An Audio-based Sports Video Segmentation and Event Detection Algorithm

Mark Baillie

Department of Computing Science
University of Glasgow
Glasgow, UK, G12 8QQ
bailliem@dcs.gla.ac.uk

Joemon M. Jose

Department of Computing Science
University of Glasgow
Glasgow, UK, G12 8QQ
jj@dcs.gla.ac.uk

Abstract

In this paper, we present an audio-based event detection algorithm shown to be effective when applied to Soccer video. The main benefit of this approach is the ability to recognise patterns that display high levels of crowd response correlated to key events. The soundtrack from a Soccer sequence is first parameterised using Mel-frequency Cepstral coefficients. It is then segmented into homogenous components using a windowing algorithm with a decision process based on Bayesian model selection. This decision process eliminated the need for defining a heuristic set of rules for segmentation. Each audio segment is then labelled using a series of Hidden Markov model (HMM) classifiers, each a representation of one of 6 predefined semantic content classes found in Soccer video. Exciting events are identified as those segments belonging to a crowd cheering class. Experimentation indicated that the algorithm was more effective for classifying crowd response when compared to traditional model-based segmentation and classification techniques.

1 Introduction

Live televised sporting events are now commonplace, especially with the arrival of dedicated digital channels. As a consequence, the volume of Sports video produced and broadcasted has increased considerably over recent years. Where such data is required to be archived for reuse, automatised indexing [2, 4, 15, 16] is a viable alternative to the manual labour intensive procedures currently in practise. To date feasible solutions have not been developed.

Current advancements include the automatic identification of low level semantic structures such as shot boundaries [4], semantic units [3, 16] and genre classification [15]. These techniques can reduce both the time and workload for manual annotation. Also, the recognition of low level structure is the basis for which further processing and indexing techniques can be developed. The labelling of low level segments can enable domain specific indexing tools to be enhanced, using prior knowledge of content. Examples include the recognition of pitch markings [8], slow-motion replay detection [14] and exciting event detection [2, 11]. However, unrelated semantic components can contain visually very similar information. It is not uncommon for advertisements to display Sport sequences during televised events to boost marketing appeal of a product. A potential source of error. Audio is a rich, low dimension alternative to visual information that can provide an effective solution to this problem [3].

In this paper we introduce an audio-based event detection algorithm. The main benefit of this approach is the ability to recognise patterns that display high levels of crowd response correlated to key events. The soundtrack from a Soccer game is first parameterised using Mel-frequency Cepstral coefficients. It is then segmented into homogenous components using a windowing algorithm with a decision process based on Bayesian model selection, named BIC-seg. This decision process eliminated the need for defining a heuristic set of rules for segmentation. Each audio segment is then labelled using a series of Hidden Markov model (HMM) classifiers. Each HMM is an optimally selected representation of one of 6 predefined semantic content classes found in Soccer, where those segments labelled into a crowd cheering class are marked as exciting exciting events. Experimentation indicated that the algorithm was more effective for classifying crowd response when compared to traditional model-based segmentation and classification techniques [2].

The remainder of the paper is structured as follows. In Section 2, we introduce the concept of event detection using audio information such as the various content groups that constitute a live match. We also illustrate how these segments can be labelled using HMM classifiers. In Section 3, we outline a windowing scheme for segmenting the audio stream. In Section 4, we evaluate the performance of our system for event detection, concluding our work in Section 5.

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)
1063-6919/04 $ 20.00 IEEE

Code	Class Description	#Samples
CN	Non speech sequences with low to medium levels of crowd sound	2238
SN	Speech sequences with low to medium levels of crowd sound	1053
SC	All crowd chanting sequences with or without speech	438
CC	All crowd cheering sequences related to exciting events	708
MS	National anthems / Music played in the stadium	1734
W	High pitched sounds including referee whistle and signal interfernce	486

Table 1: Redefined audio-based pattern classes and the volume of data from each class, in seconds.

2 Audio-based Indexing

During live Sporting broadcasts microphones are strategically placed at pitch level to recreate the stadium atmosphere. As a result, the soundtrack of a Soccer broadcast is a mixture of speech and vocal crowd reactions alongside other environmental sounds including whistles, drums and clapping. This soundtrack is then mixed with the commentary track to provide an enriched depiction of the action unfolding.

Analysing this crowd activity is important for event detection. Fans inside the stadium react to different stimuli during a match such as a goal or scoring attempt, an exciting passage of play or even a poor refereeing decision. The resulting crowd reaction can be in the form of cheering, shouting, clapping or booing.

In this work, we apply a statistical approach to recognise audio-based patterns related to excited crowd reaction. An increase in crowd response is an important indicator for the occurrence of key events. The automatic recognition of crowd reaction can be achieved by using model based classifiers that represent specific patterns found in the audio stream, such as crowd response. But before delving into the technique, we first examine the sound patterns that constitute a Soccer match.

2.1 Pattern Classes

The audio track of a live Soccer broadcast is a complex and noisy environment. There are many similar sound classes such as crowd cheering and crowd singing that are potential sources of error for an event detection system. For example, we assume crowd cheering to be correlated to exciting key events. Crowd chanting or singing on the other hand, is not directly related to exciting events. During a match it is not unusual for periods of singing from supporters. Usually these periods coincide with the start and end of the game as well as after important events, such as a goal. Singing and chanting can also occur during lulls in the game where supporters vocally encourage their team to improve performance. Distinction between these two sound classes is vital for accurate event detection.

By separating the audio data into well defined groups or pattern classes, we can discriminate between those audio sequences that can be correlated to key events and those that are not. From an earlier study [2], a series of pattern classes were defined. Each class corresponded to the level of crowd sound found in a Soccer soundtrack, ranging from high to low.

Another potential source of error was poor discrimination between speech and non speech audio, where a speech segment corresponding to a commentary sequence from one announcer. It was discovered that some speech sequences can be mistaken as clips containing crowd cheering. So, the pattern classes were further divided into those sequences containing speech and those that did not. Hence, there were 6 pattern classes corresponding to audio sequences with or with out speech with varying levels of crowd response. These pattern classes enable discrimination between sequences that are correlated with key events and those that are not.

From the earlier study, we discovered some potential weaknesses with the original pattern classes [2]. One specific problem was false classification of audio clips that contained unusual sounds that did not belong to one particular group. These sequences included signal interference, stadium announcements, music inside the stadium and the high pitched whistle used by the referee. These audio sounds were usually labelled into the crowd cheering group, causing false event detection. Another problem was due to poor definition between some pattern classes. For example, the cut off between sound levels for various crowd activity was too arbitrary and distinction between pattern classes became blurred. This resulted in the statistical models that represent each class sharing similar sound traits, causing a negative effect on system accuracy. To address these problems, we first redefined the sound classes in [2].

Two new classes were added, Table 1. The first containing high pitched sounds such as the referee whistle and signal interference. The second class representing music. Music is often played inside the stadium often during the start of a match including the national anthems of two competing countries. In some games music is also played as part of a goal celebration. Both these sound classes can be falsely identified as crowd cheering [2].

To avoid overlapping between pattern classes we also re-

defined or grouped existing classes, Table 1. By grouping problem classes, the definition of each content class during the labelling process could be clarified. Alongside poor discrimination between representative classifiers. It was believed that human error during the training data generation phase played a significant part in system accuracy deterioration. Poor distinction between crowd sound levels was reflected in the training data, generated for specific pattern classes, containing overlapping traits.

One major change was that crowd cheering with or without speech was placed into one single class. Speech and non-speech segments during high levels of crowd cheering can be blurred during the high activity of sound and noise during a key event. The change allowed for easier labelling of sequences correlated to event detection. The new crowd cheering class ('CC') contained a mixture of crowd cheering, applause and shouting, triggered by a key incident, Table 1.

We also grouped audio clips that contained crowd chanting or singing into one single content class ('SC'). These audio clips represent periods during a match that contain crowd sounds, such as singing or chanting, not related to a key event. As mentioned previously, it is important for event detection to discriminate between crowd singing and those responses correlated to key moments.

The remaining classes represented speech ('SN') and non-speech segments ('CN') that did not contain high levels of crowd sound. Classification of these speech sequences is important for two reasons. Some speech segments, especially when the commentators are engaged in heated discussion, can be wrongly classified as crowd cheering. Also, for future indexing such as speech transcription, distinction between speech segments with different background audio environments is important in order to improve word error rates [5].

2.2 Feature set

For this study we selected Mel-frequency Cepstral coefficients (MFCC) to parameterise the soundtrack. MFCC coefficients, widely used in the field of speech detection and recognition [10], are specifically designed and proven to characterise speech. MFCC's are well represented by multivariate Gaussian distributions and have been shown to be robust to noise. When applied in a statistical based framework, MFCC's are effective in discriminating between speech and other sound classes such as crowd cheering, music and speech [2, 5]. Hence, our Feature set consisted of 14 uncorrelated MFCC coefficients and the Log Energy [10].

2.3 Hidden Markov Model classifiers

We then modelled the predefined pattern classes identified in Table 1, using continuous density Hidden Markov models (HMM). HMM is an effective tool for modelling time varying processes, belonging to a family of probabilistic graphical models able to capture dynamic properties of audio data [10]. Similar static representations, such as the Gaussian Mixture Model (GMM), do not model the temporal properties of audio data, hence the popularity of HMM in the fields of Speech Recognition [5, 10], temporal data clustering [10] and more recently Video Retrieval [2, 4, 15, 16].

Each HMM is a statistical representation of one pattern class, modelling the data structure and variation found in sequences from that class. Each Markov state captures a specific part of the structure and its variation found in a signal, including differences between speakers or the rise and decay of a crowd cheer sequence. The HMMs are generated using manually labelled sequences from each class, applying the Baum-Welch expectation-maximisation algorithm [10]. New sequences are then classified using the Viterbi decoding algorithm.

2.4 HMM application issues

The current application of HMM in the field of Video Retrieval has been ad hoc. As highlighted in a previous study [3], poor model selection can result in poor classification accuracy. A crucial decision is the selection of an appropriate number of hidden Markov states, where accurate segmentation and classification is dependent on optimal selection. An insufficient number of hidden states will not capture enough detail, such as data structure, variability and common noise, thus loosing vital information required for discrimination between groups. A greater number of hidden states would encapsulate more content, though precise and consistent parameter estimation is often limited by the size and quality of the training data. As the number of parameters increase, so does the number of training samples required for accurate estimation. Larger more enriched models require a greater volume of training data for precise parameter estimation.

A further problem with complex models is overfitting. HMMs, specifically designed to discriminate between content, can become too detailed and begin to mirror nuances found in unrelated groups, deteriorating classification accuracy.

For Video Retrieval, there has been little investigation into model selection and the potential side effects on system performance. In the literature, a common theme is to apply domain knowledge or intuition for HMM model selection. Such application includes shot boundary detection [4], TV genre labelling [15] and 'Play' or 'Break' segmentation [16] for Soccer video. This strategy can be helpful when match-

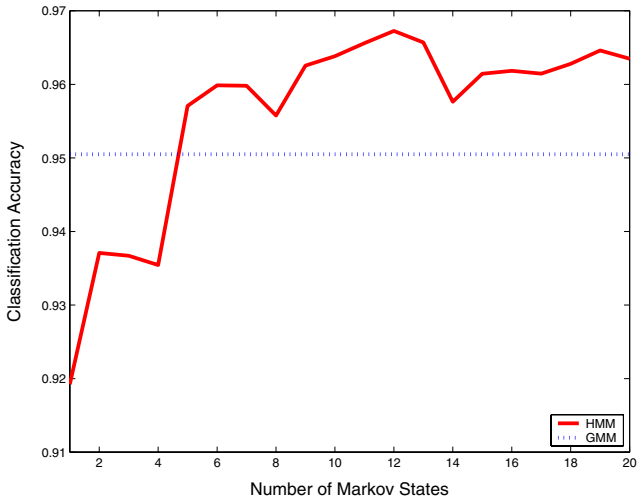

Figure 1: Classification accuracy versus number of hidden Markov states. The best GMM is presented as a baseline.

ing a known number of potential states found in the data, such as shot segmentation [4]. However, there has been little research into how suitable this strategy is when applied to broad content classes found in video. HMM model selection using heuristics can result in simpler frameworks, such as the GMM, becoming a better choice for classification [3]. Thus, more intelligent parameter selection methods for HMM should be investigated.

2.5 HMM model selection

Three selection strategies were compared in an investigation into HMM model selection: an exhaustive search approach, BIC [12], and the Akaike Information Criterion (AIC) [1]. The exhaustive search involved training and testing a range of models until a stopping threshold is reached for the predictive likelihood, the out of sample log-likelihood of a model generating a data sample. The two remaining strategies penalise the predictive likelihood with a penalty term that is derived from the number of parameters in the model. The advantage in predictive likelihood found with more complex models is eventually outweighed by the penalty term, causing a peak for both strategies that we assume is the optimal model.

We evaluated each approach using a series of HMMs modelling content classes generated from live Soccer games. For each class, a series of HMMs with increasing number of states were iteratively implemented. Each model was trained and then the predictive likelihood score was calculated on a separate test sample. From the study, we found the BIC model selection approach to select the simplest HMMs, without affecting classification accuracy.

For this work, we illustrate the advantage of HMM model selection, by investigating what effect increasing the number of hidden Markov states has on classification accuracy, Figure 1. As a comparison, the best run of a GMM classifier was used as a baseline. From Figure 1, the mean classification accuracy gradually increased as the number of hidden states were added to the HMM. After the 5^{th} hidden state was added, the HMM began to outperform the GMM classifier, where on average, a 12 state HMM performed best. A similar trend was followed for the remaining content classes. After a certain number of states were added, the HMM preformed better than the GMM, where the BIC selection strategy consistently selected a HMM that improved over the best GMM run.

In summary, this experiment indicated the importance for suitable model selection, especially given the difficulty and practicality of generating large, varied training sets. Hence, for each content class we used the BIC HMM model selection strategy to generate an optimal classifier for each pattern class in Table 1. The parameters for each optimal HMM classifier was then saved for future labelling of Soccer audio sequences.

3 Audio Segmentation

Given a parameterised audio stream and a series of model-based classifiers, a standard approach to labelling of content is to divide the audio stream into equal length segments. Each segment is then labelled into one of a series of pattern classes. A popular method is to classify individual audio frames, frame by frame, or equal groups of frames, using a maximum likelihood [2, 15] or Dynamic programming (DP) decision process [9, 16]. Segment change is then identified when there is a change in content, where the audio stream moves from one pattern class to another.

Both Wang et al. [15] and Huang et al. [9], classified video programmes into genre using HMM classifiers. The audio stream was divided into equal sized chunks of overlapping audio and then a series of HMMs were used to distinguish between News, Sport and Weather categories. Xie et al [16], applied HMM classifiers to segment Soccer video into 'play' and 'break' segments. The video was divided into segments of three seconds in length. HMMs were then used as a measure to determine the class each segment belonged to.

Applying this modeling framework to the problem of segmentation and classification of audio, provides an elegant solution. However, the HMM is known to have "one principal drawback" when applied to the problem of segmentation [6]. Due to the Markovian property [10], a HMM can be considered a poor representation of the process that generates each pattern class. For a first order HMM, any relationship that occurs between values separated by more

than one time point, must be "communicated via the intervening" values. According to the Markovian assumption, the probability that the HMM will move from one state to another during one time step, is governed only by the transition probability of moving from the state at the previous time point. Because of this limitation, employing a maximum likelihood approach for segmentation is reliant on minimising false classification. A misclassification will be reflected by an incorrect segment change.

Also, it is not uncommon when classifying groups of frames, for more than one content class to be found. A major source of error. Simple smoothing techniques can help limit the effect misclassifications have on event detection [2]. A Dynamic Programming (DP) algorithm can also be applied to find the optimal path through the model likelihoods [9, 16]. This process requires a further training and testing step though and is not addressed in this paper.

Hence, the major draw-back with model-based segmentation and classification algorithms is that they are only as successful as the weakest classifier. A high rate of misclassification between classes can result in poor segmentation. For event detection, it is important to find accurate start and end points for each event. So, we investigated alternatives to model-based segmentation.

One alternative is a sliding search window method, popular for audio-based segmentation during Automatic Speech recognition (ASR). A localised window of audio frames can be employed to find possible segment changes. Applying a window of audio frames provides greater source of evidence for segment change, though one major disadvantage with this approach is the decision process for segment change. Normally an empirically set threshold for boundary change is implemented. Segment change can also be identified using either localised thresholding schemes, generalised likelihood ratio test (GLRT) or a divergence measure on the audio features [5]. However, these techniques still require a predefined cut off point.

Due to the constant change in audio environment during live Soccer sequences, an empirically set threshold may not adapt to changing levels of noise. Measures such as the GLRT or divergence have been shown to be too noisy to threshold [5]. A more robust decision technique is required that can generalise and adjust to the difficult audio environment.

A solution to this problem is to think of the problem as one of model selection. From previous studies [2, 3], we can confidently assume the MFCC coefficients are generated from a multivariate Gaussian distribution. If we assume that each segment is generated from its own multivariate Gaussian distribution, segment change would occur when a change from one Gaussian distribution to another occurs. To identify segment change would be to find when it is optimal to select two Gaussian models over one in a window of values. This can be achieved using the Bayesian Information Criterion (BIC) [12]. The advantage of this approach over other metric based segmentation schemes is that it is threshold free. Hence its ability to generalise.

3.1 The BICseg algorithm

For simplicity, we now call the segmentation algorithm BICseg. The algorithm is an adaptation of the original implementation by Chen et al. [5]. Changes were required to the algorithm when applied to the noisy and difficult environment of Soccer video. Also, the original algorithm suffered from efficiency issues, which we addressed.

Consider modelling the data sequence of audio frames $X = \{x_i : i = 1, \ldots, N\}$ using one model from a set $M = \{M_i = 1, \ldots, K\}$. For any given model M, assume that the $\dim(M)$ parameters are chosen to maximise the likelihood and let $L(X, M)$ denote this maximum value. BIC penalises the model likelihood by the number of parameters in the model (1).

$$BIC_{M_j}(X) = \log L(X, M) - \gamma \frac{dim(M_j)}{2} \log(N) \quad (1)$$

Also, by varying $\gamma > 0$, one can trade off the relative importance and model complexity although $\gamma = 1$ is the strict definition of BIC. To detect segments or boundary changes in an audio sequence using the BIC criterion. We denote $X = \{x_i \in \mathbf{R}^d, i = 1, \ldots, N\}$ as the sequence of MFCC vectors extracted from the entire audio stream. We assume X is drawn from an independent and identically distributed multivariate Gaussian process:

$$X = x_i \sim N(\mu_i, \Sigma_i)$$

where μ_i is the mean vector and Σ_i is the full covariance matrix [1].

We first illustrate how to detect one segment change. We are interested in the hypothesis test for a change occurring at time i:

$$H_0 : x_1 \ldots x_N \sim N(\mu, \Sigma)$$
$$H_1 : x_1 \ldots x_i \sim N(\mu_1, \Sigma_1); \quad x_{i+1} \ldots x_N \sim N(\mu_2, \Sigma_2)$$

Hence, the maximum likelihood ratio statistic is,

$$R(i) = N \log |\Sigma| - N_1 \log |\Sigma_1| - N_2 \log |\Sigma_2| \quad (2)$$

where Σ is the sample covariance matrix for the model of all the data $\{x_1, \ldots, x_N\}$, and Σ_1 and Σ_2 are

[1] To avoid parameter estimation problems with the original algorithm. If the number of audio frames for model estimation is less than or equal the dimension of the model, we restricted the covariance matrices to be diagonal.

the covariances for the two segments $\{x_1, \ldots, x_i\}$ and $\{x_{i+1}, \ldots, x_N\}$. The maximum likelihood estimate of the changing point can then be defined as:

$$\hat{t} = \arg\max_i R(i) \tag{3}$$

We can view this hypothesis test as a model selection problem, where we have two competing models. The first model is two separate multivariate Gaussians. Each Gaussian is a representing of a non-overlapping homogenous segment. The remaining model is one single Gaussian representation of the entire data sequence. Using BIC, the difference between the two models can be expressed as:

$$BIC(i) = R(i) - \gamma P \tag{4}$$

where the likelihood ratio $R(i)$ is defined by (2) and according to BIC. The penalty P can be defined as:

$$P = \frac{1}{2}(d + \frac{d(d+1)}{2}) \log(N) \tag{5}$$

where γ is the penalty weight, N is the size of the decision window and d is the dimension of the feature space.

Hence, the BIC criterion can be seen as a replacement for an empirically set threshold of the log likelihood distance. The new threshold being automatically chosen by (5). If (4) is positive, modeling the data as two separate Gaussian models would be optimal. We can then decide if there is a change when,

$$\{\max_i BIC(i)\} > 0 \tag{6}$$

where the boundary change will be found at frame

$$\hat{t} = \arg\max_i BIC(i) \tag{7}$$

3.1.1 Algorithm Changes

Chen et al. [5] identified problems with the original BICseg algorithm when detecting multiple change points. These included efficiency issues and its tendency to over segment (insertion errors)[2]. The algorithm also missed many boundaries (deletion errors) that were from segments less than 2 seconds in length. As the algorithm uses model selection to identify segment changes, the decision process is dependent on how well each Gaussian model represents its corresponding segment. Parameter estimation for some Gaussian models suffered due to limited data. For example, the BICseg algorithm searches for segment changes in a data sequence $X = \{x_i \in \mathbf{R}^d, i = 1, \ldots, N\}$, using a search window $[left, right]$, where $left = \{1, \ldots, N-1\}$ and $right = \{2, \ldots, N\}$, $right > left$. When the search window is too small, segments were missed because of insufficient data to adequately develop each Gaussian model.

To address this shortcoming, we limited situations when model generation occurred using small amounts of data. First, a limit was placed on the minimum size of the search window. The window would be initialised and following each boundary change thereafter, to length min_{win}. We also placed a limit on searching at the beginning and end of the window. i.e. when searching a window, we did not check for boundaries at the extremes. This would allow for a minimum buffer of audio frames to be used for Gaussian model estimation when searching at the start and end points of the window. That is, $buff_{win}$ seconds from each extreme. Each opposing Gaussian model would then have a minimum number of audio frames ($buff_{win} > 1$) to estimate Σ_1 and Σ_2, as apposed to potentially just one audio frame.

Another problem with the original algorithm was that it was computationally expensive. The search window sequentially expands after each search cycle. A new audio frame is added to the search window until a boundary is found. By expanding the window $[left, right]$, the decision of a boundary change is determined on as much data as possible. However, in practice the algorithm could be linearly searching through a potentially huge window. To avoid this problem, we placed a maximum limit on the size of the search window, i.e. a window could not grow to exceed max_{win} frames.

A final change to the algorithm was to address its efficiency. A major limitation to the original algorithm was that it was computationally expensive. Searching through large volumes of data, such as Soccer video, was time consuming. To address this, more than one frame was added after each boundary search. Instead of sequentially increasing the window, frame by frame, after each boundary search step. The search window would grow (or skip) by $skip > 1$ audio frames.

3.1.2 Detecting multiple change points

Hence, the adapted algorithm for detecting multiple changes in a Gaussian process X is:

1. Initialise the interval $[left, right]$
   $left = 1$; $right = left + min_{win}$

2. Using BIC, search for a segment change in the window
   $[left + buff_{win}, right - buff_{win}]$

3. **if** there is no segment change in the window
   $[lef + buff_{win}, right - buff_{win}]$;

   if $(right - left) \leq max_{win}$

[2] An insertion error is when a segment change is detected that did not exist. A deletion error is when a segment boundary is missed.

γ	Insertion Err	Deletion Err
$\gamma = 1.0$	44.12%	1.75%
$\gamma = 1.1$	32.14%	1.75%
$\gamma = 1.2$	10.94%	1.75%
$\gamma = 1.3$	0.87%	10.53%
$\gamma = 1.4$	0%	22.81%
$\gamma = 1.5$	0%	33.33%
$\gamma = 1.6$	0%	50.88%

Table 2: Evaluating the effect γ has on BICseg accuracy.

$right = right + skip$;

else

$left = left + skip$;
$right = right + skip$;

else

set $\hat{t}$ as a segment change boundary;
$left = \hat{t}$; $right = left + min_{win}$;

end

4. **goto** 2. until **end**

3.1.3 Variable Estimation

As previously highlighted, there are a few variables to the new implementation of the BICseg algorithm to be estimated. The first variables were the minimum and maximum length of the search window $[left, right]$. If the search window is too small, we would not contain sufficient information for parameter estimation of the the Gaussian model(s). If the window is too large, the window could have more than one boundary in the window. Instead of two separate Gaussian models, the window could contain three or more. We did not want the window to become too large for efficiency reasons also.

From examining our test data, we found the minimum segment length to be 0.4 seconds. The largest segment being over 5 seconds in length. The average segment length was approximately 1.1 seconds. Using this as a guide, the minimum window (min_{win}) size was set to 0.5 seconds. The window would be large enough to contain two segments and sufficient discriminatory information for model generation. The maximum window length (max_{win}) was set to be 5 seconds in length. The largest segment length in our data sample. We jumped ($skip$) 0.05 seconds for each new window. Searching at the extremes was restricted in the first and last 0.1 seconds, ($buff_{win}$) of the search window, believed to be the absolute minimum amount of information allowed for adequate estimation.

After defining these algorithm parameters, we investigated an appropriate value for γ, the penalty weight. By changing γ, greater or less emphasis is placed on the model dimensionality. Chen et al. [5], set $\gamma = 1$, which is the strict definition of BIC. Tritschler et al. [13], investigated the algorithm further for speaker segmentation and found $\gamma = 1.3$ produced better algorithm performance. The goal for our segmentation system was to find changes in semantic content. Changes including crowd cheering to music or speech to non-speech segments. The goal was to minimise the problem of over-segmentation that the original algorithm suffered from. But not at the expense of missing important segment boundaries.

To estimate γ, we extracted 20 minutes of audio from 4 Soccer games. The segment boundary locations for each clip were marked. A boundary was identified as a change from one content class to another, Table 2. We then ran the BICseg algorithm, incrementally increasing the parameter γ in 0.1 steps, e.g. $\gamma = \{1.0, \ldots, 1.6\}$. The number of insertion and deletion errors were noted for each parameter value.

We were interested in limiting the number of deletion errors, as insertion errors, we believe, can be corrected during segment classification. From the results, Table 2, we found $\gamma = 1.2$ to maximise the desired balance between insertion and deletion errors. Increasing γ generated further deletion errors, while decreasing the parameter, increased the number of insertion errors.

3.2 Summary

Applying a segmentation scheme first has its advantages over a model-based segmentation approach. For example, over segmentation problems can be addressed by joining wrongly segmented content into one unified segment, during classification. Whereas, employing a model-based scheme alone for both classification and segmentation, will result in segmentation errors occurring during false classification. A model-based segmentation approach requires the dividing of the audio stream into equal sized groups of frames. Many of these groups will contain more than one content class, creating potential classification errors that can only be corrected with further processing or filtering [2, 9, 16].

Once a sequence of new observations has been classified, we can then identify possible key events within the sequence. A key event is likely to occur during periods of high crowd response, i.e. classes 'CC'.

4 Experimental Results

We present the results from two experiments carried out to evaluate our approach to event detection. The first experiment measured the accuracy of each HMM classifier for

Class	CN	MS	SN	SC	CC	W
CN	**81.5**	2.77	1.14	0.68	0	5.35
MS	4.16	**87.02**	0.57	0	0	1.23
SN	4.02	5.54	**89.74**	4.11	1.27	3.5
SC	4.69	3.11	6.84	**95.21**	0	1.85
CC	4.96	0.87	1.42	0	**98.31**	0.41
W	0.67	0.69	0.28	0	0.42	**87.65**

Table 3: Confusion matrix for the HMM classifiers.

all content classes. The second experiment evaluated the overall accuracy of the event detection algorithm on eight new games, unseen by the system. We also compare the algorithm with a model-based classification and segmentation scheme, using an 'event window' decision process for event detection [2].

4.1 HMM Classifier Performance

Using the BIC model selection strategy, a HMM classifier for each content class was trained and then evaluated on two separate manually labelled data samples, generated from 12 soccer broadcasts, see Table 2(a). 75% of the data was used for training and 25% for testing. Audio clips were classified choosing the content class that produced the maximum likelihood score. The performance of each classifier is presented in Table 3.

From the experiment, Table 3, the important classifier for event detection, 'CC', produced a high classification accuracy of 98%. This was strong evidence that we were able to discriminate well between this class and the others, a major weakness in the original method [2]. Although not directly comparable, this provided some evidence that the redefinition of the content classes did help. Importantly, we had clear distinction between those samples that contained crowd chanting or singing, 'SC', and crowd cheering, 'CC', which was vital for accurate event detection. However, almost 5% of general crowd sound, 'CN' was falsely labelled as 'CC', a potential source of false event classification. Interestingly, almost 7% of the 'MS' samples were wrongly classified as 'SC'. The explanation for this was, these samples contained singing during national anthems, which blurred the distinction between tboth classes.

4.2 Event Detection Results

To evaluate the event detection approach, we gathered the official match reports and detailed game statistics for each match, taken from the FIFA 2002 World Cup web-site [7], the world governing body for the sport. Important events were considered to be goals, scoring attempts, cautions or other key incidents highlighted in the match report, forming the ground truth against which our system could be compared. The match reports also indicated approximate time points for each event that aided this process.

Using this information as a guide, a window from the start of the event to the end of the crowd response was created, for each true event. To measure performance, a correctly identified event was determined to be if a classified crowd cheering ('CC') segment overlapped a 'true event window' at any time-point. If there was some overlap between an actual event and a 'CC' segment, a correct detection was noted. If there was no true event during a classified segment, a false detection was noted.

4.2.1 Algorithms

We compared four schemes, the first scheme being BICseg and the remaining three schemes are a model-based segmentation and classification approach, using an "event window" for event detection [2]. For the first algorithm, the audio track was segmented using BICseg, then classified using the HMMs generated in the previous section. An event was declared as a classified 'CC' segment.

For the HMM model-based segmentation algorithm approach, the audio stream is divided into sequential audio clips of 1 second in length, with an overlap of 0.25 seconds between clips. From experimentation, 1 second was found to be the shortest segment length possible that provided sufficient information for classification, Figure 2. A series of audio sequences, ranging in length, were classified using the HMMs. It was found that a shorter segment length than 1 second increased classification error. However, it was thought that a longer segment than 1 second, would risk more than one pattern class found in that segment. Again causing classification problems. Hence the audio stream was divided into overlapping segments of 1 second in length before classification.

Each audio segment was then classified using the HMMs in the previous section. A segment boundary was found when there is a change in content class. For event detection, an "event window" [2] was applied to limit the effect false classification would have on performance. For example, a key event triggers a crowd response that normally lasts longer than 1 second in length. Therefore, a key event is flagged if n sequential audio clips are classified as belonging to the crowd cheering class, 'CC'. Each scheme employed a different length for the "event window". For example, the first scheme (Win1), detected events if only 1 audio clip were classified as 'CC'. Hence no event window filter was applied. The second scheme (Win5), flagged an event if a minimum of 5 sequential audio clips were classified 'CC'. The final approach (Win10), 10 sequential audio clips were required for an event to be flagged.

Approach		BICseg		Win1		Win5		Win10	
File	#Events	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
BRA-CHN	27	1.00	0.71	1.00	0.14	1.00	0.36	1.00	0.56
BRA-TUR	21	0.86	1.00	0.86	0.44	0.86	0.86	0.71	1.00
CRC-BRA	18	1.00	0.78	1.00	0.21	0.94	0.47	0.89	0.67
CRC-TUR	48	0.54	0.54	0.54	0.09	0.54	0.23	0.54	0.35
FRA-SEN	16	0.88	1.00	0.88	0.47	0.88	0.94	0.81	1.00
GER-IRE	48	1.00	0.32	1.00	0.06	1.00	0.15	0.92	0.23
GER-SUAD	29	0.97	0.97	0.97	0.16	0.97	0.44	0.97	0.88
POR-KOR	56	0.98	0.26	0.98	0.05	0.98	0.13	0.95	0.21
Overall	**263**	**0.90**	**0.70**	**0.90**	**0.20**	**0.90**	**0.45**	**0.85**	**0.61**

Table 4: Event Detection Results

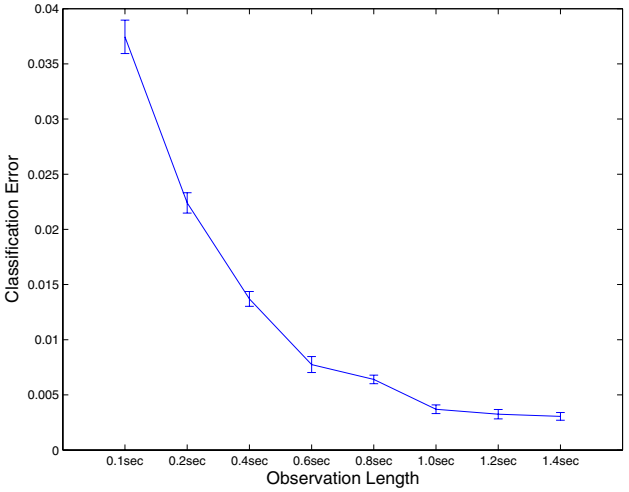

Figure 2: The mean classification accuracy versus audio segment length.

4.2.2 Results

Table 4 presents the results for each technique. We found a high recall for all techniques, with each scheme producing comparable performance. Events that were not picked up by all schemes included, exciting action that did not produce a high level of crowd response in the stadium. This often occurred when a team was supported by a small section of the crowd in the stadium. The small support produced little crowd response, thus the event was not detected by the algorithm.

The BICseg scheme was the most accurate in terms of precision, with the remaining approaches found to over segment. Hence producing a greater frequency of false event detections. Increasing the length of the "event window" improved precision. However, this improvement in precision was followed by a dip in recall for Win10. It was interesting that some false events detected by each algorithm did contain periods of crowd cheering. These false events, were passages of play such as a flamboyant skill by a player or an important defensive play, producing a crowd response not recorded in the truth data. This was exaggerated in games of high importance, such as GER-IRE and POR-KOR. In these games the crowd were extremely vocal in comparison to other matches, thus producing a higher rate of false detections. This was an issue to be addressed during future truth data generation.

5 Conclusions and Future Work

The audio-based event detection approach outlined in this paper, was shown to be effective when applied to Soccer broadcasts. The main benefit of the algorithm is its ability to recognise patterns that indicate high levels of crowd response, correlated to key events. By applying HMM-based classifiers to the problem, we were able to eliminate the need for defining a heuristic set of rules, such as employing and event window to determine event detection.

The performance of the individual HMM-based classifiers was also encouraging given the difficult nature of the Soccer soundtrack. This provided evidence that the redefined pattern classes allowed for greater discrimination between classes, as well as easier training data generation for our HMM classifiers. Another reason for improvement was that, the HMM classifiers were optimised using a BIC model selection strategy. Thus improving the HMM representation for each pattern class.

By segmenting the audio track into these predefined content classes using BICseg. We improved the efficiency of the original algorithm and experimentally set parameters for Soccer video. When applied to the problem of event detection, the accuracy of the event detection algorithm was improved over a model-based segmentation and classification approaches. The BICseg segmentation algorithm did not require thresholding and was able to generalise for the difficult audio environment of Soccer audio. Using the BICseg algorithm allowed for the audio stream to be segmented

into homogenous segments. This limited classification errors further and improved event detection precision when compared to modeled-based segmentation and classification algorithms.

In future research, we plan to improve the maximum likelihood decision process for event detection, by detecting patterns in the content classes to determine event type and importance. For example, it was highlighted that games of high importance produced an exaggerated level of crowd response when compared to other matches. There were also many false events picked up by the system. However, many false event detections did in fact correlate with a crowd response. Human involvement would currently be required to verifying which false detections were key events or not. Thus making the technique semi-automatic. A more intelligent decision process strategy may improve the precision of the event detection algorithm, minimising human involvement in the indexing process. Also, other sources of information such as crowd cheering duration and, the pattern classes that immediately follow the detected event, could be an important indicator for event type and importance. For example, music played after a segment of crowd cheering could be an important clue for event detection.

Future indexing techniques can be aided using this framework, such as speech transcription, another advantage in employing these predefined classifiers. By identifying and differentiating between speech segments that contain a lot of noise or not, word error rate could improve [5]. Finally, other potential improvements to the algorithm include correcting missed events. By integrating visual, textual and motion feature descriptors, we hope to pick up those events that do not correlate with high levels of crowd response. Finally, we also plan to apply this technique to other Sports.

6 Acknowledgements

The authors would like to thank Prof. Keith van Rijsbergen, Prof. Mark Girolami, Vassilis Plachouras, Wilmar Gunnis and both reviewers for their insightful comments.

References

[1] H. Akaike. A new look at the statistical model identification. In *Transactions in Automatic Control*, volume AC-19, pages 716–723. IEEE, Dec 1974.

[2] M. Baillie and J. M. Jose. Audio-based event detection for sports video. In *Proc. 2nd International Conference of Image and Video Retrieval (CIVR'03)*, pages 300–310, Il, USA, July 2003.

[3] M. Baillie, J. M. Jose, and C. J. van Rijsbergen. Hmm model selection issues for soccer video. In *Proc. 3rd International Conference of Image and Video Retrieval (CIVR'04)*, Dublin, EIRE, July 2004.

[4] J. S. Boreczky and L. D. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, pages 3741–3744, Seattle, WA, May 1998.

[5] S. S. Chen, E. M. Eide, M. J. F. Gales, R. A. Gopinath, D. Kanevsky, and P. Olsen. Automatic transcription of broadcast news. *Speech Communication*, 37(1):69–87, 2002.

[6] T. G. Dietterich. Machine learning for sequential data: A review. In T. Caelli, editor, *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[7] FIFA. Official japan-korea world cup 2002 website. Web, March 2002. http://www.fifaworldcup.com (Last visited March 2004).

[8] Y. Gong, T. S. Lim, and H. C. Chua. Automatic parsing of tv soccer programs. *Proc. International Conference on Multimedia Computing and Systems*, pages 167–174, May 1995.

[9] Z. Huang, J. Liu and Y. Wang. Joint video scene segmentation and classification based on hidden markov model. In *Proc. International Conference on Multimedia and Expo (ICME'00)*. IEEE, 2000.

[10] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[11] Y. Rui, A. Gupta, and A. Acero. Proc. automatically extracting highlights for tv baseball programs. In *Proc. ACM Multimedia*, pages 105–115. ACM Press, 2000.

[12] G. Schwarz. Estimating the dimension of a model. In *Annals of Statistics*, volume 6, pages 461–464. 1978.

[13] A. Tritschler and R. A. Gopinath. Improved segmentation and segment clustering using the bayesian information criterion. In *Proc. Eurospeech 1999*, Budapest, Hungary.

[14] P. van Beek, H. Pan, and M. I. Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Salt Lake City, Utah, May 7-11 2001. IEEE.

[15] Y. Wang, Z. Liu, and J. Huang. Multimedia content analysis using both audio and visual clues. In *IEEE Signal Processing Magazine*. IEEE, 2000.

[16] L Xie, S-F Chang, A Divakaran, and H Sun. Structure analysis of soccer video with hidden markov models. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*. IEEE, 2002.