

# A Content-Adaptive Analysis and Representation Framework for Audio Event Discovery from “Unscripted” Multimedia

Regunathan Radhakrishnan,<sup>1</sup> Ajay Divakaran,<sup>1</sup> Ziyou Xiong,<sup>1</sup> and Isao Otsuka<sup>2</sup>

<sup>1</sup> *Mitsubishi Electric Research Laboratory, Cambridge, MA 02139, USA*

<sup>2</sup> *Advanced Technology R&D Center, Mitsubishi Electric Corporation, Hyogo 661-8661, Kyoto, Japan*

Received 1 September 2004; Revised 21 April 2005; Accepted 4 May 2005

We propose a content-adaptive analysis and representation framework to discover events using audio features from “unscripted” multimedia such as sports and surveillance for summarization. The proposed analysis framework performs an inlier/outlier-based temporal segmentation of the content. It is motivated by the observation that “interesting” events in unscripted multimedia occur sparsely in a background of usual or “uninteresting” events. We treat the sequence of low/mid-level features extracted from the audio as a time series and identify subsequences that are outliers. The outlier detection is based on eigenvector analysis of the affinity matrix constructed from statistical models estimated from the subsequences of the time series. We define the confidence measure on each of the detected outliers as the probability that it is an outlier. Then, we establish a relationship between the parameters of the proposed framework and the confidence measure. Furthermore, we use the confidence measure to rank the detected outliers in terms of their departures from the background process. Our experimental results with sequences of low- and mid-level audio features extracted from sports video show that “highlight” events can be extracted effectively as outliers from a background process using the proposed framework. We proceed to show the effectiveness of the proposed framework in bringing out suspicious events from surveillance videos without any a priori knowledge. We show that such temporal segmentation into background and outliers, along with the ranking based on the departure from the background, can be used to generate content summaries of any desired length. Finally, we also show that the proposed framework can be used to systematically select “key audio classes” that are indicative of events of interest in the chosen domain.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

## 1. INTRODUCTION

The goals of multimedia content summarization are twofold. One is to capture the essence of the content in a succinct manner and the other is to provide a top-down access into the content for browsing. Towards achieving these goals, signal processing and statistical learning

tools are used to generate a suitable representation for the content using which summaries can be created. For content that is carefully produced and edited (scripted content) such as news, movie, drama, and so forth, a representation that captures the sequence of semantic units that constitute the content has been shown to be useful. Hence, past work on summarization of scripted content has mainly focussed on coming up with a table of contents (ToC) representation as shown in Figure 1. With such a representation of the detected semantic units, a summary can be constructed using abstractions (e.g., skims, keyframes) from each of the detected semantic units.

The following is a list of approaches towards constructing a hierarchical ToC-like representation for summarization of scripted content.

- (i) News video.
  - (a) Detection of news story boundaries through closed caption or speech transcript analysis [1–3].
  - (b) Detection of news story boundaries using speaker segmentation and face information [4, 5].
- (ii) Situation comedies.
  - (a) Detection of “physical setting” using mosaic representation of a scene [6].
  - (b) Detection of major cast using audio-visual cues [7].
- (iii) Movie content.
  - (a) Detection of syntactic structures like two-speaker dialogs [8].
  - (b) Detection of some specific events like explosions [7].

In unscripted content such as sports and surveillance, interesting events happen sparsely in a background of usual events. Hence, past work on summarization of unscripted

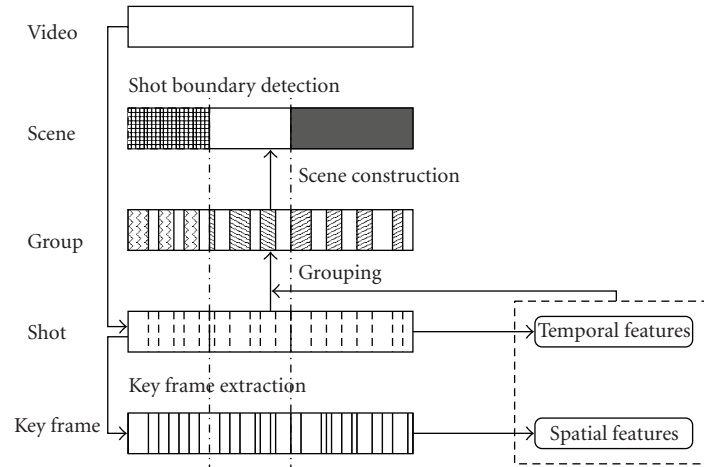


FIGURE 1: A hierarchical video representation for scripted content.

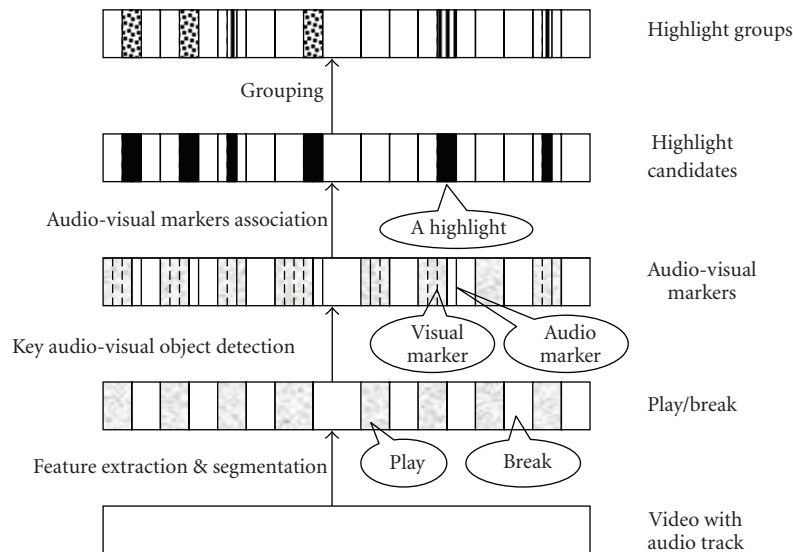


FIGURE 2: A hierarchical video representation for unscripted content.

content has mainly focussed on detecting these specific events of interest.

The following is a list of approaches from literature that detect specific events for summarization of unscripted content.

- (i) Sports video.
  - (a) Detection of domain-specific events and objects that are correlated with highlights using audio-visual cues [9–12].
  - (b) Unsupervised extraction of play-break segments from sports video [13].
- (ii) Surveillance video.
  - (a) Detection of “unusual” events using object segmentation and tracking from video [14].

Based on the detection of such domain-specific key audio-visual objects (audio-visual markers) that are indicative of the “highlight” or “interesting” events, we proposed a hierarchical representation for unscripted content as shown in Figure 2 [15]. The detected events can also be ranked according to a chosen measure which would allow generation of summaries of desired length [16]. In this representation, for each domain the audio-visual markers are chosen manually based on intuition.

For scripted content, the representation framework is based on the detection of the semantic units. Past work has shown that the representation units starting from the “keyframes” up to the “groups” can be detected using unsupervised analysis. However, the highest-level representation unit requires content-specific rules to bridge the gap between semantics and the low/mid-level analysis.

For unscripted content, the representation framework is based on the detection of specific events. Past work has shown that the play/break representation for sports can be achieved by an unsupervised analysis by bringing out repetitive temporal patterns. However, the rest of the representation units require the use of domain knowledge in the form of supervised audio-visual object detectors that are correlated with events of interest. This necessitates a separate analysis framework for each domain in which the key audio-visual objects are chosen based on intuition. However, what is more desirable is a content-adaptive analysis and representation framework that postpone content-specific processing to as late a stage as possible. Then, some challenging questions towards achieving such a framework are as follows.

- (i) Can we come up with a representation framework for unscripted content which requires the use of the domain knowledge only at the last stage as the representation framework for scripted content?
- (ii) Discovery of what kind of patterns would support such a representation framework?
- (iii) Can such a framework help in the systematic choice of the key audio-visual objects for events of interest?

In this paper, the above questions motivate us to propose a content-adaptive analysis framework aimed towards a representation framework for event discovery from unscripted multimedia. We are motivated towards an inlier/outlier-based representation for unscripted multimedia based on the observation that “interesting” events are outliers in a background of usual events. In this paper, we focus on the analysis of audio features for such a representation. We treat the sequence of low-level/mid-level features extracted from the input audio as a time series. Then, we discover subsequences from the input time series that are outliers. The outlier detection is based on eigenvector analysis of the affinity matrix constructed from statistical models estimated from the subsequences of the time series. The detected outliers are ranked based on the deviation from the usual. This results in a temporal segmentation of the input time series, that will henceforth be referred to as “inlier/outlier-based segmentation,” with observations during inliers corresponding to the usual process and observations during outliers corresponding to the unusual events. The analysis thus far is content-adaptive (in the sense that the framework adapts to content statistics to discover the usual and unusual for a given set of parameter choices) and genre-independent, enabling us to come up with a representation for summarization without a priori knowledge. However, since the meaning of “interesting” is dependent on the genre, in order to present an “interesting” summary to the end user, a genre-dependent postprocessing incorporating the domain knowledge can be performed on the discovered outlier subsequences.

The rest of the paper is organized as follows. In the next section, we propose our framework for event discovery using audio features in unscripted content. In Sections 3, 4, and 5, we describe each of the components in the proposed framework in detail. In Section 6, we present the results of the

proposed framework on sports audio content and surveillance audio content. In Section 7, we present our discussion on systematic choice of key audio classes for a chosen domain before presenting our conclusions.

## 2. PROPOSED FRAMEWORK

With the knowledge of the domain of the unscripted content, one can come up with an analysis framework with supervised learning tools for the generation of the hierarchical representation of events in unscripted content for summarization as shown in Figure 2. We propose a content adaptive analysis framework which does not require any a priori knowledge of domain of the unscripted content. It is aimed towards an inlier/outlier-based representation of the content for event discovery and summarization as shown in Figure 3.

We briefly describe the role of each component in the proposed framework as follows.

(i) *Feature extraction*: in this step, low-level features are extracted from the input content in order to generate a time series from which events are to be discovered. For example, the extracted features from the audio stream, could be Mel-frequency cepstral coefficients (MFCC).

(ii) *Classification/clustering*: in this step, the low-level features are classified using supervised models for classes that span the whole domain to generate a discrete time series of mid-level classification/clustering labels. One could also discover events from this sequence of discrete labels. For example, Gaussian mixture models (GMMs) can be used to classify every frame of audio into one of the following five audio classes which span most of the sounds in sports audio: applause, cheering, music, speech, and speech with music. At this level, the input unscripted content is represented by a time series of mid-level classification/cluster labels.

(iii) *Detection of subsequences that are outliers in a time series*: in this step, we detect outlier subsequence from the time series of low-level features or mid-level classification labels motivated by the observation that “interesting” events are unusual events in a background of “uninteresting” happenings. At this level, the input content is represented by a temporal segmentation of the time series into inlier and outlier subsequences. The detected outlier subsequences are illustrated in Figure 3 as  $O_i$ ,  $1 \leq i \leq n$ .

(iv) *Ranking outlier subsequences*: in order to generate summaries of desired length, we rank the detected outliers with respect to a measure of statistical deviation from the inliers. At this level, the input content is represented by a temporal segmentation of the time series into inlier and ranked outlier subsequences. The ranks of detected outlier subsequences are illustrated in Figure 3 as  $r_i$ ,  $1 \leq i \leq n$ .

(v) *Summarization*: detected outlier subsequences are statistically unusual. All unusual events need not be interesting to the end-user. Therefore, with the help of domain knowledge, we prune the outliers to keep only the interesting ones and modify their rank. For example, commercials and highlight events are both unusual events and hence using domain knowledge in the form of a supervised model for audience-reaction sound class will help in getting rid

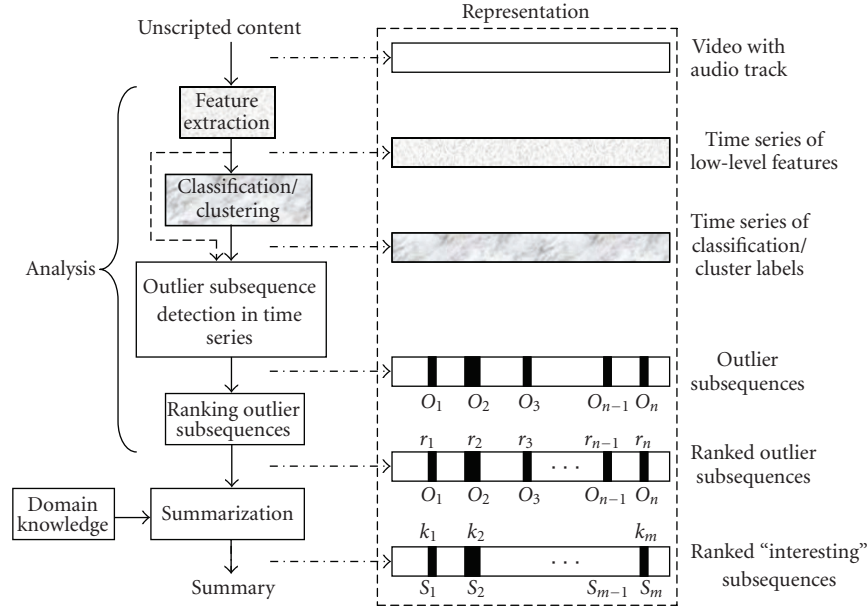


FIGURE 3: Proposed event discovery framework for analysis and representation of unscripted content for summarization.

of commercials from the summary. At this level, the input content is represented by a temporal segmentation of the time series into inlier and ranked “interesting” outlier subsequences. The “interesting” outlier subsequences are illustrated in Figure 3 as  $S_i$ ,  $1 \leq i \leq m$ , with ranks  $k_i$ . The set of “interesting” subsequences ( $S_i$ ’s) is a subset of outlier subsequences ( $O_i$ ’s).

In the following sections, we describe each of these components in detail.

### 3. CLASSIFICATION/CLUSTERING FRAMEWORK FOR MID-LEVEL REPRESENTATION

We extracted low-level features and model the distribution of features for classification into one of the several classes that span the whole domain of unscripted content. We took sports content as an example of unscripted content to explain the classification framework. The following sound classes span almost all of the sounds in sports domain: applause, cheering, music, speech, and speech with music. We have collected 679 audio clips from TV broadcasts of golf, baseball, and soccer games. This database is a subset of that in [17]. Each clip is hand-labeled into one of the five classes as ground truth: applause, cheering, music, speech, and “speech with music.” The corresponding numbers of clips are 105, 82, 185, 168, and 139. The duration of the clips differs from around 1 s to more than 10 s. The total duration is approximately 1 h and 12 min. The audio signals are all monochannel with a sampling rate of 16 kHz. We extracted 12 Mel-frequency cepstral coefficients (MFCC) for every 8 ms frame and logarithm of energy, from all the clips in the training data. We performed classification experiments with varying number of MFCC coefficients and chose 12 as a tradeoff between computational complexity and performance. We trained Gaussian mixture models (GMMs) to model the distribution of

features for each of the sound classes. The number of mixture components were found using the minimum description length principle [16]. Then, given a test clip, we extract the features for every frame and assign a class label corresponding to the sound class model for which the likelihood of the observed features is maximum. For all the experiments to be described in the following sections, we use one of the following time series to discover “interesting” events at different scales:

- (i) the time series of 12 MFCC features and logarithm of energy extracted for every frame of 8 milliseconds;
- (ii) the time series of classification labels for every frame;
- (iii) the time series of classification labels for every second of audio. The most frequent frame label in one second is assigned as the label for that second.

In the following section, we describe the outlier subsequence detection from one of the three time series defined in this section.

### 4. OUTLIER SUBSEQUENCE DETECTION IN TIME SERIES

Outlier subsequence detection is at the heart of the proposed framework and is motivated by the observation that “interesting” events in unscripted multimedia occur sparsely in a background of usual or “uninteresting” events. Some examples of such events are:

- (i) *sports*: a burst of overwhelming audience reaction in the vicinity of a highlight event in a background of commentator’s speech,
- (ii) *surveillance*: a burst of motion and screaming in the vicinity of a suspicious event in a silent or static background.

This motivates us to formulate the problem of discovering “interesting” events in multimedia as that of detecting outlier subsequences or “unusual” events by statistical modeling of a stationary background process in terms of low/mid-level audio-visual features. Note that the background process may be stationary only for small period of time and can change over time. This implies that background modeling has to be performed adaptively throughout the content. It also implies that it may be sufficient to deal with one background process at a time and detect outliers. In the following subsection, we elaborate on this more formally.

#### 4.1. Problem formulation

Let  $p_1$  represent a realization of the “usual” class ( $\mathbf{P}_1$ ) which can be thought of as the background process. Let  $p_2$  represent a realization of the “unusual” class  $\mathbf{P}_2$  which can be thought of as the foreground process. Given any time sequence of observations or low-level audio-visual features from the two classes of events ( $\mathbf{P}_1$  and  $\mathbf{P}_2$ ), such as

$$\cdots p_1 p_1 p_1 p_1 p_1 p_2 p_2 p_1 p_1 p_1 \cdots, \quad (1)$$

then the problem of outlier subsequence detection is that of finding the times of occurrences of realizations of  $\mathbf{P}_2$ .

To begin with, the statistics of the class  $\mathbf{P}_1$  are assumed to be stationary. However, there is no assumption about the class  $\mathbf{P}_2$ . The class  $\mathbf{P}_2$  can even be a collection of a diverse set of random processes. The only requirement is that the number of occurrences of  $\mathbf{P}_2$  is relatively rare compared to the number of occurrences of the dominant class. Note that this formulation is a special case of a more general problem, namely, clustering of a time series in which a single highly dominant process does not necessarily exist. We treat the sequence of low/mid-level audio-visual features extracted from the video as a time series and perform a temporal segmentation to detect transition points and outliers from a sequence of observations.

Before we present our framework for detection of outlier subsequences, we review the related theoretical background on the graph-theoretical approach to clustering.

#### 4.2. Segmentation using eigenvector analysis of affinity matrices

Segmentation using eigenvector analysis has been proposed in [18] for images. This approach to segmentation is related to graph-theoretic formulation of grouping. The set of points in an arbitrary feature space are represented as a weighted undirected graph where the nodes of the graph are points in the feature space and an edge is formed between every pair of nodes. The weight on each edge is the similarity between nodes. Let us denote the similarity between nodes  $i$  and  $j$  as  $w(i, j)$ .

In order to understand the partitioning criterion for the graph, let us consider partitioning it into two groups  $A$  and  $B$  and  $A \cup B = V$ :

$$N_{\text{cut}}(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(A, V)} + \frac{\text{cut}(A, B)}{\text{asso}(B, V)}, \quad (2)$$

where

$$\begin{aligned} \text{cut}(A, B) &= \sum_{i \in A, j \in B} w(i, j), \\ \text{asso}(A, V) &= \sum_{i \in A, j \in V} w(i, j). \end{aligned} \quad (3)$$

Note that  $\text{cut}(A, B)$  measures the total connection from nodes in  $A$  to all the nodes in  $B$ , whereas  $\text{asso}(A, V)$  measures the total connection from nodes in  $A$  to all the nodes in the graph. It has been shown in [18] that minimizing  $N_{\text{cut}}$  minimizes similarity between groups while maximizing association within individual groups. Shi and Malik [18] show that

$$\min_x N_{\text{cut}}(x) = \min_y \frac{y^T(D - W)y}{y^T D y} \quad (4)$$

with the condition that  $y_i$  belongs to  $\{-1, b\}$ . Here  $W$  is a symmetric affinity matrix of size  $N \times N$  (consisting of the similarity between nodes  $i$  and  $j$ ,  $w(i, j)$  as entries) and  $D$  is a diagonal matrix with  $d(i, i) = \sum_j w(i, j)$ .  $x$  and  $y$  are cluster indicator vectors, that is, if  $y(\mathbf{i})$  equals  $-1$ , then feature point “ $\mathbf{i}$ ” belongs to cluster  $A$ , else cluster  $B$ . It has also been shown that the solution to the above equation is same as the solution to the following generalized eigenvalue system if  $y$  is relaxed to take on real value:

$$(D - W)y = \lambda D y. \quad (5)$$

This generalized eigenvalue system is solved by first transforming it into the standard eigenvalue system by substituting  $z = D^{1/2}y$  to get

$$D^{-1/2}(D - W)D^{-1/2}z = \lambda z. \quad (6)$$

It can be verified that  $z_0 = D^{1/2}\vec{1}$  is a trivial solution with eigenvalue equal to 0. The second generalized eigenvector (the smallest nontrivial solution) of this eigenvalue system provides the segmentation that optimizes  $N_{\text{cut}}$  for two clusters. In this paper, we use the term “the cluster indicator vector” interchangeably with “the second generalized eigenvector of the affinity matrix.”

Also, note that although this method of segmentation using eigenvector analysis has been introduced by Shi and Malik, in the context of image segmentation, it also can be used to segment a time series of audio features as we will see later. The key is to compute an affinity from the input times series of audio features in a meaningful way. Thereafter, the nature of the source from which the affinity matrix is computed has no influence on the mathematics.

#### 4.3. Proposed outlier subsequence detection in time series

Given the problem of detecting times of occurrences of  $\mathbf{P}_1$  and  $\mathbf{P}_2$  from a time series of observations from  $\mathbf{P}_1$  and  $\mathbf{P}_2$ , we propose the following time series clustering framework.

- (1) Sample the input time series on a uniform grid. Let each time series sample at index “ $i$ ” (consisting of a

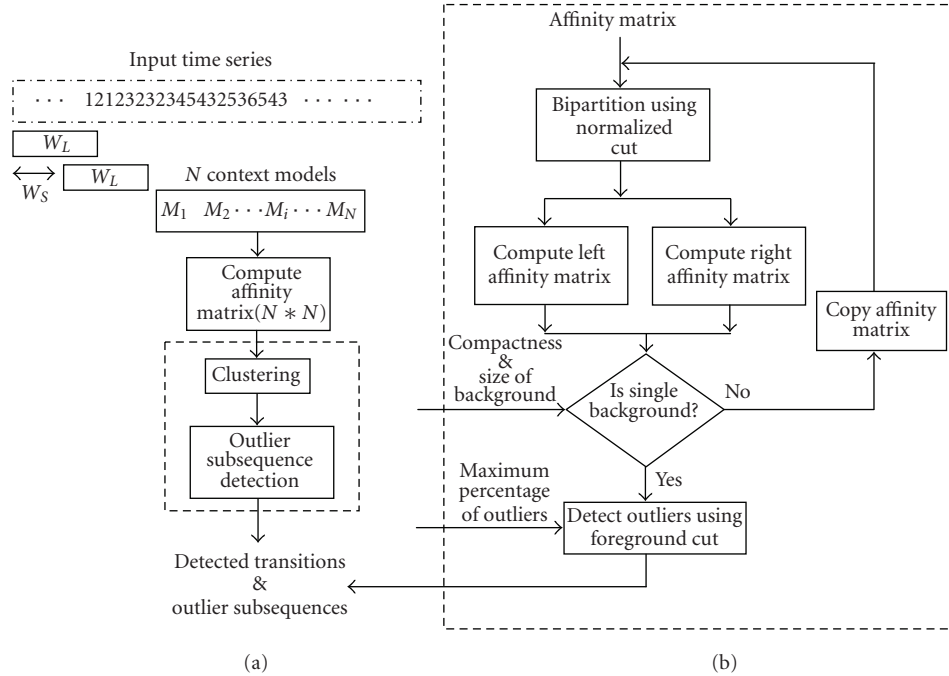


FIGURE 4: Proposed outlier subsequence detection framework.

sequence of observations) be referred to as a context  $C_i$ .

- (2) Compute a statistical model  $M_i$  from the time series observations within each  $C_i$ .
- (3) Compute the affinity matrix for the whole time series using the context models and a commutative distance metric ( $d(i, j)$ ) defined between two context models ( $M_i$  and  $M_j$ ). Each element,  $A(i, j)$ , in the affinity matrix is  $e^{-d(i, j)/2\sigma^2}$ , where  $\sigma$  is a parameter that controls how quickly affinity falls off as distance increases.
- (4) The computed affinity matrix represents an undirected graph where each node is a context model and each edge is weighted by the similarity between the nodes connected by it. Then, we can use a normalized cut solution to identify distinct clusters of context models and “outliers context models” that do not belong to any of the clusters. Note that the second generalized eigenvector of the computed affinity matrix is an approximation to the cluster indicator vector, as discussed in Section 4.2.

Figure 4 illustrates the proposed framework. The portion of Figure 4(b) is a detailed illustration of the two blocks: clustering and outlier detection in Figure 4(a). In this framework, there are two key issues, namely, the statistical model for the context and the choice of the two parameters, the context window size ( $W_L$ ) and the sliding window size ( $W_S$ ) (see Figure 4(a)). The choice of the statistical model for the time series sample in a context would depend on the underlying background process. A simple unconditional probability density function (PDF) estimate would suffice for a memoryless background process. However, if the process has some

memory, the chosen model would have to account for it. For instance, a hidden Markov model (HMM) would provide a first-order approximation.

The choice of the two parameters ( $W_L$  and  $W_S$ ) would be determined by the confidence with which a subsequence is declared to be an outlier. The size of the window  $W_L$  determines the reliability of the statistical model of a context. The size of the sliding factor,  $W_S$ , determines the resolution at which the outlier is detected.

Before we discuss the choice of these parameters, we show some results on synthetic time series data.

#### 4.4. Results with synthetic time series data

In this section, first, we show the effectiveness of the proposed outlier subsequence detection framework using synthetic time series data. Second, we compare the normalized cut with other clustering approaches for outlier subsequence detection from time series.

The synthetic time series generation framework is shown in Figure 5.

In this framework, we have a generative model for both  $P_1$  and  $P_2$  and the dominance of one over the other can also be governed by a probability parameter. It is also possible to control the percentage of observations from  $P_2$  in a given context.

There are four possible scenarios one can consider with the proposed generative model for label sequences.

*Case 1.* Sequence completely generated from  $P_1$ . This case is trivial and less interesting.

*Case 2.* Sequence dominated by observations from  $\mathbf{P}_1$ , that is,  $P(\mathbf{P}_1) \gg P(\mathbf{P}_2)$ . An example for this case is a time series of audio class labels for each second of a news program. Here a burst of music and speech-with-music audio class labels corresponds to commercial messages ( $\mathbf{P}_2$ ) in the recording. The speech background in the news program corresponds to the usual background process,  $\mathbf{P}_1$ .

*Case 3.* Sequence dominated by observations from  $\mathbf{P}_1$ , that is,  $P(\mathbf{P}_1) \gg P(\mathbf{P}_2) \approx P(\mathbf{P}_3) \approx P(\mathbf{P}_4)$ , where  $\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$  are foreground processes. An example for this case is a time series of audio class labels for each second from a sports broadcast. In this case, a burst of audience-reaction audio class labels may correspond to  $\mathbf{P}_2$  and a burst of music audio class labels may correspond to  $\mathbf{P}_3$ .

*Case 4.* Sequence with observations from  $\mathbf{P}_1$  and  $\mathbf{P}_2$  with no single dominant class with a number of foreground processes, that is,  $P(\mathbf{P}_1) \approx P(\mathbf{P}_2)$  and  $(P(\mathbf{P}_1)+P(\mathbf{P}_2)) \gg (P(\mathbf{P}_3)+P(\mathbf{P}_4))$ . An example for this case is a time series of features from a clip that has two different genres, say news and sports.

#### 4.4.1. Performance of the normalized cut for Case 2

In this section, we show the effectiveness of normalized cut for Case 2, that is, when  $P(\mathbf{P}_1) \gg P(\mathbf{P}_2)$ . Without loss of generality, let us consider an input discrete time series with an alphabet of three symbols (1, 2, 3) generated from two HMMs ( $\mathbf{P}_1$  and  $\mathbf{P}_2$ ).

The parameters of  $\mathbf{P}_1$  (the state transition matrix ( $\mathbf{A}$ ), the state observation symbol probability matrix ( $\mathbf{B}$ ), the initial state probability matrix ( $\mathbf{\Pi}$ )) are:

$$\begin{aligned} \mathbf{A}_{\mathbf{P}_1} &= \begin{pmatrix} 0.3069 & 0.0353 & 0.6579 \\ 0.0266 & 0.9449 & 0.0285 \\ 0.5806 & 0.0620 & 0.3573 \end{pmatrix}, \\ \mathbf{B}_{\mathbf{P}_1} &= \begin{pmatrix} 0.6563 & 0.2127 & 0.1310 \\ 0.0614 & 0.0670 & 0.8716 \\ 0.6291 & 0.2407 & 0.1302 \end{pmatrix}, \\ \mathbf{\Pi}_{\mathbf{P}_1} &= (0.1 \quad 0.8 \quad 0.1). \end{aligned} \quad (7)$$

The parameters of  $\mathbf{P}_2$  are

$$\begin{aligned} \mathbf{A}_{\mathbf{P}_2} &= \begin{pmatrix} 0.9533 & 0.0467 \\ 0.2030 & 0.7970 \end{pmatrix}, \\ \mathbf{B}_{\mathbf{P}_2} &= \begin{pmatrix} 0.0300 & 0.8600 & 0.1100 \\ 0.3200 & 0.5500 & 0.1300 \end{pmatrix}, \\ \mathbf{\Pi}_{\mathbf{P}_2} &= (0.8 \quad 0.2). \end{aligned} \quad (8)$$

Then, using the generative model shown in Figure 5 with  $P(\mathbf{P}_1) = 0.8$  and  $P(\mathbf{P}_2) = 0.2$  we generate a discrete time series of symbols as shown in Figure 6(a).

We sample this series uniformly using a window size of  $W_L = 200$  and a step size of  $W_S = 50$ . We use the observations within every context to estimate an HMM with 2 states. Using the distance metric defined below for comparing two HMMs, we compute the distance matrix for the whole time

series. Given two context models ( $\lambda_1$  and  $\lambda_2$ ) with observation sequences  $O_1$  and  $O_2$ , respectively, we define

$$D(\lambda_1, \lambda_2) = \frac{1}{W_L} (\log P(O_1 | \lambda_1) + \log P(O_2 | \lambda_2) - \log P(O_1 | \lambda_2) - \log P(O_2 | \lambda_1)). \quad (9)$$

The computed distance matrix,  $D$ , is normalized to have values between 0 and 1. Then, using a value of  $\sigma = 0.2$ , we compute the affinity matrix,  $\mathbf{A}$ , where  $A(i, j) = e^{-d(i, j)/2\sigma^2}$ . The affinity matrix is shown in Figure 6(b). We compute the second generalized eigenvector of this affinity matrix as a solution to cluster indicator vector. Since the cluster indicator vector does not assume two distinct values, a threshold is applied on the eigenvector values to get the two clusters. In order to compute the optimal threshold, normalized cut value is computed for the partition resulting from each candidate threshold between the range of eigenvector values. The optimal threshold is selected as the threshold at which normalized cut value is minimum as shown in Figure 6(c). The corresponding second generalized vector and its optimal partition is shown in Figure 6(d). The detected outliers are at times of occurrences of  $\mathbf{P}_2$ . Figure 6(e) marks the detected outlier subsequences in the original time series based on normalized cut. It can be observed that the outlier subsequences have been detected successfully without having to set any threshold manually. Also, note that since all outlier subsequences are from the same foreground process ( $\mathbf{P}_2$ ), the normalized cut solution found the outlier subsequences. In general, as we will see later, when the outliers are from more than one foreground process (Case 3), the normalized cut solution may not perform as well. This is because each outlier can be different in its own way and it is not right to emphasize association between the outlier cluster members as normalized cut does.

In the following subsection, we show the performance of other competing clustering approaches for the same task of detecting outlier subsequences using the computed affinity matrix.

#### 4.4.2. Comparison with other clustering approaches for Case 2

After constructing the affinity matrix in step (3), step (4) finds clusters in model space. Instead of using normalized cut solution for clustering, one could use one of the following three methods for clustering.

##### Clustering using alphabet-constrained K-means

Given the pairwise distance matrix and the knowledge of the number of clusters, one can perform top-down clustering based on alphabet-constrained  $k$ -means as follows. Since the clustering operation is performed in model space, the centroid model of a particular cluster of models is not merely the average of the parameters of cluster members. Therefore, the centroid model is constrained to be one of the models and it is that model which has minimum average distance to the cluster members.

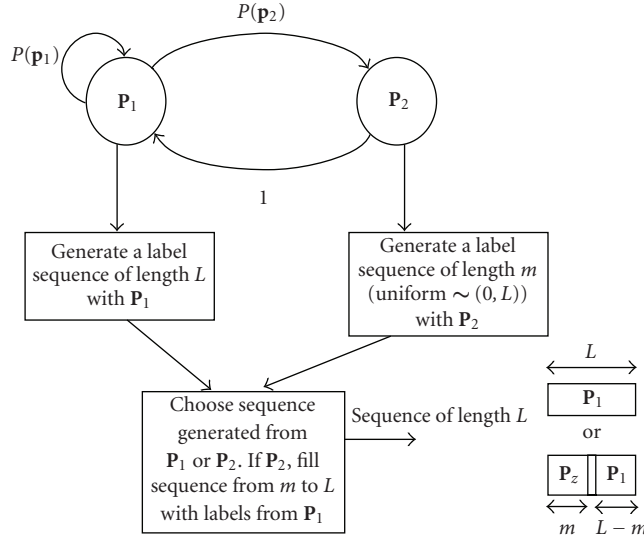


FIGURE 5: Generative model for synthetic time series with one background process and one foreground process.

Given that there is one dominant cluster and the distance matrix, we can use the following algorithm to detect outliers.

- (1) Find the row in the distance matrix for which the average distance is minimum. This is the centroid model.
- (2) Find the semi-Hausdorff distance between the centroid model and the cluster members. The semi-Hausdorff distance, in this case, is simply the maximum of all the distances computed between the centroid model and the cluster members. Hence, semi-Hausdorff distance would be much larger than the average distance if there are any outliers in the cluster members.
- (3) Remove the farthest model and repeat step (2) until the difference between average distance and Hausdorff distance is less than a chosen threshold.
- (4) The remaining cluster members constitute the inlier models.

For more than one cluster, repeat steps (1)–(3) on the complementary set which does not include members of the detected cluster. For more details on alphabet-constrained  $k$ -means, please see [19]. Figure 7(a) shows the distance matrix values of the row that is corresponding to the centroid row. By using a threshold on the difference between average distance and Hausdorff distance, we detect outlier subsequences as shown in Figure 7(b).

#### Clustering based on dendrogram

Given the pairwise distance matrix one can perform a bottom-up agglomerative clustering. At the start, each point is considered to be an individual cluster. By merging two closest clusters at every level until there is only one cluster, a dendrogram can be constructed as shown in Figure 8(a). Then, by partitioning this dendrogram at a particular height, one can get the individual clusters. The criteria for evaluating

a partition could be similar to what normalized cut tries to optimize. There are several choices for creating partitions in the dendrogram and one has to exhaustively compute the objective function value for each partition and choose the one that is optimal. For more details on dendrogram-based agglomerative clustering, please see [20]. For example, by manually selecting a threshold of 5.5 for the height, we can detect outlier subsequences as shown in Figure 8(b). As can be seen from the figure, there are some false alarms and misses in the detected outlier subsequences as the threshold was chosen manually.

#### Clustering based on factorization of the affinity matrix

As mentioned earlier, minimizing  $N_{\text{cut}}$  minimizes similarity between groups while maximizing association within the individual groups. Perona and Freeman modified the objective function of the normalized cut to discover a “salient” foreground object from an unstructured background. Since the background is assumed to be unstructured, the objective function of normalized cut was modified as follows:

$$N_{\text{cut}}^*(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(A, V)}, \quad (10)$$

where cluster  $A$  is the foreground and cluster  $B$  is the background. Note that the objective function only emphasizes the compactness of foreground cluster while minimizing similarity between cluster  $A$  and cluster  $B$ . Perona and Freeman solved this optimization problem by setting up the problem in the same way as in the normalized cut. The steps of the resulting “foreground cut” algorithm is as follows [21].

- (i) Calculate the left singular matrix  $\mathbf{U}$  of the affinity matrix  $\mathbf{A}$ . The singular value decomposition (SVD) of  $\mathbf{A}$  can be written as  $\mathbf{USV}$  where  $\mathbf{U}$  is the left singular matrix,  $\mathbf{S}$  is a diagonal matrix whose elements are the singular values of  $\mathbf{A}$ , and  $\mathbf{V}$  is the right singular matrix.



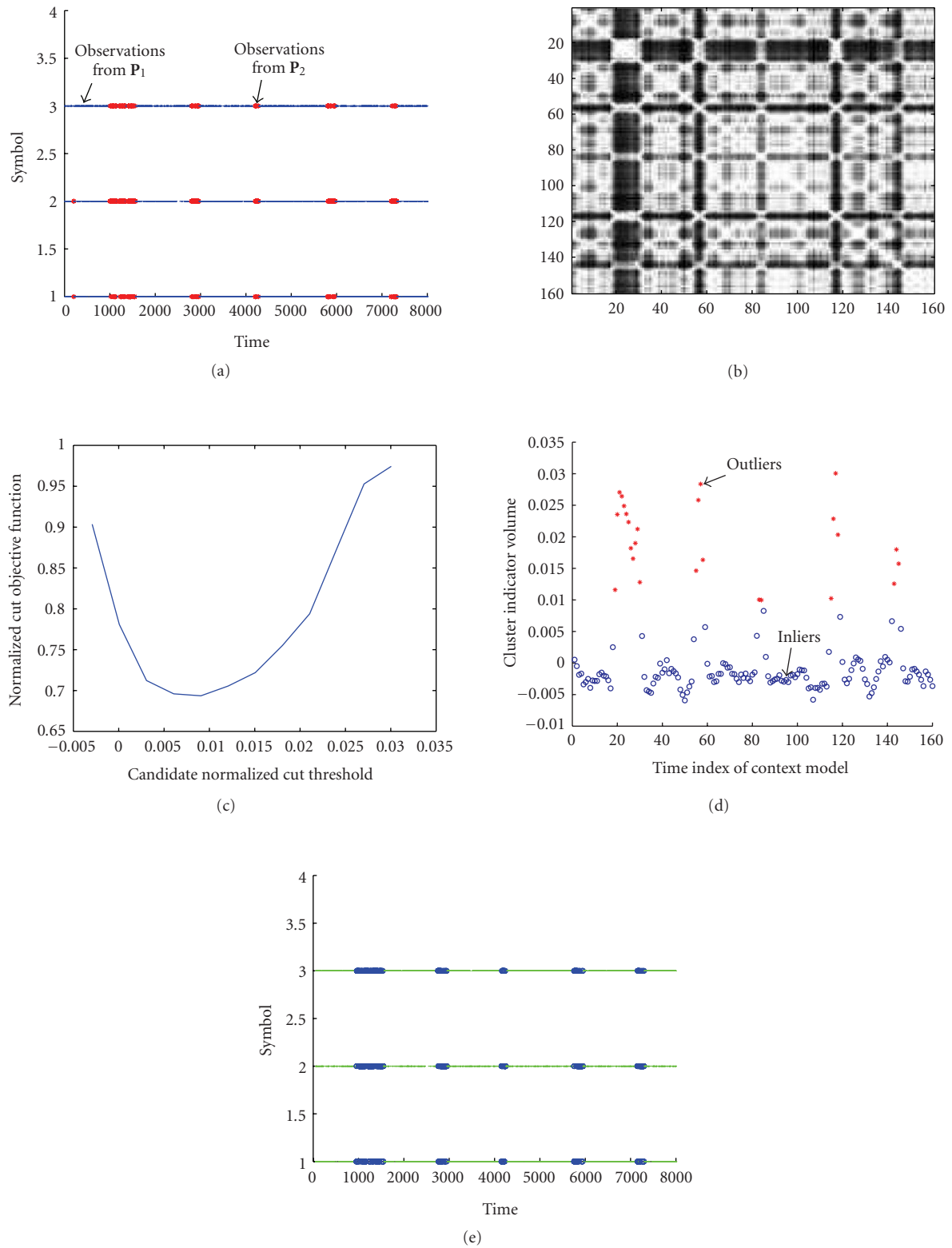


FIGURE 6: Performance of normalized cut on synthetic time series for Case 2. (a) Input time series of discrete labels. (b) The affinity matrix  $A$ . (c) Normalized cut value. (d) Second generalized eigenvector. (e) Outlier subsequence detection using normalized cut.

- (ii) Compute the vector  $\mathbf{u} = \mathbf{S}\mathbf{U}\mathbf{1}$  where  $\mathbf{1}$  is a column vector of ones.
- (iii) Determine the index  $k$  of the maximum entry of  $\mathbf{u}$ .
- (iv) Define the foreground vector  $\mathbf{x}$  as the  $k$ th column of  $\mathbf{U}$ .

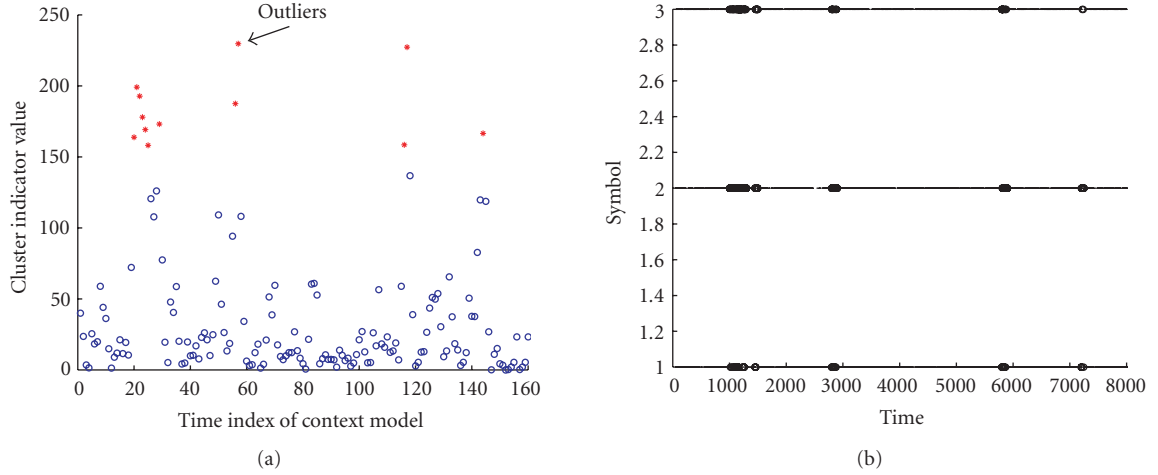
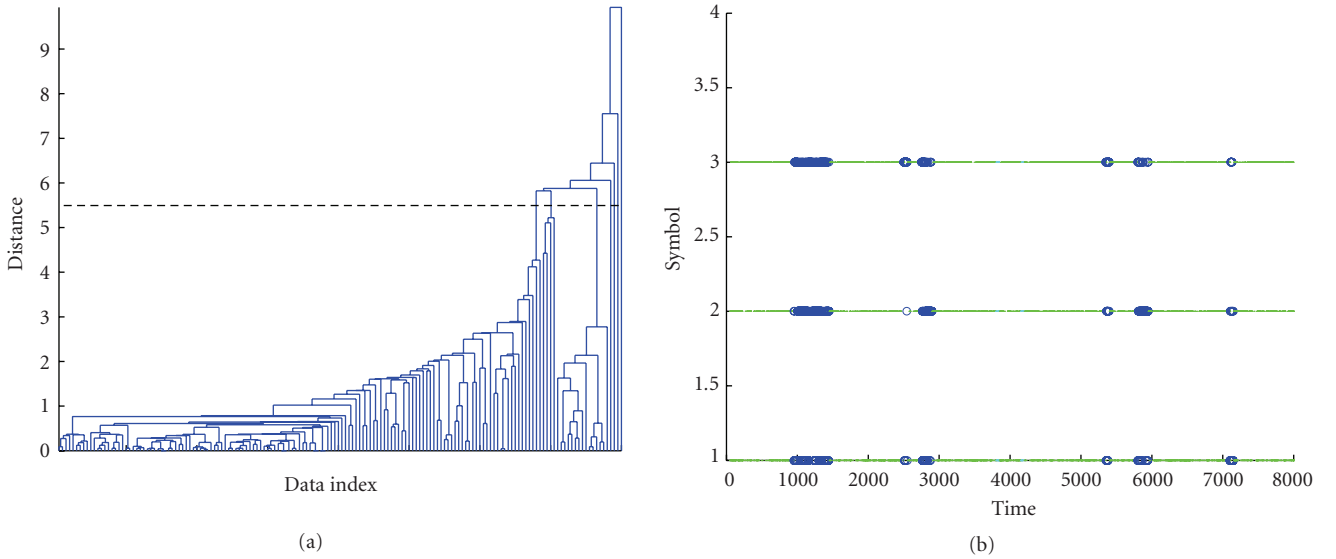
FIGURE 7: Performance of  $K$ -means on synthetic time series for Case 2.

FIGURE 8: Performance of dendrogram cut on synthetic time series for Case 2. (a) Dendrogram. (b) Outlier subsequence detection using the dendrogram.

- (v) Threshold  $\mathbf{x}$ , to obtain the foreground and background.  $\mathbf{x}$  is similar to the cluster indicator vector in normalized cut.

The threshold in the last step can be obtained in the same way as it was obtained for normalized cut.

For the problem at our hand, the situation is reversed, that is, the background is structured while the foreground can be unstructured. Therefore, the same “foreground cut” solution should apply as the modified objective function is

$$N_{\text{cut}}^{**}(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(B, V)}. \quad (11)$$

However, a careful examination of the modified objective function would reveal that the term in the denominator  $\text{asso}(B, V)$  would not be affected drastically by changing

the cluster members of  $A$ . This is because the background cluster is assumed to be dominant. Hence, minimizing this objective function would be the same as minimizing the value  $\text{cut}(A, B)$ . Minimizing  $\text{cut}(A, B)$  alone is notorious for producing isolated small clusters. Our experiments with the synthetic time series data also support these observations. Figure 9(a) shows the value of the objective function  $\text{cut}(A, B)$  for candidate threshold values in the range of values of the vector  $\mathbf{x}$ . Figure 9(b) shows the value of the objective function  $N_{\text{cut}}^{**}(A, B)$  for the same candidate threshold values. Figures 9(c) and 9(d) show the detected outlier subsequences for the optimal threshold. There are some misses because the modified normalized cut finds isolated small clusters. Note that this procedure could be repeated recursively on the detected background until some stopping criterion

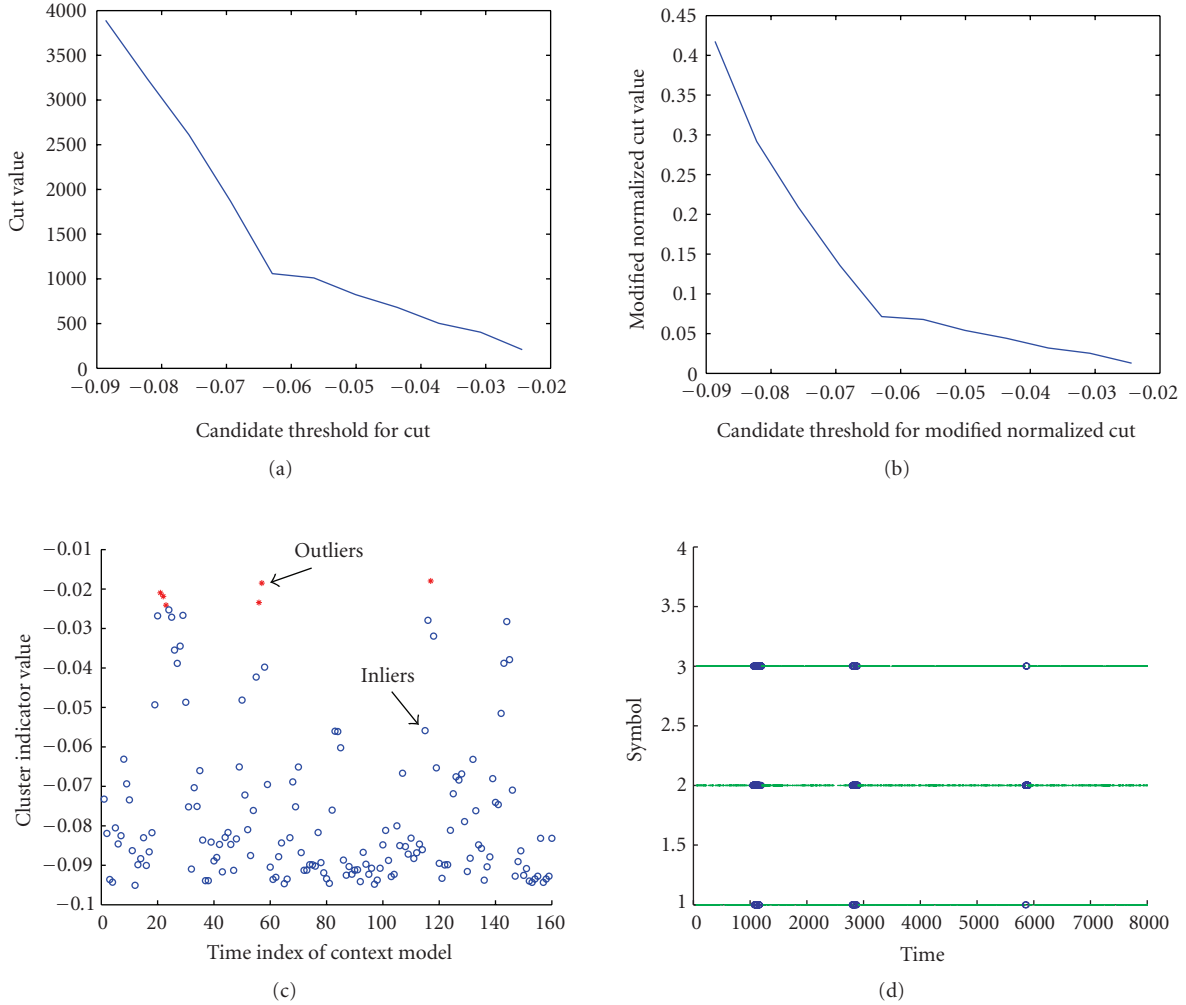


FIGURE 9: Performance of modified normalized cut on synthetic time series for Case 2. (a) The value of the objective function cut for candidate threshold values. (b) Modified normalized cut (affinity matrix factorization). (c), (d) Outlier subsequence detection based on affinity matrix factorization.

is met. For example, the stopping criterion could either be based on percentage of foreground points or based on the radius of the background cluster.

As shown in this section, all of the competing clustering approaches need a threshold to be set for detecting outlier subsequences. The alphabet-constrained  $k$ -means algorithm needs the knowledge of the number of the clusters and a threshold on the difference between the average distance and the semi-Hausdorff distance. The dendrogram-based agglomerative clustering algorithm needs a suitable objective function to evaluate and select the partitions. The foreground cut (modified normalized cut) algorithm finds small isolated clusters and can be recursively repeated on the background until the radius of the background cluster is smaller than a chosen threshold. Therefore, for the case of a single dominant process with outlier subsequences from a single foreground process, the normalized cut outperforms other clustering approaches.

In the following section, we consider the next case where there can be multiple foreground processes generating observations against a single dominant background process.

#### 4.4.3. Performance of normalized cut for Case 3

The input time series for Case 3 is generated using a single dominant background process  $P_1$  and three different foreground processes ( $P_2, P_3, P_4$ ) and  $P(P_1) \gg P(P_2) + P(P_3) + P(P_4)$ .  $P(P_1)$  was set to be 0.8 as in Case 2. Figure 10(a) shows the input time series. As mentioned earlier, since normalized cut emphasizes the association between the cluster members for the two clusters resulting from the partition, there are false alarms from the process  $P_1$  in the cluster containing outliers. Figure 10(b) shows the normalized cut value for candidate threshold values. There are two minima in the objective function but the global minimum corresponds to the threshold that results in an outlier cluster with false alarms. Figure 10(c) shows the partition corresponding to the global

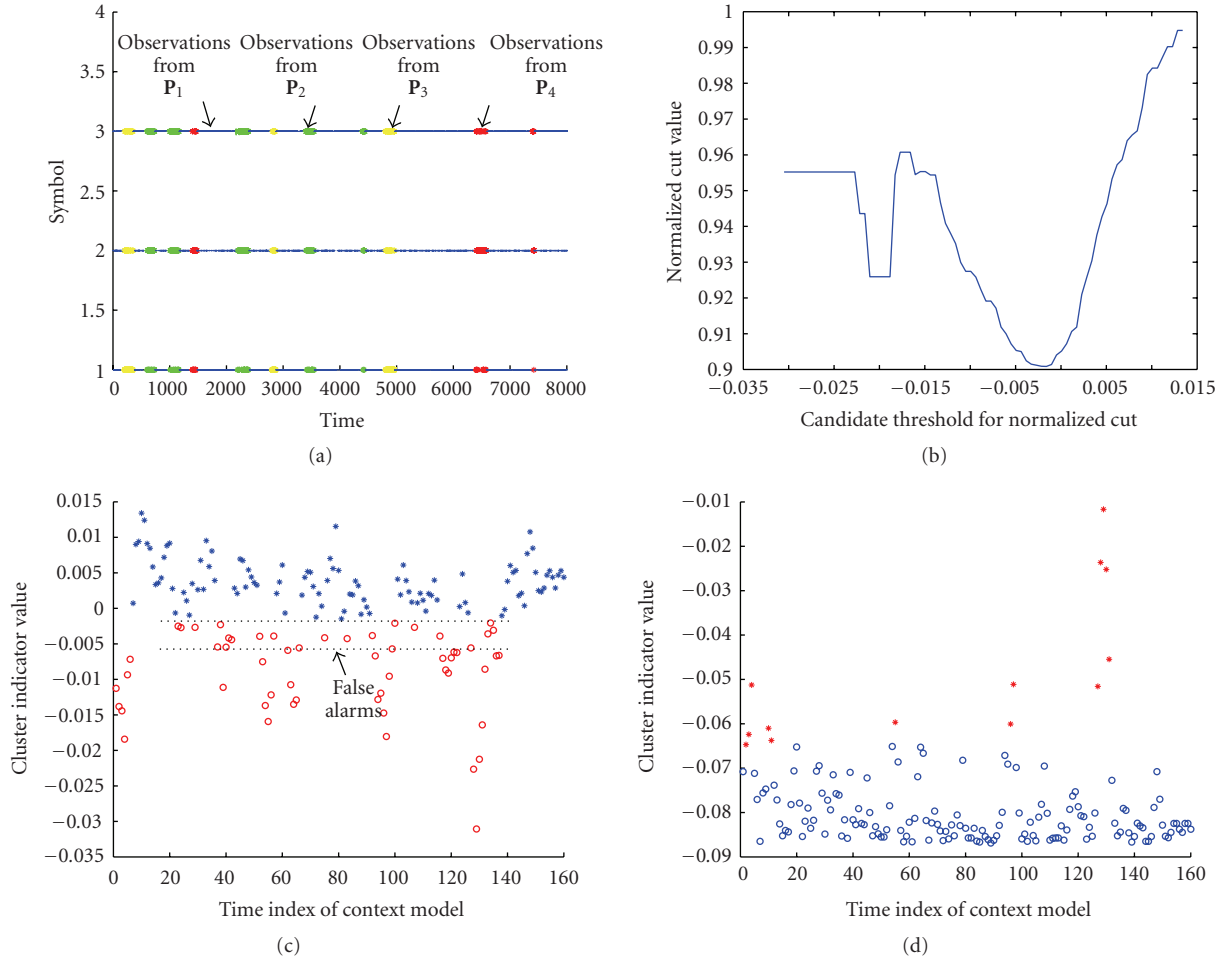


FIGURE 10: Performance comparison of normalized cut and modified normalized cut on synthetic time series for Case 3. (a) The input time series. (b) The normalized cut value for candidate threshold values. (c) The partition corresponding to the global minimum threshold. (d) Modified normalized cut (foreground cut) applied to the same input series.

minimum threshold. On the other hand, when modified normalized cut (foreground cut) is applied to the same input time series, it detects the outliers without any false alarms as shown in Figure 10(d) as the objective function does not emphasize association between the foreground processes.

#### 4.4.4. Hierarchical clustering using normalized cut for Case 4

From the experiments on synthetic time series for Cases 2 and 3, we can make the following observations.

- (i) The normalized cut solution is good for detecting distinct time series clusters (backgrounds) as the threshold for partitioning is selected automatically.
- (ii) The foreground cut solution is good for detecting outlier subsequences from different foreground processes that occur against a single background.

Both of these observations lead us to a hybrid solution which uses both normalized cut and foreground cut

for handling the more general situation in Case 4. In Case 4, there is no single dominant background process and the outlier subsequences are from different foreground processes. Figure 11(a) shows the input time series for Case 4. There are two background processes and three foreground processes.

Given this input time series and the specifications of a single background in terms of its “compactness” and “size relative to the whole time series” and the maximum percentage of outlier subsequences, we use the following algorithm to detect outlier subsequences.

- (1) Use normalized cut recursively to first identify all individual background processes. The decision of whether or not to split a partition further can be automatically determined by computing the stability of normalized cut as suggested in [18] or according to the “compactness” and “size” constraint.
- (2) From the detected distinct backgrounds in step (1), use foreground cut recursively to detect the outlier subsequences while making sure that the detected percentage of outliers does not exceed the specified limit.

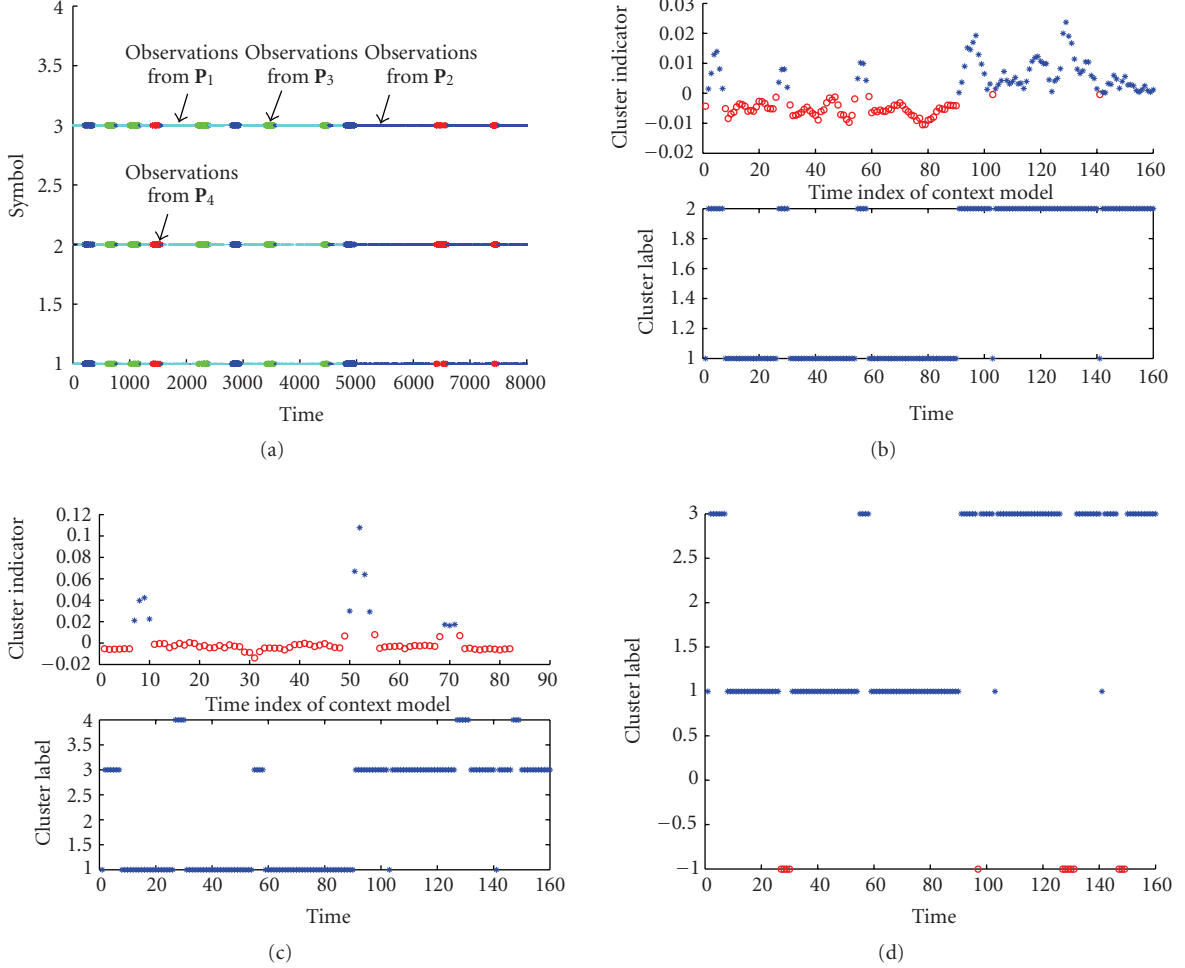


FIGURE 11: Performance of hybrid (normalized cut and foreground cut) approach on synthetic time series for Case 4. (a) The input time series. (b) Top: result of normalized cut (root partition, 1 minute,  $N_{cut} = 8.275415e - 001$ ); bottom: corresponding temporal segmentation (left radius=0.240746, right radius=1.414331). (c) Top: result of normalized cut (p1-1 partition, 2 minutes,  $N_{cut} = 8.439419e - 001$ ); bottom: corresponding temporal segmentation (left radius=0.663607, right radius=0.594490). (d) Final detected outlier subsequences using foreground cut on individual background clusters.

The “compactness” of a cluster can be specified by computing its radius using the pairwise affinity matrix as given below:

$$r = \max_{1 \leq i \leq N} \left( A(i, i) - \left( \frac{2}{N} \sum_{j=1}^N A(i, j) \right) + \left( \frac{1}{N^2} \sum_{k=1}^N \sum_{j=1}^N A(k, j) \right) \right). \quad (12)$$

Here  $N$  represents the number of cluster members. The first term represents the self affinity and is equal to 1. The second term represents the average affinity of the  $i$ th cluster member with others and the last term is average affinity between all the members of the cluster. The computed value of  $r$  is guaranteed to be between 0 and 1.

For this input time series, we specified the following parameters: compactness of the background in terms of its radius  $\leq 0.5$ , relative size of background with respect to the size of whole time series  $\geq 0.35$ , and maximum outlier percentage was set to 20%. Figure 11(b) and 11(c) show the result

of normalized cut and the corresponding temporal segmentation of the input time series. Figure 11(d) shows the final detected outlier subsequences using foreground cut on individual background clusters.

Now that we have shown the effectiveness of outlier subsequence detection on synthetic time series, we will show its performance on the time series obtained from audio data of sports and surveillance content in the experimental results section. In the following section, we analyze how the size of window used for estimating a context model ( $W_L$ ) determines the confidence on the detected outlier. The confidence measure is then used to rank the detected outliers.

## 5. RANKING OUTLIERS FOR SUMMARIZATION

In this section, first, we show that the confidence on the detected outlier subsequences is dependent on the size of  $W_L$ . Second, we use the confidence metric to rank the outlier subsequences.

Recall that in the proposed outlier subsequence detection framework, we sample the input time series on a uniform grid of size  $W_L$  and estimate the parameters of the background process from the observations within  $W_L$ . Then, we measure how different it is from other context models. The difference is caused either by the observations from  $\mathbf{P}_2$  within  $W_L$  or by the variance of the estimate of the background model. If the observed difference between two context models is “significantly higher than allowed” by the variance of the estimate itself, then we are “somewhat confident” that it was due to the corruption of one of the contexts with observations from  $\mathbf{P}_2$ .

In the following, before we quantify what is “significantly higher than allowed” and what is “somewhat confident” in terms  $W_L$  for two types of background models that we will be dealing with, we will review kernel density estimation.

### 5.1. Kernel density estimation

Given a random sample  $x_1, x_2, \dots, x_n$  of  $n$  observations of  $d$ -dimensional vectors from some unknown density ( $f$ ) and a kernel ( $K$ ), an estimate for the true density can be obtained

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \quad (13)$$

where  $h$  is the bandwidth parameter. If we use the mean squared error (MSE) as a measure of efficiency of the density estimate, the tradeoff between bias and variance of the estimate can be seen as shown below:

$$\text{MSE} = E[\hat{f}(x) - f(x)]^2 = \text{Var}(\hat{f}(x)) + [\text{Bias}(\hat{f}(x))]^2. \quad (14)$$

It has been shown in [22] that the bias is proportional to  $h^2$  and the variance is proportional to  $n^{-1}h^{-d}$ . Thus, for a fixed bandwidth estimator one needs to choose a value of  $h$  that achieves the optimal tradeoff. We use a data-driven bandwidth selection algorithm proposed in [23] for the estimation. The proposed scheme uses the plug-in rule and has been shown to be superior to other approaches for fixed bandwidth estimation. For details on the plug-in rule, please see the appendix of [24].

### 5.2. Confidence measure for outliers with binomial and multinomial PDF models for the contexts

For the background process to be modeled by a binomial or multinomial PDF, the observations have to be discrete. Without loss of generality, let us represent the set of 5 discrete labels (the alphabet of  $\mathbf{P}_1$  and  $\mathbf{P}_2$ ) by  $S = \{A, B, C, D, E\}$ . Given a context consisting of  $W_L$  observations from  $S$ , we can estimate the probability of each of the symbols in  $S$  using the relative frequency definition of probability.

Let us represent the unbiased estimator for probability of the symbol  $A$  as  $\hat{p}_A$ .  $\hat{p}_A$  is a binomial random variable but can be approximated by a Gaussian random variable with mean as  $p_A$  and variance as  $\sqrt{p_A(1 - p_A)/W_L}$  when  $W_L \geq 30$ .

As mentioned earlier, in the proposed framework we are interested in knowing the confidence interval of the random

variable,  $d$ , which measures the difference between two estimates of context models. For mathematical tractability, let us consider the Euclidean distance metric between two PDFs, even though it is only a monotonic approximation to a rigorous measure such as the Kullback-Leibler distance:

$$d = \sum_{i \in S} (\hat{p}_{i,1} - \hat{p}_{i,2})^2. \quad (15)$$

Here  $\hat{p}_{i,1}$  and  $\hat{p}_{i,2}$  represent the estimates for the probability of  $i$ th symbol from two different contexts of size  $W_L$ . Since  $\hat{p}_{i,1}$  and  $\hat{p}_{i,2}$  are both Gaussian random variables,  $d$  is a  $\chi^2$  random variable with  $n$  degrees of the freedom where  $n$  is the cardinality of the set  $S$ .

Now, we can assert with certain probability,

$$P_c = \int_L^U f_{\chi_n^2}(x) dx, \quad (16)$$

that any estimate of  $d$  ( $\hat{d}$ ) lies in the interval  $[L, U]$ . In other words, we can be  $P_c$  confident that the difference between two context model estimates outside this interval was caused by the occurrence of  $\mathbf{P}_2$  in one of the contexts. Also, we can rank all the outliers using the probability density function of  $d$ .

To verify the above analysis, we generated two contexts of size  $W_L$  from a known binomial or multinomial PDF (assumed to be the background process). Let us represent the models estimated from these two contexts by  $M_1$  and  $M_2$ , respectively. Then, we use Bootstrapping and kernel density estimation to verify the analysis on PDF of  $d$  as shown below.

- (1) Generate  $W_L$  symbols from  $M_1$  and  $M_2$ .
- (2) Reestimate the model parameters ( $\hat{p}_{i,1}$  and  $\hat{p}_{i,2}$ ) based on the generated data and compute the chosen distance metric ( $d$ ) for comparing two context models.
- (3) Repeat steps (1) and (2)  $N$  times.
- (4) Use kernel density estimation to get the PDF of  $d$ ,  $\hat{p}_{i,1}$ , and  $\hat{p}_{i,2}$ .

Figure 12(a) shows the estimated PDFs for binomial model parameters for two contexts of the same size ( $W_L$ ). It can be observed that  $\hat{p}_{i,1}$  and  $\hat{p}_{i,2}$  are Gaussian random variables in accordance with Demoivre-Laplace theorem [25]. Figure 12(b) shows estimated PDFs of the defined distance metric for different context sizes. One can make the following two observations:

- (i) the PDF of the distance metric is  $\chi^2$  with two degrees of freedom in accordance with our analysis;
- (ii) the variance of the distance metric decreases as the number of observations within the context increases from 100 to 600.

Figure 12(c) shows the PDF estimates for the case of multinomial PDF as a context model with different context sizes ( $W_L$ ). Here, the PDF estimate for the distance metric is  $\chi^2$  with 4 degrees of freedom which is consistent with the number of symbols in the used multinomial PDF model.

These experiments show the dependence of the PDF estimate of the distance metric on the context size  $W_L$ . Hence

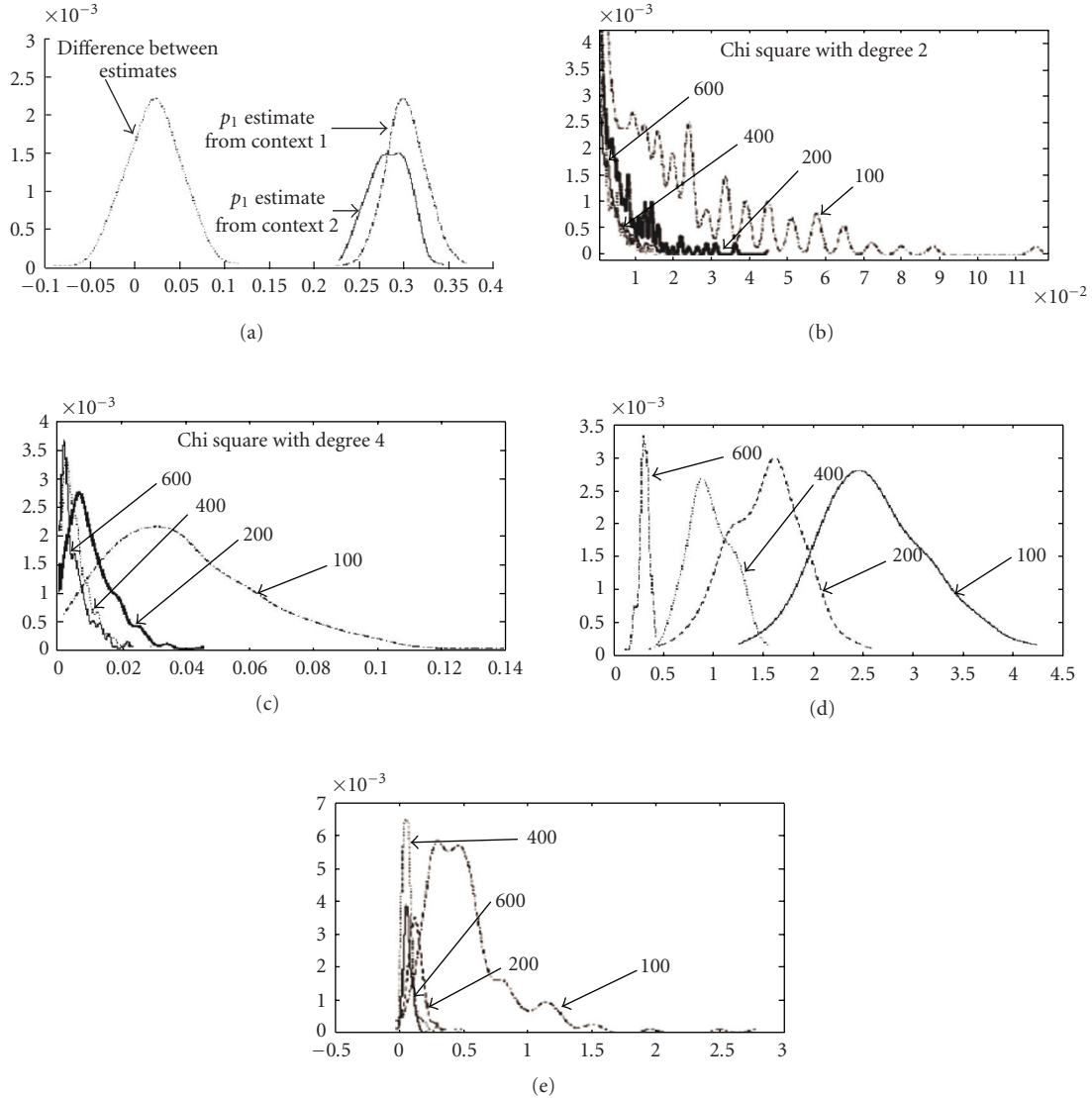


FIGURE 12: PDFs of distance metrics for different background models. (a) PDF of an estimate of a context model parameter (context size of 400 symbols), (b) PDF of distances for a binomial context model for different context sizes, (c) PDF of distances for a multinomial context model for different context sizes, (d) PDF of distances for a GMM as a context model, and (e) PDF of distances for an HMM as a context model. X-axis for value of the random variable, Y-axis for probability density.

for a chosen  $W_L$ , one can compute the PDF of the distance metric, and any outlier caused by the occurrence of symbols from another process ( $\mathbf{P}_2$ ) would result in a sample from the tail of this PDF. This would let us quantify the “unusualness” of an outlier in terms of its cumulative distribution function (CDF) value.

In the next subsection, we perform a similar analysis for HMMs and GMMs as context models.

**5.3. Confidence measure for outliers with GMM and HMM models for the contexts**

When the observations of the memoryless background process are not discrete, one would model its PDF using a Gaus-

sian mixture model (GMM). If the process has first-order memory, one would model its first-order PDF using a hidden Markov model (HMM). Let  $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$  represent the model parameters for both the HMM and GMM, where  $\mathbf{A}$  is the state transition matrix,  $\mathbf{B}$  is the observation symbol probability distribution, and  $\pi$  is the initial state distribution. For a GMM,  $\mathbf{A}$  and  $\pi$  are simply equal to 1 and  $\mathbf{B}$  represents the mixture model for the distribution. For an HMM with continuous observations,  $B$  is a mixture model in each of the states. For an HMM with discrete labels as observations,  $B$  is a multinomial PDF in each of the states. Two models (HMMs/GMMs) that have different parameters can be statistically equivalent [26] and hence the following distance measure is used to compare two context models

( $\lambda_1$  and  $\lambda_2$  with observation sequences  $O_1$  and  $O_2$ , resp.):

$$D(\lambda_1, \lambda_2) = \frac{1}{W_L} (\log P(O_1 | \lambda_1) + \log P(O_2 | \lambda_2) - \log P(O_1 | \lambda_2) - \log P(O_2 | \lambda_1)). \quad (17)$$

The first two terms in the distance metric measure the likelihood of training data given the estimated models. The last two cross-terms measure the likelihood of observing  $O_2$  under  $\lambda_1$  and vice versa. If the two models are different, one would expect the cross-terms to be much smaller than the first two terms. Unlike in Section 5.2, the PDF of  $D(\lambda_1, \lambda_2)$  does not have a convenient parametric form. Therefore, we directly apply bootstrapping to get several observations of the distance metric and use kernel density estimation to get the PDF of the defined distance metric.

Figure 12(d) shows the PDF of the log likelihood differences for GMMs for different sizes of context. Note that the support of the PDF decreases as  $W_L$  increases from 100 to 600. The reliability of the two context models for the same background process increases as the amount of training data increases and hence the variance of normalized log likelihood difference decreases. Therefore, again it is possible to quantify the “unusualness” of outliers caused by corruption of observations from another process ( $\mathbf{P}_2$ ). Similar analysis shows that the same observations hold for HMMs as context models as well. Figure 12(e) shows the PDF of the log likelihood differences for HMMs for different sizes of the context.

#### 5.4. Using confidence measures to rank outliers

In the previous two sections, we looked at the estimation of the PDF of a specific distance metric for context models (memoryless models and HMMs) used in the proposed framework. Then, for a given time series of observations from the two processes ( $\mathbf{P}_1$  and  $\mathbf{P}_2$ ), we compute the affinity matrix for a chosen size of  $W_L$  for the context model. We use the second generalized eigenvector to detect inliers and outliers. Then, the confidence metric for an outlier context  $M_j$  is computed as

$$p(M_j \in O) = \frac{1}{\#I} \left( \sum_{i \in I} P_{d,i}(d \leq d(M_i, M_j)) \right), \quad (18)$$

where  $P_{d,i}$  is the density estimate for the distance metric using the observations in the inlier context  $i$ .  $O$  and  $I$  represent the set of outliers and inliers, respectively, and  $\#$  refers to cardinality operator.

## 6. EXPERIMENTAL RESULTS

In this section, we present the results of the proposed framework with two different content genres mainly using low-level audio features and semantic audio classification labels at the “8 ms frame level” and “one-second level.” The proposed framework has been tested with a total of 12 hours of soccer, baseball, and golf content from Japanese, American, and Spanish broadcasts. For surveillance, we chose 1.5

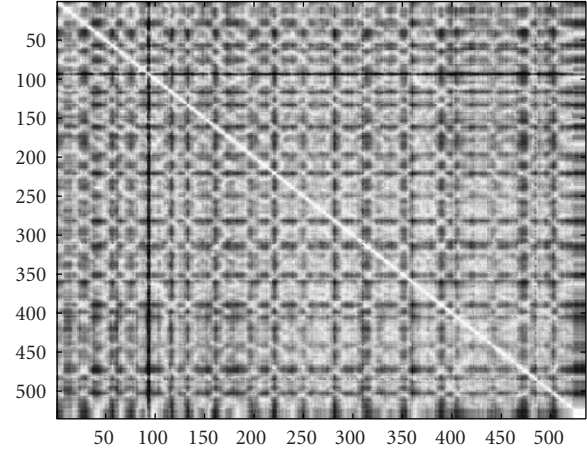


FIGURE 13: Affinity matrix for a 3-hour-long British open golf game using one-second classification labels.

hours of elevator surveillance data and 2.5 hours of traffic intersection video. To our knowledge, this is the first time that outlier-detection-based methods have been applied for audio event discovery in sports and surveillance.

### 6.1. Results with sports audio content

As mentioned earlier, there are three possible choices for time series analysis from which events can be discovered using the proposed outlier subsequence detection framework. They are

- (i) low-level MFCC features;
- (ii) frame-level audio classification labels;
- (iii) one-second-level audio classification labels.

In the following subsections, we show the pros and cons of using each of these time series for event discovery with some example clips from sports audio. Since the one-second-level classification label time series is a coarse representation, we can detect commercials as outliers and extract the program segments from the whole video using the proposed framework. For discovering highlight events (for which the time span is only in the order of few seconds), we use a finer scale time series representation such as the low-level features and frame-level labels.

#### 6.1.1. Outlier subsequence detection using one-second-backgroundlevel labels to extract program segments

Based on the observation that commercials are outliers in the background of the whole program at a coarser time scale, we use the one-second-level audio classification labels as input time series for the proposed framework. Figure 13 shows the affinity matrix for a 3-hour-long golf game. We used 2-state HMMs as context models with  $W_L$  as 120 ( $W_L$ ) classification labels with a step size of 10 ( $W_S$ ). The affinity matrix was constructed using the computed pairwise likelihood distance metric defined earlier. Note that the affinity matrix shows



dark regions against a single background. The dark regions, with low affinity values with the rest of the regions (outliers), were verified to be times of occurrences of commercial sections. Since we use the time series of the labels at one-second resolution, the detected outliers give a coarse segmentation of the whole video into two clusters: the segments that represent the program and the segments that represent the commercials. Also, such a coarse segmentation is possible only because we used a time series of classification labels instead of low-level features. Furthermore, the use of low-level audio features at this stage may bring out some fine-scale changes that are not relevant for distinguishing program segments from nonprogram segments. For instance, low-level features may distinguish two different speakers in the content while a more general speech label would group them as one.

### 6.1.2. *Outlier subsequence detection from the extracted program segments*

Highlight events together with audience reaction in sports video last for only a few seconds. This implies that we cannot look for “interesting” events using the one-second-level classification labels to extract highlight events. If we use one-second-level classification labels, the size of  $W_L$  has to be small enough to detect events at that resolution. However, our analysis on the confidence measures earlier indicates that a small value of  $W_L$  would lead to a less reliable context model thereby producing a lot of false alarms. Therefore, we are left with the following two options:

- (1) to detect outlier subsequences from the time series of frame-level classification labels instead of second-level labels;
- (2) to detect outlier subsequences from the time series of low-level MFCC features.

Clearly, using the frame-level classification labels is computationally more efficient. Also, as pointed out earlier, working with labels can suppress irrelevant changes (e.g., speaker changes) in the background process. Figure 14(a) shows the cluster indicator vector for a section of golf program segment. The size of  $W_L$  used was equal to 8 seconds of frame level classification labels with a step size of 4 seconds. The context model used for classification labels was a 2-state HMM. In the case of low-level features, the size of  $W_L$  was equal to 8 seconds of low-level features with a step size of 4 seconds (see Figure 14(b)). The context model was a 2-component GMM. Note that there are outliers at times of occurrences of applause segments in both cases. In the case of outlier detection from low-level features, there were at least two clusters of speech as indicated by the plot of eigenvector and affinity matrix. Speech 3 (marked in the figure) is an interview section where a particular player is being interviewed. Speech 1 is the commentator’s speech itself during the game. Since we used low-level features, these time segments appear as different clusters. However, the cluster indicator vector from frame-level labels time series affinity matrix shows a single speech background from the 49th

minute to the 54th minute. However, the outliers from the 47th minute to the 49th minute in the frame-level time series were caused by misclassification of speech in “windy” background as applause. Note that the low-level feature time series does not have this false alarm. In summary, low-level feature analysis is good only when there is a stationary background process in terms of low-level features. In this example, stationarity is lost due to speaker changes. Using a frame-level label time series, on the other hand, is susceptible to noisy classification and can bring out false outliers.

Figure 14(c) and 14(d) show the outliers in the frame labels time series and the low-level features time series respectively, for 10 minutes of a soccer game with the same set of parameters as for the golf game. Note that both of them show the goal scoring moment as an outlier. However, the background model of the low-level features time series has a smaller variance than the background model of the frame labels time series. This is mainly due to the classification errors at the frame levels for soccer audio.

In the next subsection, we present our result on inlier/outlier-based representation for a variety of sports audio content.

## 6.2. *Inlier/outlier-based representation and ranking of the detected outliers*

In this section, we show the results of the outlier detection and ranking of the detected outliers. For all the experiments in this section, we have detected outliers from the low-level features time series to perform an inlier/outlier-based segmentation of every clip. The parameters of the proposed framework were set to the following: context window size ( $W_L$ ) = 8 seconds, step size ( $W_S$ ) = 4 seconds, frame rate at which MFCC features are extracted = 125 frames per second, maximum percentage of outliers = 20%, compactness constraint on the background = 0.5, relative time span constraint on the background = 0.35, and the context model is a 2-component GMM. They were not changed for each genre or clip of video. The first three parameters ( $W_L$ ,  $W_S$ , *frame rate*) pertain to the affinity matrix computation from the time series for a chosen context model. The fourth parameter (maximum percentage of outliers) is an input to the system for the inlier/outlier-based representation. The system then returns a segmentation with at most the specified maximum percentage of outliers. The fifth and sixth parameters (compactness and relative size) help in defining what a background is.

First, we show an example inlier/outlier-based segmentation for a 20-minute Japanese baseball clip. In this clip, for the first six minutes of the game, the audience were relatively noisy compared to the later part of the game. There is also a two-minute commercial break between the two parts of the game. Figure 15 shows the temporal segmentation of this clip during every step of the analysis using the proposed framework. The top part of Figure 15(a) shows the result of first normalized cut on the affinity matrix. The bottom part of the same figure (Figure 15(a)) shows the corresponding time segmentation. Since the compactness constraint is not

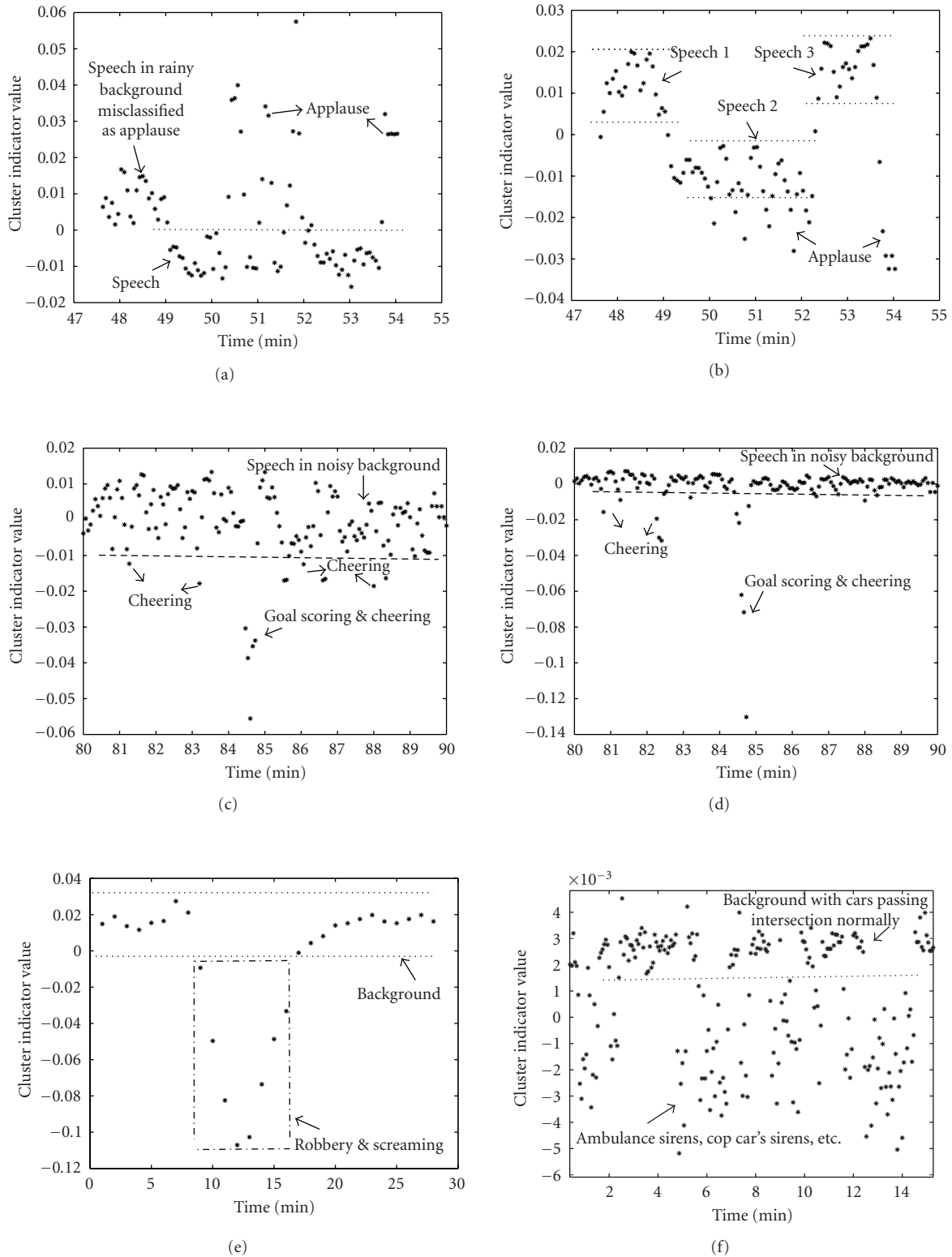


FIGURE 14: Comparison of outlier subsequence detection with low-level audio features and frame-level classification labels for sport and surveillance: (a) outlier subsequences in frame labels time series for golf; (b) outlier subsequences in low-level features time series for golf; (c) outlier subsequences in frame labels time series for soccer; (d) outlier subsequences in low-level features time series for soccer; (e) outlier subsequences in low-level features time series for elevator surveillance; (f) outlier subsequences in low-level Features time series for traffic intersection surveillance.

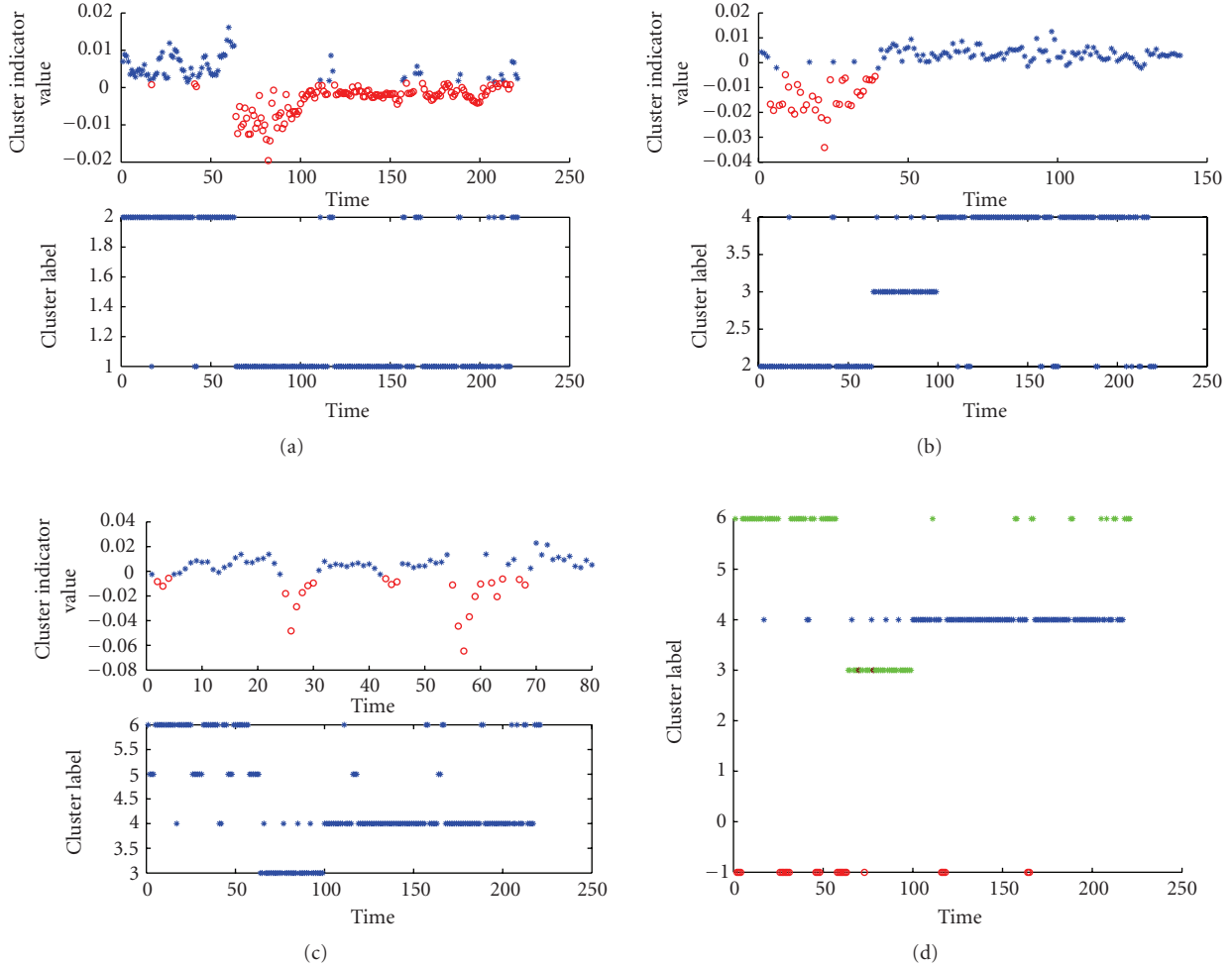


FIGURE 15: Inlier/outlier-based segmentation for a 20-minute clip of a Japanese baseball content. (a) Top: first normalized cut (root partition, 1 minute,  $N_{cut} = 8.980124e - 001$ ); bottom: corresponding temporal segmentation (left radius = 0.876797, right radius=0.805713). (b) Top: second normalized cut (p1 partition, 2 minutes,  $N_{cut} = 9.091716e - 001$ ); bottom: corresponding temporal segmentation (left radius=0.542590, right radius = 0.535301). (c) Top: third normalized cut (p1-1 partition, 3 minutes,  $N_{cut} = 9.473603e - 001$ ); bottom: corresponding temporal segmentation (left radius = 0.411986, right radius = 0.382114). (d) Final temporal segmentation after foreground cut from each background.

satisfied by these two partitions, the normalized cut is recursively applied on these two partitions. When the normalized cut is applied for the second time, the commercial segment is detected as an outlier as shown in Figure 15(b). Figure 15(c) shows the result of normalized cut on the other partition. The final segmentation is shown in Figure 15(d). The outliers were manually verified to be reasonable. As mentioned earlier, outliers are statistically unusual subsequences and not all of them are interesting. Commercial segments and lull periods of the game during which the commentator is silent but the audience are cheering are some example cases which are statistically unusual and not “interesting.” Therefore, after this stage one needs to use a supervised detector such as the excited speech detector to pick out only the “interesting” parts for the summary.

We repeated this kind of inlier/outlier-based segmentation on a total of 4 hours of baseball audio from 5 different

TABLE 1: Outlier ranks in baseball audio;  $R_1$ : average normalized rank using PDF estimate;  $R_2$ : average normalized distance.

Type of outlier	$R_1$	$R_2$
Speech with cheering	0.3648	0.1113
Cheering	0.7641	0.3852
Excited speech-with-cheering	0.5190	0.1966
Speech with music	0.6794	0.3562
Whistle, drums with cheering	0.6351	0.2064
Announcement	0.5972	0.3115

games (2 baseball games from Japanese broadcasts and 3 from American broadcasts). We listened to every outlier clip and classified it by hand as one of the types shown in Table 1. Apart from the three types of outliers mentioned before, we had outliers when there is an announcement in the stadium

TABLE 2: Outlier ranks in soccer audio;  $R_1$ : average normalized rank using PDF estimate;  $R_2$ : average normalized distance.

Type of outlier	$R_1$	$R_2$
Speech with cheering	0.3148	0.1606
Cheering	0.7417	0.4671
Excited speech-with-cheering	0.4631	0.2712
Speech with music	0.5098	0.2225
Whistle, drums with cheering	0.4105	0.2430
Announcement	0.5518	0.3626

TABLE 3: Outlier ranks in golf audio;  $R_1$ : average normalized rank using PDF estimate;  $R_2$ : average normalized distance.

Type of outlier	$R_1$	$R_2$
Silence	0.7573	0.5529
Applause	0.7098	0.4513
Interview	0.1894	0.1183
Speech	0.3379	0.3045

and when there was a small percentage of speech in the whole context. In Table 1, we also show the average normalized ranking and average normalized distance from the inliers for each type of the outlier over all the clips analyzed. It is intuitively satisfying that the speech-with-cheering class is closest to the inliers and has the smallest average rank of all the types. Of all the types, the excited speech-with-cheering and the cheering classes are the most indicative of highlight events.

With the same setting of parameters, we segmented a total of 6 hours of soccer audio from 7 different soccer games (3 from Japanese broadcasts, 3 from American broadcasts, 1 from Spanish broadcasts). The types of outliers in the soccer games were similar to those obtained from baseball games. The results of ranking are also presented for these types of outliers in Table 2. Again, speech-with-cheering outlier is ranked the lowest.

We also segmented 90 minutes of a golf game using the proposed approach. Since the audio characteristics of a golf game is different from that of baseball and soccer, the types of outliers were also different. Applause segments were outliers as expected. The other new types of outliers in golf were: when the commentator was silent and when there is new speaker being interviewed by the commentator. The ranks of the detected outlier types are shown in Table 3.

In the following subsection, we apply the same framework on surveillance audio data to detect unusual events.

### 6.3. Results with surveillance audio content

In the case of sports audio analysis, we used some a priori knowledge about the domain to train sound classes such as applause and cheering to extract two more time series apart from the time series of low-level features. In surveillance, often we do not know beforehand what kinds of sounds can characterize the given data and help us detect unusual events.

TABLE 4: Recognition matrix (confusion matrix) on a 70% training/30% testing split of a data set composed of 4 audio classes: (1) neutral speech; (2) foot steps; (3) banging; (4) nonneutral or excited speech; average recognition rate = 95%.

	(1)	(2)	(3)	(4)
(1)	1.00	0.00	0.00	0.00
(2)	0.00	0.93	0.00	0.07
(3)	0.00	0.00	0.97	0.03
(4)	0.00	0.00	0.10	0.90

We show that the proposed framework provides a systematic methodology to acquire domain knowledge to identify “distinguishable” sound classes. We use low-level features in such scenarios to effectively characterize the domain and detect events without any a priori knowledge. We will discuss more about this in Section 7.

#### 6.3.1. Results with elevator surveillance audio

In this section, we apply the outlier subsequence detection procedure on a collection of elevator surveillance audio data. The data set contains recordings of suspicious activities in elevators as well as some event-free clips. A 2-component GMM was used to model the PDF of the low-level audio features in the 8-second context. Figure 14(d) shows the second generalized eigenvector and the affinity matrix for one such clip with a suspicious activity.

In all the clips with suspicious activity, the outliers turned out to be clips of banging sound against elevator walls and excited speech. Since the key audio classes correlated with suspicious activity turned out to be banging and excited speech, one might argue for the use of audio energy as a feature instead of cepstral features. However, audio energy is an inadequate feature to represent sound classes and cannot characterize the domain. For instance, one would not be able to discriminate between a scream and a loud unsuspecting event. On the other hand, cepstral features enabled identification of typical audio classes to train supervised models (GMMs) for each of the following classes: normal speech, foot steps, bang, excited, or nonneutral speech.

Table 4 presents the classification results for these audio classes. The audio classes of neutral speech and foot steps characterize the background process ( $C_1$ ), whereas short bursts of excited speech and banging sounds correlate with the unusual event in this scenario. After extracting the audio labels, the outlier subsequence detection procedure can be repeated with the discrete audio labels as well to detect events.

#### 6.3.2. Results with traffic intersection surveillance audio

The 2-hour 40-minute long traffic intersection surveillance audio was analyzed using the same framework. The whole audio data consists of clips where cars cross an intersection without an event. It also has an accident event and a number

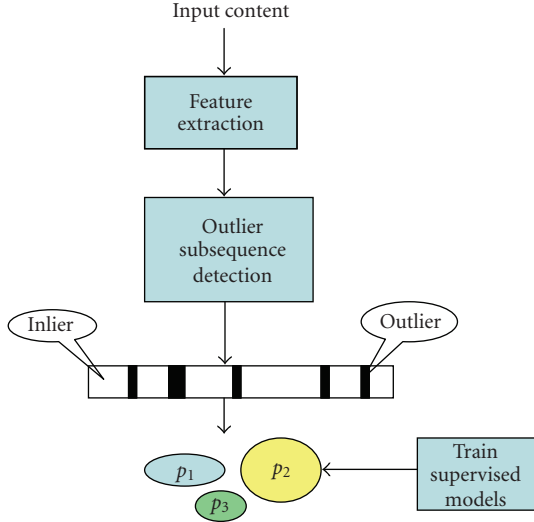


FIGURE 16: Systematic acquisition of domain knowledge using the inlier/outlier-based representation framework.

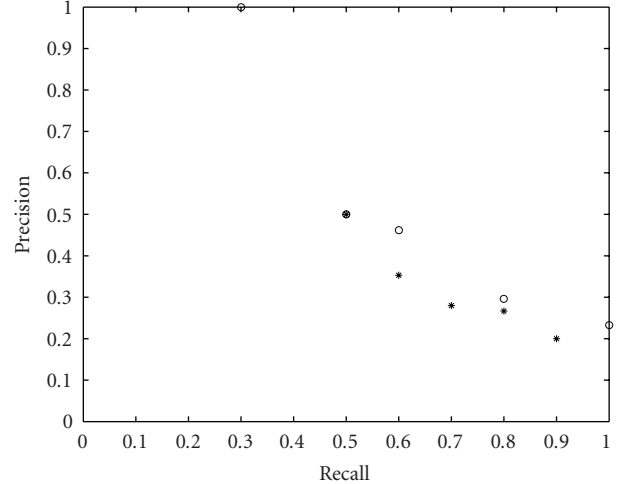
of ambulances and police cars crossing the intersection. The proposed framework was used with the following parameters to detect outliers:  $W_L =$  low-level features for 8 seconds with  $W_S = 4$  seconds. The context model used was a 2-component GMM. Figure 14(e) shows the second generalized eigenvector for the first 15 minutes of this content. It was observed that there were outliers whenever an ambulance crossed the intersection. The accident that occurred with a crashing sound was also an outlier.

## 7. SYSTEMATIC CHOICE OF KEY AUDIO CLASSES

From all the experiments with sports and surveillance audio content, one can infer that the proposed framework not only gave an inlier/outlier-based temporal segmentation of the content but also distinguishable sound classes for the chosen low-level features in terms of distinct backgrounds and outlier sound classes. Then, by examining individual clusters and outliers, one can identify consistent patterns in the data that correspond to the events of interest and build supervised statistical learning models.

Thus, the proposed analysis and representation framework can be used for systematic choice of key audio classes as shown in Figure 16.

We cite an example in which this framework was useful for acquiring domain knowledge. In the previous section, we showed that one can also use audio classification labels as a time series and discover events. However, the choice of audio classes to be trained for the audio classification framework involves knowledge of the domain in terms of coming up with representative sound classes that cover most of the sounds in the domain. For example, we chose the following five classes for the audio classification framework in sports domain, namely, applause, cheering, music, speech,



\* Cheering & applause (%)  
 ○ "Highlight" class labels (%)

FIGURE 17: Comparison of precision-recall performance using cheering and applause against using the discovered "highlight" audio class.

and speech with music. The intuition was that the first two classes capture the audience-reaction sounds and the rest of the classes represent bulk of sounds in the "uninteresting" parts of the sports content. However, by using the proposed outlier subsequence detection framework on the low-level features, we discover that the "key" highlight audio class is a mixture of audience cheering and the commentator's excited speech and not cheering of the audience alone. We used time series analysis on the low-level features to come-up with an inlier/outlier-based representation of the content without any a priori knowledge. After examining the detected outliers, we discovered that the "key" highlight class is a mixture of audience cheering and commentator's excited speech. We collected several training examples from the detected outliers for the key highlight audio class and trained a GMM. The learnt model has been tested for highlights extraction from 27 sports videos including soccer, baseball, sumo wrestling, and horse race. In terms of precision, the highlights extraction system based on this discovered highlight class outperforms the state-of-the-art highlights extraction system that uses the percentage of cheering audio class as a measure of interestingness as shown in Figure 17 [27]. In other words, the systematic choice of audio classes led to a distinct improvement in the highlights extraction even though sports is a very "familiar" or "well-known" content genre. Note that with less understood domains such as surveillance, choice of audio classes based on pure intuition could lead to even worse accuracy of event detection. Furthermore, for surveillance domains especially, the audio classes cannot all be anticipated since there is no restriction on the kinds of sounds.

As pointed out earlier, we used this framework for selecting the sound classes to characterize the elevator surveillance audio data and achieved accurate detection of notable events. In this case, the isolation of the elevator car results

in a relatively noise-free environment, which makes the data set much more amenable to analysis than is broadcast sports content.

Before we conclude, let us look at the computational complexity of each of the blocks in the analysis framework. The computational complexity of the first stage (MFCC feature extraction) is  $\approx N \times W_L \times O(256 \log 256) + O(32 \log 32)$ . Here  $N$  is the number of context windows;  $W_L$  is the number of MFCC frames per context window.  $O(256 \log 256)$  is for the FFT computation for a frame size of 256 at 16 kHz.  $O(32 \log 32)$  is for the DCT computation from the filter bank outputs. The computational complexity of the second stage (EM for GMMs) is  $\approx N \times O(i \times W_L \times D^2)$  [28]. Here  $D$  is the dimensionality of the MFCC features;  $i$  is the number of training iterations. The computational complexity of the third stage (affinity matrix computation) is  $\approx O(N^2)$ . The computational complexity of the last stage (eigenvector computation) is  $\approx O(O(N^{1/2}) \times N) + O(O(N^{1/2}) \times O(N))$  [18]. The computational complexity of the current framework is clearly very high. Our future work will focus on reducing the computational complexity while allowing for graceful degradation in performance.

## 8. CONCLUSION

We proposed a content-adaptive analysis and representation framework for audio event discovery from unscripted multimedia. The proposed framework is based on the observation that “interesting” events happen sparsely in a background of usual events. We used three time series for audio event discovery, namely, low-level audio features, frame-level audio classification labels, one-second-level audio classification. We performed an inlier/outlier-based temporal segmentation of these three time series. The segmentation was based on eigenvector analysis of the affinity matrix obtained from statistical models of the subsequences of the input time series. The detected outliers were also ranked based on deviation from the background process. Experimental results on a total of 12 hours of sports audio from three different genres (soccer, baseball, and golf) from Japanese, American, and Spanish broadcasts show that unusual events can be effectively extracted from such an inlier/outlier-based segmentation resulting from the proposed framework. It was also observed that not all outliers correspond to “highlight” events and one needs to incorporate domain knowledge in the form of supervised detectors at the last stage to extract highlights. Then, using the ranking of the outliers a summary of desired length can be generated. We also discussed the pros and cons of using the aforementioned three kinds of time series for audio event discovery. We also showed that unusual events can be detected from surveillance audio without any a priori knowledge using this framework. Finally, we have shown that such an analysis framework resulting in an inlier/outlier-based temporal segmentation of the content postpones the use of content-specific processing to as late a stage as possible and can be used to systematically select the key audio classes that are indicative of events of interest.

## 9. ACKNOWLEDGMENTS

The authors would like to thank Professor Nasir Memon at Polytechnic University for helpful discussions. We would also like to thank Professor Yair Weiss (Hebrew University) for his suggestions on spectral clustering and Professor Eamonn Keogh (University of California, Riverside) for his useful comments on time series analysis. We would also like to thank Dr. Daniel Nikovski, Dr. Mathew Brand, and Dr. Baback Moghaddam from MERL for helpful suggestions. We would like to thank Dr. Bhiksha Raj and Dr. Paris Smaragdhis for the help on audio classification framework.

## REFERENCES

- [1] R. S. Jasinschi, N. Dimitrova, T. McGee, L. Agnihotri, J. Zimmerman, and D. Li, “Integrated multimedia processing for topic segmentation and classification,” in *Proceeding of International Conference on Image Processing (ICIP '01)*, vol. 3, pp. 366–369, Thessaloniki, Greece, October 2001.
- [2] R. Lienhart, “Automatic text recognition for video indexing,” in *Proceeding of 4th ACM International Conference on Multimedia (ACM Multimedia '96)*, pp. 11–20, Boston, Mass, USA, November 1996.
- [3] A. Hanjalic, G. Kakes, R. L. Lagendijk, and J. Biemond, “DANCERS: Delft advanced news retrieval system,” in *IS&T/SPIE Electronic Imaging 2001: Storage and Retrieval for Media Databases 2001*, vol. 4315 of *Proceedings of SPIE*, pp. 301–310, San Jose, Calif, USA, January 2001.
- [4] Y. Wang, Z. Liu, and J.-C. Huang, “Multimedia content analysis-using both audio and visual clues,” *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.
- [5] H. Winston, H.-M. Hsu, and S.-F. Chang, “A statistical framework for fusing mid-level perceptual features in news story segmentation,” in *Proceeding of IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 413–416, Baltimore, Md, USA, July 2003.
- [6] A. Aner and J. R. Kender, “Video summaries through mosaic-based shot and scene clustering,” in *Proceeding of 7th European Conference on Computer Vision (ECCV '02)*, vol. 4, pp. 388–402, Copenhagen, Denmark, May–June 2002.
- [7] Y. Li and C. C. Kuo, *Content-based video analysis, indexing and representation using multimodal information*, Ph.D. thesis, University of Southern California, Los Angeles, Calif, USA, 2003.
- [8] H. Sundaram and S.-F. Chang, “Determining computable scenes in films and their structures using audio-visual memory models,” in *Proceeding of 8th ACM International Conference on Multimedia (ACM Multimedia '00)*, pp. 95–104, Los Angeles, Calif, USA, October–November 2000.
- [9] N. Nitta, N. Babaguchi, and T. Kitahashi, “Extracting actors, actions and events from sports video—a fundamental approach to story tracking,” in *Proceeding of 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 718–721, Barcelona, Spain, September 2000.
- [10] A. Ekin and A. M. Tekalp, “Automatic soccer video analysis and summarization,” in *IS&T/SPIE 15th Annual Symposium on Electronic Imaging Science and Technology: Storage and Retrieval for Media Databases 2003*, vol. 5021 of *Proceedings of SPIE*, pp. 339–350, Santa Clara, Calif, USA, January 2003.
- [11] H. Pan, P. van Beek, and M. I. Sezan, “Detection of slow-motion replay segments in sports video for highlights

- generation,” in *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 3, pp. 1649–1652, Salt Lake City, Utah, USA, May 2001.
- [12] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian, “Creating audio keywords for event detection in soccer video,” in *Proceeding of IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 281–284, Baltimore, Md, USA, July 2003.
- [13] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, “Unsupervised mining of statistical temporal structures in video,” in *Video Mining*, A. Rosenfeld, D. Doermann, and D. DeMenthon, Eds., Kluwer Academic, Boston, Mass, USA, pp. 279–309, October 2003.
- [14] G. Wu, Y. Wu, L. Jiao, Y.-F. Wang, and E. Y. Chang, “Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance,” in *Proceeding of 11th ACM International Conference on Multimedia (ACM Multimedia '03)*, pp. 528–538, Berkeley, Calif, USA, November 2003.
- [15] Z. Xiong, Y. Rui, R. Radhakrishnan, A. Divakaran, and T. S. Huang, “A unified framework for video summarization, browsing and retrieval,” in *Handbook of Image & Video Processing*, pp. 1013–1030, Academic Press, San Diego, Calif, USA, 2nd edition.
- [16] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, “Effective and efficient sports highlights extraction using the minimum description length criterion in selecting GMM structures [audio classification],” in *Proceeding of IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 3, pp. 1947–1950, Taipei, Taiwan, June 2004.
- [17] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, “Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework,” in *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 5, pp. 632–635, Hong Kong, China, April 2003.
- [18] J. Shi and J. Malik, “Normalized cuts and image segmentation,” in *Proceeding of Computer Vision and Pattern Recognition (CVPR '97)*, pp. 731–737, San Juan, Puerto Rico, USA, June 1997.
- [19] R. P. Rao and W. A. Pearlman, “Multirate vector quantization of image pyramids,” in *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91)*, vol. 4, pp. 2657–2660, Toronto, Ontario, Canada, April 1991.
- [20] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY, USA, 1973.
- [21] P. Perona and W. T. Freeman, “A factorization approach to grouping,” in *Proceeding of 5th European Conference on Computer Vision (ECCV '98)*, vol. 1, pp. 655–670, Freiburg, Germany, June 1998.
- [22] M. P. Wand and M. C. Jones, *Kernel Smoothing*, Chapman & Hall, London, UK, 1995.
- [23] S. J. Sheather and M. C. Jones, “A reliable data-based bandwidth selection method for kernel density estimation,” *Journal of the Royal Statistical Society. Series B*, vol. 53, no. 3, pp. 683–690, 1991.
- [24] D. Comaniciu, V. Ramesh, and P. Meer, “The variable bandwidth mean shift and data-driven scale selection,” in *Proceeding of 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 438–445, Vancouver, British Columbia, Canada, July 2001.
- [25] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, NY, USA.
- [26] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
- [27] R. Radhakrishnan, I. Otsuka, Z. Xiong, and A. Divakaran, “Modeling sports highlights using a time-series clustering framework and model interpretation,” in *Storage and Retrieval Methods and Applications for Multimedia 2005*, vol. 5682 of *Proceedings of SPIE*, pp. 269–276, San Jose, Calif, USA, January 2005.
- [28] B. Upcroft, L. L. Ong, S. Kumar, M. Ridley, T. Bailey, et al., “Rich probabilistic representations for hearing only decentralised data fusion,” in *Proceeding of The Eighth International Conference on Information Fusion*, Philadelphia, Pa, USA, July 2005.

**Regunathan Radhakrishnan** received the B.E. (with honors) degree in electrical engineering and M.S. (with honors) degree in chemistry from Birla Institute of Technology and Science (BITS), Pilani, India, in 1999. He worked as DSP Engineer in Multimedia Codecs Group at SASKEN Communication Technologies Ltd., Bangalore, India, in 1999–2000. He received the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in 2002 and 2004, respectively. He was a Research Fellow in the ECE Department and also an intern at Mitsubishi Electric Research Labs, Cambridge, Mass, during his graduate studies. He joined Mitsubishi Electric Research Laboratories (MERL) in 2005 as a Visiting Researcher. His current research interests include audio classification, multimedia mining, digital watermarking and content security, and data mining. He has published several conference papers, as well as 5 journal papers and 3 book chapters on multimedia content analysis and security.



**Ajay Divakaran** received the B.E. (with honors) degree in electronics and communication engineering from the University of Jodhpur, India, in 1985, and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute, Troy, NY, in 1988 and 1993, respectively. He was an Assistant Professor with the ECE Department, University of Jodhpur, in 1985–1986. He was a Research Associate at the ECE Department, Indian Institute of Science, Bangalore, in 1994–1995. He was a Scientist with Iterated Systems Inc., Atlanta, Ga, from 1995 to 1998. He joined Mitsubishi Electric Research Laboratories (MERL) in 1998 and is now a Senior Team Leader/Senior Principal Member of technical staff. He has been a key contributor to the MPEG-7 video standard. His current research interests include video and audio analysis, summarization, indexing and compression, and related applications. He has published several journal and conference papers, as well as six invited book chapters on video indexing and summarization. He has supervised three doctoral theses. He currently serves on program committees of key conferences in the area of multimedia content analysis. He currently leads the Data and Sensor Systems Team at the Technology Laboratory of MERL.



**Ziyou Xiong** received his B.S. degree from Wuhan University, Hubei Province, China, in July 1997. He received his M.S. degree in electrical and computer engineering from University of Wisconsin, Madison, in December 1999 and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in October 2004. Since September 2004, he has been with the United Technologies Research Center as a Senior Researcher/Scientist in East Hartford, Connecticut. In the summers of 2003 and 2004, he worked on sports video analysis at Mitsubishi Electric Research Laboratories, Cambridge, Mass. He has coauthored a book titled *Facial Analysis from Continuous Video with Application to Human-Computer Interface* (Kluwer, 2004). His research interests include image and video analysis, computational audio-visual scene analysis, and machine learning.



**Isao Otsuka** received his B.E. degree in precision mechanical engineering from Meiji University, Japan, in 1989. He joined Mitsubishi Electric Corporation in 1989 and has been engaged in the research and development of speaker system and sound field analysis for home theater and car audio applications. Recently, he has been engaged in the development of system and man-machine interface for storage devices such as DVD-HDD recorders/players. His interests span applications for digital storage devices as well as relevant core technologies such as audio-visual content analysis. He has recently published several conference papers on video summarization and audio classification. He has several issued and pending patents on video summarization and other storage device-related topics. He has participated in standardization activities in the DVD.

