at location 8.

5.3 Probabilistic Fusion

In the previous two sections we have developed techniques to compute the probability that a *segment* is an excited speech segment ($P(ES)$) and the probability that a *frame* contains a baseball hit ($P(HT)$). The two assumptions we made in Section 3 tell us that if a segment has high $P(ES)$ and it occurs right after a high $P(HT)$ frame, it is very likely to be a true exciting segment. From the training data, we find that a hit can occur upto 5 sec ahead of the excited speech segment. In all the following discussions, we search a hit *frame* within the 5 sec interval of the excited speech segment. We next explore two techniques to fuse $P(ES)$ and $P(HT)$ into the final probability if a segment is an exciting segment ($P(E)$).

5.3.1 Weighted Fusion

In this approach, both $P(ES)$ and $P(HT)$ directly contribute to $P(E)$, with appropriate weights:

$$P(E) = W_{ES} P(ES) + W_{HT} P(HT)$$

where W_{ES} and W_{HT} are the weights that sum up to *1.0*. They can both be estimated from the training data, and we use values of *0.83* and *0.17*.

5.3.2 Conditional Fusion

In this approach, we try to capture the intuition that the key value of a detected hit P(HT) is not in directly adding to the probability that a segment is exciting P(E). Instead it contributes indirectly to P(E) by adjusting the by adjusting the confidence level of the $P(ES)$ estimation (e.g., that the excited speech probability is not high due to mislabeling a car horn as speech):

$$P(E) = P(CF)P(ES)$$
$$P(CF) = P(CF \mid HT)P(HT) + P(CF \mid \overline{HT})P(\overline{HT})$$

where $P(CF)$ is the probability that how much confidence we have in $P(ES)$ estimation, and $P(\overline{HT}) = 1 - P(HT)$ is the probability that there is no hit. $P(CF|HT)$ represents the probability that we are confident that $P(ES)$ is accurate *given* there is a baseball hit. Similarly, $P(CF \mid \overline{HT})$ represents the probability that we are confident that $P(ES)$ is accurate *given* there is no baseball hit. Both conditional probabilities $P(CF|HT)$ and $P(CF \mid \overline{HT})$ can be estimated from the training data and we obtain values *1.0* and *0.3*.

5.4 Final Presentation

Starting at the beginning of the algorithm, various probability values are computed and flow to the end of the algorithm. This probabilistic framework allows us to avoid information loss due to intermediate-step hard thresholding and can solve the problem in a principled way. At the end of the algorithm flow chart, there is only a single probability value ($P(E)$) associated with each segment.

When presenting an exciting segment to the end user, overlapping and close-by segments are merged into a single segment. In addition, because we already know the most likely baseball hit locations, each segment starts a few seconds before the hit. Figure 3 is a typical sequence of an exciting segment.

Depending on users' interest level and/or time available to view the game, the users can specify an interest threshold. This is the only threshold that a user needs to specify. Based on this threshold, the algorithm generates a summary of suitable duration.

Of course, the algorithm may generate false positives and negatives. Lowering the threshold will minimize false negatives (reduce missing exciting segments) though it may increase false positives (include non-exciting segments). Our belief is that if these are few, then the benefits of automation will far exceed the costs. In WebTV/TiVo/ReplayTV environments it is particularly easy for the end-user to skip incorrectly identified false positives due to the instant seek capability.

6. EXPERIMENTAL RESULTS

In this section, we will give detailed reports on our experiments to evaluate various proposed approaches. We will describe the data set used, evaluation framework, experimental results and observations.

6.1 Data Set

In most of the existing systems, only limited amount of tests have been conducted (e.g. less than 1 hr video in [15], 45 min in [18],

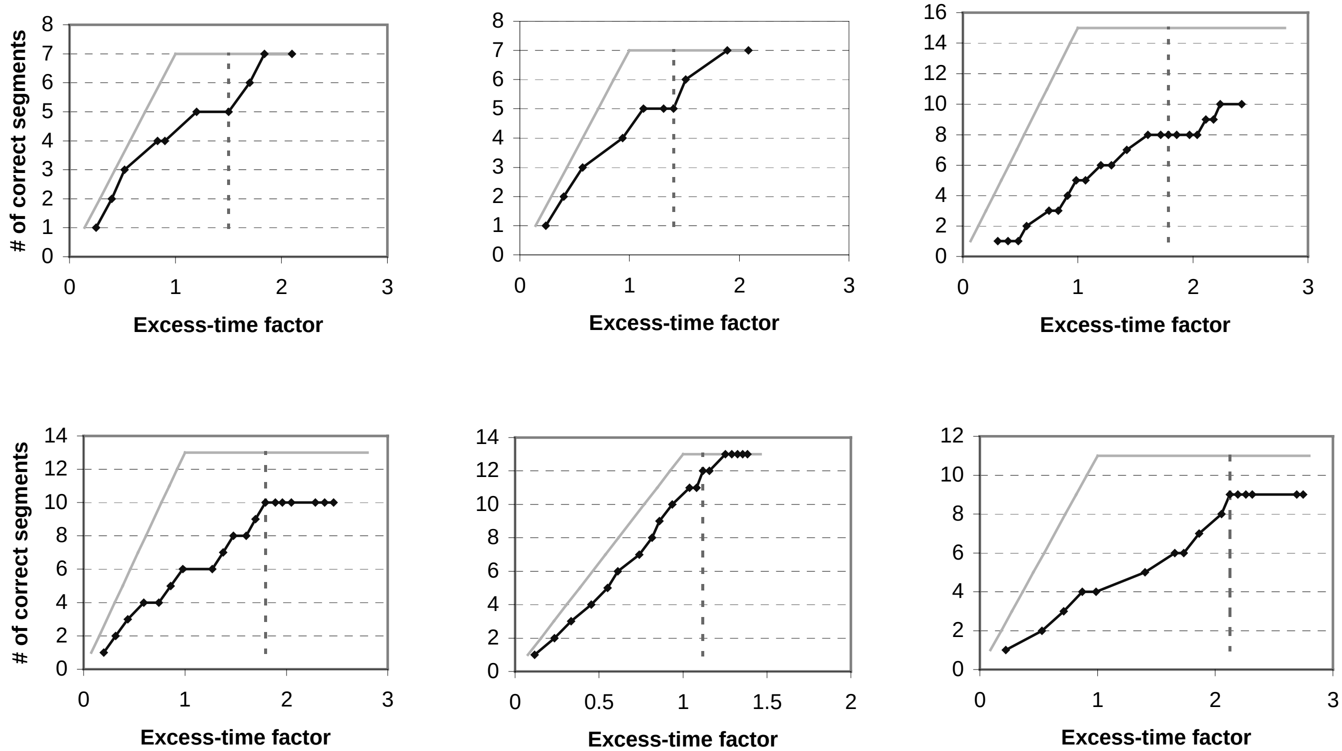


Figure 4. Overall performance curves for clips A through F, in raster scan order. Y-axis shows number of exciting segments identified correctly by the algorithm. X-axis indicates excess-time factor, i.e., duration of algorithmically selected segments divided by duration of human selected segments. For each graph, the left light-gray curves shows *ideal* performance, corresponding to choices made by human. Right dark curve shows our algorithm's performance assuming *E+MFCC* for speech selection, SVM for classifying exciting speech segments, and using conditional fusion for including baseball hit data. The vertical dash line indicates the time duration of algorithmically selected segments when threshold was set so that the number of segments selected was same as that generated by human. The overall graph was plotted with a slightly lower threshold, with number of generated segments being 1.5 times human segments. This allows us to see if we capture some more of the human selected segments if we lower the threshold.

Table 1. Data set. The clips cover about 7 hours of video, with 4 different announcers. The Energy Level is given as compared to maximum allowable level (in percentage).

Clip	Length	Announcer	Samp. Fr.	Energy Lev
A	1:10:05	A	16 KHz	50
B	1:05:34	A	16 KHz	55
C	1:01:54	B	16 KHz	80
D	0:41:14	C	16 KHz	80
E	1:58:26	A	11 KHz	30
F	1:06:19	D	16 KHz	120

and 30 min in [17]). To validate the effectiveness and robustness of the proposed approach, we have collected baseball game videos from various sources (see Table 1). In total we have seven hours of baseball games consisting of eight giga bytes of data. They come from different sources, digitized at different studios, sampled at different frequencies and amplitude, and reported by different announcers. The first half (35 min) of Clip A is used as the training data. The second half of Clip A is used as a *clip-dependent* test case. Clip B has many similar conditions as Clip A and is used as a *similar-clip* test case. Clips C and D differ significantly from Clip A, and are used as *clip-independent* test cases. To further stress test, we included Clips E and F. Clip E is sampled at a lower frequency and may lose some higher frequency information, as needed in the algorithm. Clip F's audio level was over amplified (clipped), i.e., 20% over the limit of maximum allowable level. These two tapes represent the *stress* test cases. A summary of the six clips is given in Table 1.

6.2 Evaluation Framework

We wanted to compare our automatically generated highlight segments to the ones marked by humans. A human subject (not working on the project) was asked to watch the baseball games A-F and mark the exciting segments. Given the certain amount of subjectivity in what is exciting, we would have ideally liked multiple people to do such markings. The results are quite interesting nonetheless.

There are two methods we use to evaluate our results. The first called "segment-overlap method" is as follows. We vary the threshold until the number of segments selected by our algorithm is the same as that selected by the human. We then ask the question how many of these are the same as those selected by the human. The larger the overlap, clearly the algorithm is performing better. We can also do sensitivity analysis by letting our algorithm select fewer or more segments than that selected by the human.

The second method, called “excess-time method”, is used to deal with a possible pitfall of the first method. For example, if the segments determined by an algorithm are very long (e.g., each is 2 minutes long) then obviously the probability of covering human-selected segments (each is typically about 10 seconds) would be higher, and first metric would indicate good results. However, that would not be as good as an algorithm that more tightly identifying the exciting segments. So in this method we plot the number of correctly generated segments as a function of T/T_0 (e.g., Figure 4). The numerator T corresponds to the duration of the algorithmically selected segments (ordered based on decreasing $P(E)$ values), and the denominator T_0 corresponds to the total duration of the human-selected segments. For example, a point on this graph could indicate that to get coverage of 5 of the 7 human-selected segments, we have to spend 1.4 times as long watching the video as duration of human-selected segments. These excess-time curves illustrate how the algorithm performs when more and more segments are added to the final presentation.

6.3 Overall Performance

We begin by comparing the performance of the best of our algorithms with the ground truth as marked by the human. The best overall algorithm combines energy plus delta *MFCC* for speech-endpoint detection, SVM for learning excited speech segments, and conditional fusion for including baseball hit information. Table 2 summarizes the performance when the threshold of our algorithm was set to pick the same number of segments as selected by human.

Table 2. Overall performance. Second row indicates # of segments selected by human. Third row indicates # of correct segments identified by algorithm, when asked to pick same number of segments as human.

Clip	A	B	C	D	E	F	Total
# human	7	7	15	13	13	11	66
# algorithm	5	5	8	10	12	9	49

Comparing the performance in Table 2, we see that algorithm identifies 49 out of 66 segments correctly (~75%). This is quite remarkable, if we consider that some of the exciting segments identified by the human start falling into the gray area, where there may have been others segments just as exciting to another human. The performance for clip C is poorest of all the clips (8 out of 15 correct), and this is due to pitch tracking reasons. We discuss this aspect in greater detail in Section 5.8 after discussing rest of results.

Figure 4 shows the overall performance using excess-time plots for all the clips (shown in raster-scan order). We also show the “ideal” curves corresponding to the ground truth, i.e., human selected segments. These represent the least amount of time to achieve the highest “correctness”. The vertical dashed lines in each graph indicates the time duration of algorithmically selected segments when threshold was set so that the number of segments selected was same as that generated by human (the correctness at this threshold is 49 out of 66 as shown in Table 2). The location of these dashed lines indicate how much more time a viewer need to spend to view the same number of segments as the human marked ones. For example, if the vertical dashed line’s location is 1.3, it says a viewer will spend 30% more time. The closer the line is to 1.0, the better the algorithm’s performance. As can be seen in the figure, except for Clip F, all the clips’ vertical lines are within 2.0, with average of ~1.5.

This is a strong result indicating that the algorithm is not achieving its correctness by marking up excessively long duration segments. In fact, even this factor of 1.5 is partly due to the fact that the human in our case was particularly conservative in identifying the exciting segments. For example, he did not include the pitcher pitching the baseball and player hitting the baseball in his highlighted segments; he only included the action after that. Our instructions to the human were not so precise, and we did not want to change his markings after the fact.

The overall graphs in Figure 4 were plotted with a slightly lower threshold, with number of generated segments being 1.5 times human segments. Thus if human identifies 10 segments, we adjusted the algorithm to generate 15 segments. These portions of the curves cover the region on the right of the vertical line, and also provide useful information. If the curves continue upwards, it means it is still beneficial to include more segments into the presentation at the cost of increased viewing time. On the other hand, if the curves become flat after the vertical lines, it is almost of no advantage to include more segments. The curves in Figure 4 show that it is still beneficial to include more segments (other than for Clips D and F). By increasing our excess-time factor slightly, we can achieve correctness of 57 out of 66 segments (~86%).

After establishing the overall performance of the proposed approach, we next examine various algorithms in greater detail along three orthogonal dimensions: speech endpoint detection, excited speech classification, and probabilistic fusion.

6.4 Speech-Endpoint Detection

We had presented three speech-segment endpoint detection algorithms in Section 4.2: energy only (*E*), energy and entropy (*E+Etr*), and energy and delta *MFCC* (*E+MFCC*). We now explore the impact of the speech-endpoint detection algorithm on the overall end results. For this comparison, we fix the other control conditions: the learning algorithm is fixed to SVM (it was the best as we will show later), and the hit-detection and fusion algorithm to “conditional fusion”.

The relative performance is summarized in Table 3. It is clear that overall *E+MFCC* does substantially better (49 out of 66 correct) than the other two approaches, while *E+Etr* does better than E alone (40 vs. 30 out of 66). *E+MFCC* does best for each of the six individual clips (A-F) too, while there are some performance reversals between *E* and *E+Etr* (clips C and F).

Table 3. Performance of various speech-endpoint detection algorithms. Second row indicates # of segments selected by human. Subsequent rows indicate correct segments identified by algorithm, when asked to pick the same number of segments as human.

Clip	A	B	C	D	E	F	Total
# human	7	7	15	13	13	11	66
E+MFCC	5	5	8	10	12	9	49
E+Etr	5	5	7	9	9	5	40
E	4	4	8	5	2	7	30

6.5 Excited Speech Classification

In Section 4.3, we discussed three approaches to excited speech classification: Gaussian fitting (*GAU*), K nearest neighbors

(*KNN*), and support vector machines (*SVM*). Table 3 summarizes the impact of the different learning machines on the overall end results. For this comparison, we fix the other control conditions: we use *E+MFCC* as the speech endpoint detection algorithm and use conditional fusion as the fusion algorithm.

While *SVM* performs the best in the three learning machines as we expected, the gain is not significant. After analyzing the data, we found one major reason accounting for this was the following. The input to all the learning machines was the pitch and energy statistics of each speech *window(*Section 4.3*).* Our proposed *E+MFCC* did a very good job in separating other audio signals from human speech. Once this is done, excited speech classification becomes less difficult and less sophisticated learning paradigms (e.g., *GAU* and *KNN*) can achieve reasonable good results. One thing worth pointing out is that, even though *KNN* achieves almost the same accuracy as *SVM*, it is the slowest of the three learning machines.

Table 4. Performance of the three learning machines. Second row indicates # of segments selected by human. Subsequent rows indicate correct segments identified by algorithm, when asked to pick the same number of segments as human.

Clip	A	B	C	D	E	F	Total
# human	7	7	15	13	13	11	66
SVM	5	5	8	10	12	9	49
GAU	5	5	8	9	12	7	46
KNN	5	5	8	9	12	9	48

6.6 Baseball Hits Detection

The output of the *directional* template matching (Section 4.4) is the probability if a *frame* contains a baseball hit. Even though there is no need to set any threshold at this intermediate stage, we can set a threshold (*TH*) for *evaluation* purpose. We vary *TH* from 0.05 to 0.5 and Table 5 summarizes the baseball hits detection performance for Clip D. (We did not do other Clips due to resource involved in marking the ground truth.) There are *58* true baseball hits in this clip. Considering many baseball hits are corrupted by background noise and even completely overlapping with announcers' speech, the proposed approach's performance is very encouraging. For example, at *TH = 0.20*, it detects 47 (81%) of all the true hits and only introduced 8 (less than 14%) of false positives. Among the undetected hits, for some the audio was too weak to be detected even by human, and in ground truth we simply assumed there was a hit based on video analysis.

Table 5. Baseball hits detection

TH	.05	.10	.15	.20	.30	.40	.50
Correct	53	50	47	47	41	32	23
False Alarms	23	13	9	8	2	2	1

6.7 Probabilistic Fusion

In Section 4.5, we proposed two methods to fuse *P(ES)* and *P(HT):* weighted fusion and conditional fusion. Table 6 summarizes the performance between conditional fusion, weighted fusion, and no fusion – just use *P(ES)* and discard *P(HT).* For this comparison, we fix the other control conditions: we use *E+MFCC* as the speech endpoint detection algorithm and use SVM as the learning machine for classifying excited speech.

We find no significant difference between the two fusion algorithms. When we looked at the details, we found that conditional fusion was giving more weight to hits than weighted fusion. As a result, when conditional fusion was used, if hits were correctly identified the algorithm did a better job. If, however, an actual hit was not detected, the algorithm often resulted in a mis-classification. On the balance, the results looked the same as weighted fusion, that gave an overall low level of importance to presence of hits.

Table 6 shows, however, that both conditional fusion and weighted fusion outperform no-fusion by about 8% (column Total in Table 6). This demonstrates that *sports-specific* features (e.g., baseball hits) provide useful cues to calibrate the accuracy of *generic* features (e.g. pitch estimation) and thus improve the overall system performance. We believe such features can also be valuable for sports like Golf, which have an impact involved and share the property with baseball of considerable slack time between exciting plays.

Table 6. Performance of the three learning machines. Second row indicates # of segments selected by human. Subsequent rows indicate correct segments identified by algorithm, when asked to pick the same number of segments as human.

Clip	A	B	C	D	E	F	Total
# human	7	7	15	13	13	11	66
Cond fusion	5	5	8	10	12	9	49
Wei. fusion	5	6	8	9	12	9	49
No fusion	5	5	8	7	12	8	45

6.8 Discussion

When we examine the highlights marked by the human subject, there are different exciting levels associated with the highlights. Some of highlights are clearly very exciting and most people will agree that they are exciting segments. Others, however, are subtle: they are exciting to some degree and from a certain perspective. After our experiments, we discussed with the human subject for some of the segments he marked but the algorithm missed, and some segments the algorithm detected but he did not select. He agreed that those segments belong to the "gray area", where even humans may have different answers. On the other hand, our algorithm almost never misses the really exciting segments. Considering the "gray area" effects, 49 out of 66 (75%) accuracy is a very encouraging result. In fact, if we ignore clip C for which the accuracy is the worst (we discuss clip C below), the overall accuracy increases to 41 out of 51 (~80%). We also have the possibility of increasing coverage by asking the algorithm to generate a larger # of segments than that generated by the human. While this would increase number of false positives, this might work well in practice, because given the instant-seek functionality provided by WebTV/TiVo/Replay boxes, it is very easy for end-user to skip incorrectly identified exciting segments.

The algorithm missed quite a few highlights in Clip C. When we carefully traced the reason, we found the pitch tracker was giving wrong estimations. The pitch tracker [22] we used in this paper is already one of the best in speech research community. However, like other pitch trackers, it is designed and tuned to clean speech pitch estimation. Even though it performs well in those situations, it failed when the background noise's level is almost comparable to that of human speech.

Baseball hits detection is still far from satisfactory. This *sports-specific* event is very useful in providing additional cues for highlights detection. If we had more accurate hit detection, the performance of conditional fusion and weighted fusion would have more significantly outperformed that of no-fusion. Even with current hit detection accuracy, conditional fusion and weighted fusion already exhibit clear performance advantage (around 8%) over no-fusion.

7. CONCLUDING REMARKS

In this paper, we have explored solutions to the challenging task of extracting baseball game highlights on set-top devices. Our task is highly constrained by the computing power and noisy audio data. We presented effective techniques to speech detection in noisy environment. We show that energy level plus delta MFCC performs best, and it improves the final performance considerably over alternatives. We discussed the relative strength of three types of learning machines and successfully applied SVM in excited speech classification. To incorporate domain knowledge more flexibly, we developed a *directional* template matching approach to baseball hits detection and achieved encouraging results. Finally, we developed probabilistic framework that intelligently integrates $P(ES)$ and $P(HT)$. The proposed probabilistic framework does not lose useful information at intermediate stages and allows us to solve the problem in a principled way.

We tested various methods over a diverse collection of six baseball games covering 7 hours of game time. The results are very encouraging. When our algorithm is asked to generate the same number of highlight segments as marked by human subject, on average, 75% of these are the same as that marked by the human. When asked to generate 1.5 times the number of segments, the overlap increases to 86%. At the same time, the total duration of the algorithmically generated segments is not significantly more than that of human segments.

Future work plans include real use of the proposed system, for example, to create highlights for the hundreds of games that are broadcast during a baseball season. An implementation on a PC acting as a TiVo/WebTV/Replay box will let us explore how end-users react to the availability of such highlight metadata. We also plan to explore use of visual features to improve the system performance. Given the computing power constraints, visual features will be used only after audio features have filtered clear-cut cases. Use of visual features is also possible when highlights are detected on a server and then communicated to the set-top box over the Internet.

8. ACKNOWLEDGMENTS

The authors want to thank John Platt and Liwei He for their valuable discussions, and thank JJ Cadiz for providing ground truth of the highlight segments in the baseball games.

9. REFERENCES

[1] WebTV, http://www.webtv.com.

[2] TVReplay, http://www.tvreplay.com

[3] Tivo Inc., http://www.tivo.com

[4] Li, F.C., *et al. Browsing Digital Video*. in *ACM CHI*. 2000. Hague, The Netherlands.

[5] Zhang, H., *et al. Automatic Parsing of News Video*. in *IEEE Conference on Multimedia Computing and Systems*. 1994.

[6] *MPEG-7: Context and Objectives (V.7)*. in *ISO/IEC JTC1/SC29/WG11 N2207, MPEG98*. March 1998.

[7] ATVEF, http://www.atvef.com

[8] Rui, Y., T.S. Huang, and S. Mehrotra, *Constructing Table-of-Content for Videos.* Journal of Multimedia Sys., Sept 1999. 7(5): p. 359-368.

[9] Wactlar, H.D., *et al., Lessons Learned from Building a Terabyte Digital Video Library*. IEEE Computer, 1999(Feb): p. 66-73.

[10] McGee, T. and N. Dimitrova. *Parsing TV Program Structures for Identification and Removal of Non-Story Segments*. in *SPIE Conf. on Storage and Retrieval for Image and Video Databases*. 1999. San Jose, CA.

[11] Yeung, M., B.-L. Yeo, and B. Liu, *Extracting Story Units from Long Programs for Video browsing and Navigation*, in *Proc. IEEE Conf. on Multimedia Comput. and Syss*. 1996.

[12] Arons, B. *Pitch-based Emphasis Detection for Segmenting Speech Recordings*. in *International Conference on Spoken Language Processing*. 1994.

[13] He, L., *et al. Auto-Summarization of audio-video presentations*. in *ACM Multimedia*. 1999.

[14] Liu, Z., Y. Wang, and T. Chen, *Audio Feature Extraction and Analysis for Scene Segmentation and Classification.* Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, 1998. 20(1/2): p. 61-80.

[15] Srinivasan, S., D. Petkovic, and D. Ponceleon. *Towards Robust Features for Classifying Audio in the CueVideo System*. in *ACM Multimedia*. 1999. Orlando, FL.

[16] Zhang, T. and C.-C.J. Kuo. *Heuristic Approach for Generic Audio Data Segmentation and Annotation*. in *ACM Multimedia*. 1999. Orlando, FL.

[17] Gong, Y., *et al. Automatic Parsing of TV Soccer Programs*. in *IEEE Conf on Multimedia Computing and Systems*. 1995.

[18] Chang, Y.-L., *et al. Integrated Image and Speech Analysis for Content-Based Video Indexing*. in *IEEE Conf. on Multimedia Systems and Computing*. 1996.

[19] Rui, Y. and P. Anandan. *Segmenting Visual Action Units Based on Spatial-Temporal Motion Patterns*. in *IEEE Conf on Computer Vision and Pattern Recgonition*. 2000.

[20] Rabiner, L. and B.-H. Juang, *Fundamentals of Speech Recognition.* 1993.

[21] Dellaert, F., T. Polzin, and A. Waibel. *Recognizing Emotion in Speech*. in *IEEE ICASSP*. 1995.

[22] Droppo, J. and A. Acero. *Maximum A Posterior Pitch Tracking*. in *IEEE ICASSP*. 1999.

[23] Huang, L.-S. and C.-H. Yang. *A Novel Approach to Robust Speech Endpoint Detection in Car Environments (submitted)*. in *IEEE ICASSP*. 2000.

[24] Duda, R.O. and P.E. Hart, *Pattern Classification and Scene Analysis.* : p. 211-249.

[25] Bishop, C.M., *Neural Networks for Pattern Recognition*. 1994, Oxford, UK: Clarendon Press.

[26] Burges, C., *A Tutorial on Support Vector Machines for Pattern Recognition*, U. Fayyad, Editor. 1999, Kluwer Academic Publishers: Boston.

[27] Platt, J.C., *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*, in *Advances in Large Margin Classifiers*, P.B. Alexander J. Smola, Bernhard Scholkopf and Dale Schuurmans, Editor. 1999.

[28] Ishikawa, Y., R. Subramanya, and C. Faloutsos, *MindReader: Query databases through multiple examples*, in *Proc. of the 24th VLDB Conference*. 1998: New York.