Content Discovery from Composite Audio

An unsupervised approach

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op woensdag 2 december 2009 om 12:30 uur

door

Lie LU

Master of Science in Electrical Engineering, Shanghai Jiao Tong University, P.R. China
geboren te Zhejing, P.R. China

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. R. L. Lagendijk

Copromotor:
Dr. A. Hanjalic

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. R. L. Lagendijk,	Technische Universiteit Delft, promotor
Dr. A. Hanjalic,	Technische Universiteit Delft, copromotor
Prof. dr. G. J. Houben,	Technische Universiteit Delft
Prof. dr. ir. J. W. M. Bergmans,	Technische Universiteit Eindhoven
Prof. dr. ir. C. H. Slump,	Universiteit Twente
Dr. H.-J. Zhang,	Microsoft Advanced Technology Center, Beijing, China
Dr. A. Divakaran,	Sarnoff Corporation, Princeton, NJ, USA
Prof. dr. ir. J. Biemond,	Technische Universiteit Delft, reservelid

Microsoft Research Asia (MSRA) has provided substantial support in the preparation of this thesis.

Content Discovery from Composite Audio: An unsupervised approach
LU, Lie
Thesis Delft University of Technology – With ref. – With summary in Dutch
Published by TU Delft Mediamatica
ISBN 978-90-813811-4-7

Contents

Chapter 1

Introduction

In the age of information explosion, the amount of published information and available data is rapidly increasing. As the amount of available data grows, the problem of managing the information contained in this data becomes more and more difficult. Search – to quickly find and access the information of interest – is the most fundamental information management functionality. To maximize the effectiveness of this functionality, many information retrieval technologies and search engine systems (including Web search and desktop search engines, such as Google, Live search and Yahoo!) have been developed and achieved great successes in the past years. They have become a part of everyday life for many people.

To make the data searchable, it should be indexed first. As the existing search engines focus on offering reliable and robust solutions for searching in text document collections, index terms are obtained by finding important words (keywords or key terms) that are extracted from the available text document resources, including the title, body text and anchor text (e.g. the text linked to this a document). For example, classical text information retrieval methods [Robertson and Sparc-Jones 1997] can be used to reveal the importance of individual words based on the frequency of their appearance and their uniqueness for a particular document. Moreover, the text layout can provide useful information for indexing as well. For instance, the parts of the text having a large font or being indicated in bold can be considered more descriptive to the corresponding document than other document sections. The indexed terms together with their importance indicators (i.e. weights) can then be used to measure the *content relevance* of a particular target document to the search query. Furthermore, *content importance*, i.e. the relative importance of a document, can be combined with content relevance to rank relevant documents. Content importance can usually be revealed by link analysis, for instance by using the PageRank algorithm [Brin and Page 1998] deployed in Web search.

Besides the text information, more and more multimedia data, including video, audio and images are available in various digital libraries and databases, and on the Internet. These multimedia data include movies, sports, news broadcasts, music, TV and radio programs, and tremendous amounts of photos. Similar to text collections, there is also increasing need to search for the information of interest in large multimedia data collections. For example, how to find a video lecture talking about 'information retrieval'? How to find highlights or particular scenes of a soccer video? How could we find the sounds of applause and cheering from movies and reuse them when we are editing our own audio, video or podcast recordings? How could we identify the violent scenes from movies and prevent children from seeing these movies? Finally, how could we get music, video and photos recommended to us based on our general interest and/or our specific interest in a given use context?

Just like in the case of a text database, a multimedia database can be made searchable through indexing. However, while a text database can be indexed using the basic database items (words) directly, indexing of a multimedia database needs to be done by assigning *metadata* (data about the data) to multimedia items (also referred to as *multimedia documents*). To obtain rich metadata for multimedia indexing, the most straightforward way is manual annotation of the multimedia content. For example, YouTube.com usually asks the users to insert some keywords to describe the video content they upload, and Pandora.com assigns each song up to 400 distinct musical characteristics obtained by trained music analysts to help users discover more music they like. Manual annotation is useful in some applications and can provide accurate description of the content due to the (professional) background of the annotators. However, there are also some critical disadvantages: manual annotations are subjective, and their generation is typically tedious, expensive and time consuming. More automation in metadata generation processes can be introduced in some application scenarios, like those involving Internet multimedia. If a multimedia document is published on a webpage, its surrounding text information could be used to describe its content, so that traditional text information retrieval technology could be employed for multimedia data search in such a case. Good examples are image search and video search mechanisms provided by some search engines (e.g. Google image search). However, as an image is worth more than a thousand words (and a video thus even more), the available surrounding text is usually insufficient to enable reliable multimedia search and retrieval in a general case.

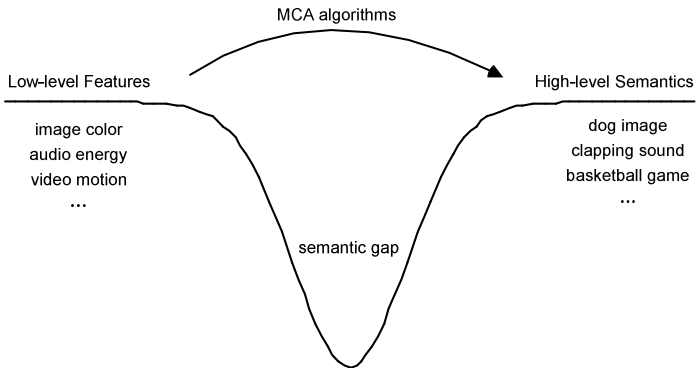


Fig. 1.1 An illustration of the semantic gap between features and semantics. Interpretation of the data requires bridging of the gap, that is, a mapping between the features and semantic descriptors.

1.1 Multimedia Indexing through Content Analysis

A promising way of automatically generating rich sets of multimedia indexes is by applying the theory and algorithms of *multimedia content analysis* (MCA). By bringing together the fields of multimedia signal processing, pattern recognition, perception, psychology and linguistics, and by combining them into sophisticated knowledge inference mechanisms, the MCA attempts to reveal the meaning of data at various levels of abstraction and so to directly provide meaningful entries for database browsing and query. MCA has become one of the most challenging and most rapidly developing research directions in computer science. Many classes of MCA algorithms have emerged over the years to address the problems like multimedia content parsing, grouping (clustering), classification, summarization and highlighting [Hanjalic et al. 2008].

The biggest hurdle faced by an MCA algorithm is the *semantic gap* – the gap between the *features* extracted from data and the *semantics* of that data, as illustrated in Fig. 1.1. We refer to the process of bridging the semantic gap (the process of interpreting the data) as *semantic inference*, and to the results of the inference process as *semantic descriptors*. The features are used to represent the signal–level properties of the analyzed data, such as a color in an image, the energy of a sound, and the properties of the object or camera motion in a video. The *semantic descriptors* are meant to capture the meaning of the data as perceived by a human, and, as such, to

index the multimedia data for the purpose of search and retrieval. We therefore refer to these descriptors also as *semantic indexes*.

Since the meaning of an image or a video clip can be defined at various abstraction levels, the same holds for inferring the semantic descriptors from data. We therefore distinguish among three main abstraction levels, at which semantic inference can be performed:

- the *affective* level,
- the level of *semantic concepts*, and
- the level of *semantic structure*.

At the affective level, an image, a video clip or a music piece is interpreted in terms of the affective response (i.e. a feeling or mood) they are expected to elicit from a human. Examples of such descriptors are those pointing to a "romantic" or an "exciting" scene [Hanjalic and Xu 2005]. Semantic concepts, also referred to as *semantic classes*, stand for meaningful visual objects (e.g. a "dog" shown in a picture) or temporal events (e.g. an "action scene" in a movie, or an "instrumental solo" in a music piece). Because some semantic concepts (e.g. visual objects) are often components of other semantic concepts (e.g. larger visual objects or temporal events), the inference of some concepts provides input into the inference of others. In this sense, various levels of semantic concepts and the corresponding semantic descriptors are often distinguished as well. Finally, the descriptors at the semantic structure level point to the meaningful breaks in the content flow (e.g. boundaries between two topics in a news video, the start of a commercial break), or guide the process of grouping together those multimedia documents that belong together in terms of their content (e.g. grouping all "dog" pictures together).

While MCA algorithms aim at finding reliable mapping between the measured features and perceived semantics, obtaining such mapping is difficult in many practical cases. This problem can best be illustrated by the cases where two images of different semantic concepts have similar features, or where two images with the same semantic concepts have completely different features. Fig. 1.2(a) shows two images with very similar visual features (such as color and texture) but representing totally different semantic concepts (a "woman" and a "dog"), while the images in Fig. 1.2(b) both show dogs, but have very different visual features.

The first intuitive step in bridging the semantic gap is to enrich the feature space that provides input into the MCA algorithms. To do this, optimal use of the available information channels of multimedia documents is required. Such information channels can be found in different *modalities*, such as

(a) (b)

Fig. 1.2 Illustration of the semantic gap: (a) different semantics but similar features (b) same semantics but different features

- **visual modality**, which includes three main categories: image, graphics and image sequence,
- **text modality**, which includes the text describing a multimedia object, including the surrounding and overlaid text and closed captions.
- **audio modality**, which also considers three main categories: speech, music and noise. We distinguish here between *structured* and *unstructured* noise. While unstructured (e.g. white) noise is typically disturbing and not interesting for search and retrieval tasks, the structured noise category includes various potentially interesting sounds that we will refer to as *background* or *environmental sounds* or *audio effects*. Examples of these are the sounds of stepping, laughter, applause, explosion, car engine and cheering.

Many multimedia documents contain multiple modalities. For example, a Web site on a given topic usually contains illustrations, figures, photos, text describing the topic, and often also the related video clips. Furthermore, a video is typically referred to as a composite audio-visual data stream consisting of an image sequence, but also often containing an audio track, overlaid text and closed captions.

To optimally combine the information from different modalities, two basic approaches could be deployed, as illustrated in Fig.1.3: feature-level fusion (*early fusion*) and decision-level fusion (*late fusion*) [Hall and Llinas 1997]. In the feature-level fusion scheme, a feature vector is extracted from each modality first. These feature vectors are then aligned and concatenated together into a single larger feature vector, which serves as input into a semantic inference mechanism based on, for

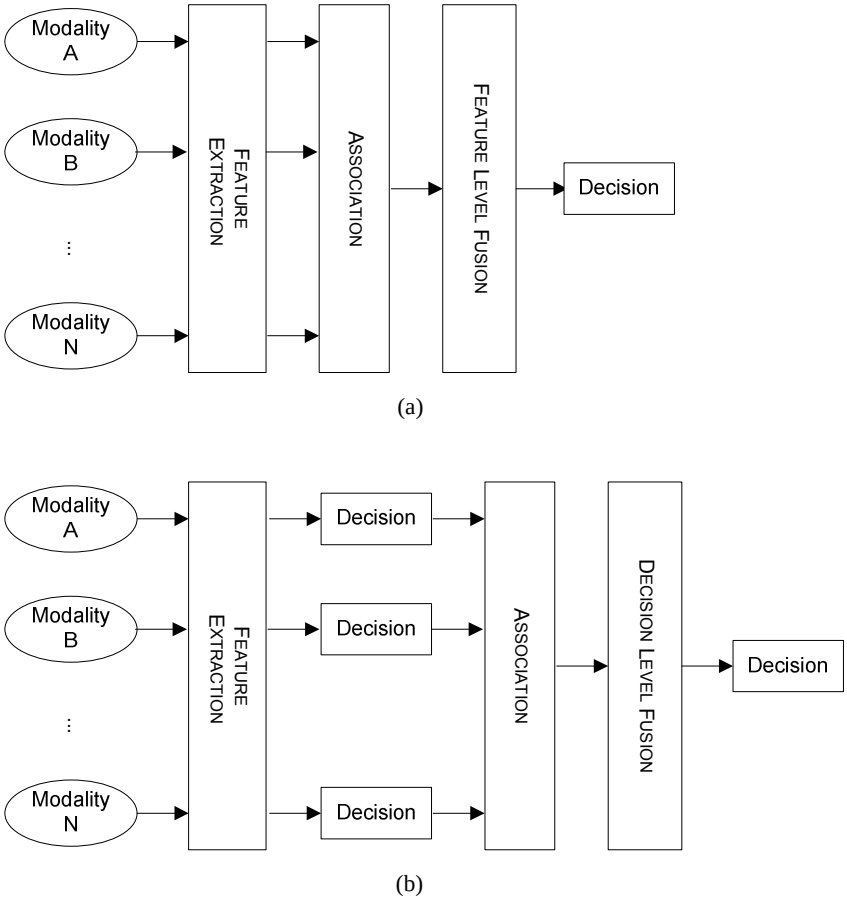


Fig.1.3 An illustration of two basic multimodality fusion mechanisms: (a) feature-level fusion, (b) decision-level fusion

instance, a support vector machine (SVM) or a clustering algorithm [Naphade et al. 2001][Hua et al. 2005]. Compared to this, in a decision-level fusion scheme, the semantic inference is performed on each individual modality first based on its own feature vector. Then, all intermediate inference results, also referred to as *mid-level semantic descriptors*, are combined together to obtain the final decision regarding the semantic content of the analyzed data by using, for instance, heuristic rules or probabilistic inference mechanisms [Rui et al. 2000][Duan et al. 2003]. As an example, in the MCA approach developed for a sports video [Duan et al. 2003], some mid-level semantic descriptors, such as *player close-up*, *field view*, and *audience view*, are

obtained first from the visual modality. Furthermore, the information regarding the occurrence of *applause*, *commentator speech*, and *whistling* is obtained through an analysis of the audio modality. By integrating this multi-modal mid-level representation of the sports video content, high-level semantic descriptors can be inferred, such as *In Play*, *Out-of-Play*, *Foul*, *Free Kick*, and *Penalty Kick*.

1.2 Thesis Focus: Content-based Audio Analysis

Intuitively, the semantic inference from multimedia documents will benefit from an analysis of the issues playing a role in the inference process at each individual modality. Examples of such issues are the features to be selected per modality and the possibilities to bring these features in relation to semantic descriptors at various abstraction levels. As an integrated part of multimedia documents, audio usually plays an important role in MCA theory and algorithms, in particular if it is combined with an image sequence into an audio-visual data stream (video). However, compared to a relatively intensive research effort invested in semantic inference from text, images and image sequences, semantic inference from audio signals has received less attention in the MCA research community. This thesis focuses on the subset of MCA theory and algorithms addressing the audio modality only, and aims at exploring the possibilities and providing insights for developing robust solutions for the semantic inference from audio signals that we will also refer to as *content-based audio analysis*.

A general scheme of content-based audio analysis can be represented by a black-box inference system shown in Fig. 1.4. There, the system outputs semantic index(es) for a given input audio signal based on pre-specified prior knowledge like, for instance, the type of semantic concepts or semantic structure expected to be found in the analyzed data, MCA model assumptions and training data used. Depending on the level at which prior knowledge is specified, this inference system can be realized by employing various approaches ranging from purely supervised to fully unsupervised ones. For example, if the scheme in Fig. 1.4 is seen as a speech recognition system, the prior knowledge, such as the labeled audio data, dictionary and grammar, need to be pre-collected to train both an acoustic model and a language model in a supervised fashion [Huang et al. 2001]. Similarly, trained models of the semantic concepts, such as *car-racing*, *siren*, *gun-shot*, and *explosion*, can be used to detect the occurrences of these concepts in movie soundtracks [Cheng et al. 2003]. Compared to these supervised realizations, an unsupervised approach can be employed to find “unusual” events in the sound track of a surveillance audio signal [Radhakrishnan et al. 2004], for which typically little prior knowledge can be collected.

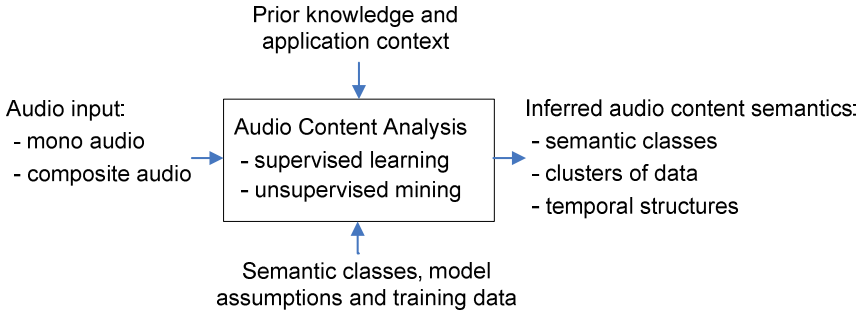


Fig. 1.4 A general scheme of content-based audio analysis, specifying various possible types of input audio signals, prior knowledge and application context, the types of inference techniques and inference results

In addition to different levels of prior knowledge, different types of inference techniques (e.g. supervised vs. unsupervised), and different corresponding types of inferred results (e.g. semantic concepts vs. semantic structure elements), practical cases also differ from each other regarding the type of audio signals serving as input. For example, in some cases, audio signals consisting of one category only (i.e. pure speech) are processed, while other deal with more complex audio signals resulting from a combination of several audio categories. In the remainder of this thesis we will refer to such compound audio signals as *composite audio*.

1.3 Thesis Scope: Unsupervised Analysis of Composite Audio

In this section we analyze in more detail the realization possibilities and applications of the scheme in Fig. 1.4, and explain the specific choices we make in this thesis, from which we expect to lead us towards a robust framework for content-based analysis of composite audio signals.

1.3.1 Composite Audio

In many applications and scenarios dealing with the audio-visual content of sports, broadcasts, movies, news, and radio programs, audio signals appearing therein contain not only speech and music, but also various audio effects, such as cheering and applauses. For instance, in a radio program, speech may be frequently interrupted by

music or sound effects, while in an action movie a much more complex sound track containing speech, music, and various sounds of explosion, gun-shots, car-chasing, and screaming can be found. These sounds are typically not only temporally interleaved (*temporally composite*), but often also spectrally mixed (*spectrally composite*) when occurring simultaneously. Therefore, to be able to support multimedia information retrieval in a general case, and to make the system in Fig. 1.4 less sensitive to unpredicted mixtures of different audio categories, we assume in this thesis that the input in Fig. 1.4 is a composite audio signal. Compared to this, pure audio categories, such as speech or music, can be considered as input when developing dedicated solutions for specific applications. The development of these solutions falls in the domain such as *speech recognition* or *music information retrieval* [Casey et al. 2008], and is beyond the scope of this thesis.

1.3.2 Audio Scene Detection and Grouping

In view of many aspects of audio content semantics, it is necessary to define which of these aspects we concentrate on in this thesis. This definition will help formulate a clear objective, based on which we can approach a specific realization of the general scheme in Fig. 1.4.

Referring to the definition of the three main abstraction levels of semantic inference in Section 1.1, we address in this thesis the problem of inferring the semantic structure of a composite audio data stream. We first search for mechanisms to discover meaningful, semantically coherent structure elements of an input composite audio signal that we will further refer to as *semantic segments* or *audio scenes*. An audio scene can be seen as an equivalent of a text paragraph, or a *logical story unit* [Hanjalic et al. 1999] targeted by the algorithms for video content segmentation. Examples of audio scenes we aim at in this thesis are the segments in the video soundtracks corresponding to a movie scene, a news report or a particular event, like the applauding audience or the segment between the serve and the end of a game in a tennis match. We emphasize here that our goal is not to infer the meaning of a segment but solely its boundaries. In this sense, we also aim at developing a segmentation framework that is generic enough to handle various content genres (e.g. sports, movies, TV shows).

While the classical approach to audio segmentation infers audio scenes based on a direct analysis of features, we consider in this thesis an alternative approach that builds on the analogy to the text document analysis. This approach requires an intermediate analysis step resulting in a first set of semantic descriptors, which are then used to facilitate the audio scene discovery step. As an analogy to the discussion in Section 1.1,

we again refer to these intermediate results as mid-level semantic descriptors, as opposed to the *high-level descriptors* that point to audio scene boundaries. We show in this thesis that our two-step segmentation approach can lead to a significant increase in segmentation robustness compared to the traditional approach.

Once audio scenes are detected, we investigate the possibilities to automatically group them together into meaningful clusters to facilitate further steps in audio and multimedia content management. In the development of our clustering approach, we again rely on the same mid-level semantic descriptors that were applied in the segmentation step. This opens the possibility to deploy alternative clustering concepts, such as *co-clustering*, which is also likely to result in a considerable increase in performance compared to the classical clustering methods. Just like in the segmentation case, we like to emphasize that we are interested in detecting which audio scenes belong together in terms of their content, rather than in recognizing that content, which again implies that generic solutions are searched for.

1.3.3 Unsupervised Semantic Inference

To infer the semantic content of audio scenes from a composite audio signal, two general classes of approaches can be deployed: *supervised* or *unsupervised* approach.

Existing works on content-based audio analysis have usually adopted supervised data analysis and classification methods. For instance, Gaussian mixture model (GMM), hidden Markov model (HMM), support vector machine (SVM), and Bayesian Network are often used to model and identify various aspects of audio content semantics. Examples can be found in [Cai et al. 2003a][Xu et al. 2003][Moncrieff et al. 2001][Cheng et al. 2003]. Although the supervised approach has proved to be effective in many applications, it shows some critical limitations. First, the effectiveness of the supervised approach relies heavily on the quality of the training data. If the training data is insufficient or badly distributed, the system performance drops significantly. Second, in most real-life applications, like pervasive computing [Ellis and Lee 2004] and surveillance [Radhakrishnan et al. 2004], it is difficult to list all the semantic categories that could possibly be found in data. Thus it is impossible to collect training data and learn proper statistical models in these cases.

In view of the described disadvantages of the supervised methods, some unsupervised techniques like clustering have emerged as an alternative to supervised content classification. The unsupervised approach has the advantage that it requires neither the predefined semantic categories nor the offline collected training data. However, the resulting wider application scope comes together with the disadvantage

that the unsupervised approach can only lead to meaningful data clusters but cannot automatically reveal the exact meaning of each cluster (e.g. class labels as links to the corresponding semantic concepts).

In view of our goals defined earlier in this section, we choose for an unsupervised semantic inference approach to develop our realization of the scheme in Fig. 1.4. In addition to the fact that the unsupervised approach has been considered much less in recent literature than the supervised one, this choice is motivated mainly by our goal to develop theoretical foundations and a practical implementation of a robust content-based audio analysis method, where robustness is mainly searched in the capability of the analysis framework to effectively deal with a wide range of variations in signal combinations characterizing a composite audio data.

As opposed to a supervised approach where the match is evaluated between a trained model of a semantic concept and the signal behavior in a given audio segment, the unsupervised approach requires mining or discovery of potentially meaningful patterns and structure elements in audio signals. To emphasize this, we will often refer to our approach also as *content discovery from composite audio*.

1.4 Thesis Contribution and Outline

We conclude this chapter by providing a brief summary of the thesis goal, objectives and contributions, and an overview of the material presented in the remainder of the thesis.

The main goal of this thesis is to develop and assess a robust unsupervised framework for semantic inference from composite audio signals. Our semantic inference approach will focus on the detection of audio scenes and their grouping into meaningful clusters. To perform both the audio scene segmentation and grouping, we choose for a two-step approach involving mid-level semantic descriptors. The main contributions reported in this thesis and resulting from pursuing the abovementioned goal and objectives can be defined as follows:

- Unraveling the problem of semantic inference from composite audio signals, by discussing both the supervised and unsupervised approach and addressing issues like reliability and scalability related to the application scope and inferred semantics,
- Mapping the abovementioned problem onto the problem of text document analysis and drawing cross-domain parallels to the relevant measurements needed for semantic inference in the audio domain,

- Introducing and assessing generic, unsupervised methods for
  - extracting mid-level semantic descriptors from composite audio that correspond to the concept of (key)words in text document analysis,
  - segmenting a composite audio track into audio scenes based on mid-level semantic descriptors,
  - grouping audio scenes into clusters corresponding to semantically meaningful categories based on mid-level descriptors,
- Unraveling the possibilities for combining supervised and unsupervised semantic inference from composite audio to benefit from the best of the two worlds,
- Expanding the ideas mentioned above that are defined for a content-based analysis of a single audio document onto a broader problem of audio document matching and clustering.

In view of the above, we first provide in Chapter 2 an overview of the related existing ideas and algorithms in the field of content-based audio analysis. This is followed by a general introduction of the main underlying idea of our envisioned realization of the scheme in Fig.1.4.

Chapter 3 addresses the fundamental step in any content-based audio analysis approach, namely feature selection and extraction. The suitability of a feature is measured based on its capability to reveal mid-level semantic descriptors from a composite audio signal, and to enable meaningful comparison of these descriptors and the audio scene detection and grouping processes based there on.

Chapter 4 presents our approach to the extraction of mid-level semantic descriptors, which follows the analogy to text document analysis. These descriptors are discovered in a similar way as the keywords are identified in a text document. In this way, an audio signal is divided into elements which can be intuitively explained as *audio words* and *audio keywords*.

In Chapter 5 we again build on ideas from text document analysis as well as the proven concepts from video content segmentation to develop our approach to audio scene detection. This approach is based on a novel semantic affinity measure that evaluates the coherence of the audio content semantics over time based on the relative temporal distribution of mid-level semantic descriptors with respect to each other.

After the audio scenes are detected, we use the approach introduced also in Chapter 5 to group them into meaningful clusters. The approach is based on the concept of

co-clustering that effectively makes use of the same mid-level semantic descriptors as those used in the previous segmentation step.

Chapter 6 revisits the goal of this thesis and the approach we proposed to reach this goal. Then, we present our views on the possibilities to expand the proposed approach in order to enable general audio search and management applications. We search for such possibilities by focusing on combining the unsupervised and supervised approaches, and on expanding the concept of document-specific audio onto a broader domain of audio document clustering and retrieval.

Chapter 2

Framework for Content Discovery from Composite Audio

In this chapter, we first discuss the previous work related to content-based analysis of composite audio, and analyze the advantages and disadvantages of the existing methods regarding their reliability and scalability in terms of the inferred semantics. Then, we propose our framework for content discovery from composite audio, position it with respect to the previous work, and present an implementation of this framework targeting the detection of audio scenes and grouping them into meaningful clusters.

2.1 Related Work

To infer the semantics from audio signals and bridge the semantic gap, considerable research effort has been invested in developing the theories and methods for content-based audio analysis. In general, most of these works can be located in one or more blocks indicated in Fig. 2.1, including the general processes of *audio segmentation*, *audio classification*, and *audio retrieval*.

Parts of this chapter are based on the following publications (also to be found in the list of references):

- Lu, L., Cai, R., Hanjalic, A. "Towards a Unified Framework for Content-based Audio Analysis," *Proc. 30th Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. II, 1069-1072, 2005
- Cai, R., Lu, L., Hanjalic, A. "Unsupervised Content Discovery in Composite Audio," *Proc. 13th ACM Int'l Conf. on Multimedia*, 628-637, 2005
- Cai, R., Lu, L., Hanjalic, A., Zhang, H.-J., and Cai, L.-H. "A Flexible Framework for Key Audio Effects Detection and Auditory Context Inference," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14, No. 3, 1026 – 1039, 2006

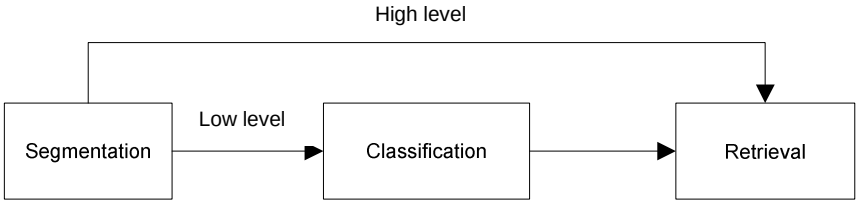


Fig. 2.1. Block scheme representing the most prominent classes of content-based audio analysis algorithms. The arrows indicate the typical causal relations between the blocks.

As indicated in the scheme in Fig. 2.1, *audio segmentation* provides inputs to all other blocks. We distinguish between two basic segmentation levels. At the first (*low*) level, an audio signal can be divided into elementary segments, also referred to as *audio frames*. Due to their short duration (typically around 10-50 ms) and the assumption that their signal properties can be considered stationary, audio frames provide a standard framework for feature extraction, that is, for feature-based representation of an audio signal. In case a compromise is required between the resolution of feature-based audio representation and the computational efficiency of the analysis processes based on this representation, longer segments (e.g. up to several seconds long) containing series of audio frames can be used as well. In that case, the feature vector of the longer segment can be inferred from the feature values measured within individual audio frames contained therein, and then adopted for the subsequent content-based audio analysis steps.

At the second (*high*) level, we can divide an audio signal into meaningful, that is, self-consistent and semantically coherent segments that can serve as the objects of audio or multimedia retrieval. Referring to the definitions provided in Section 1.1, this type of audio segmentation corresponds to the semantic inference process at the semantic structure level. While considerable effort has been invested in developing methods for detecting meaningful segments in text [Beeferman et al. 1999] and video documents [Kender and Yeo 1998][Hanjalic et al. 1999], much less has been done regarding the development of reliable high-level audio segmentation methods.

Audio classification associates semantic indexes (also referred to as *labels*) with audio signals. This association is typically inferred at the level of semantic concepts (classes). Audio classification, which can also be defined as *audio indexing*, *audio categorization* or *audio recognition*, can be performed at different content abstraction and complexity levels. In this thesis we distinguish among three main levels, namely

the *basic*, *mid*- and *high*-level. While the basic level covers the elementary audio categories, like speech and music, typical examples of mid-level semantic concepts include *audio effects*, such as *applause*, *cheering*, *ball-hit*, *whistling*, *car engine running*, *siren*, *gun-shot*, *instrumental solo*, *guitar sequence*, and *explosion*. High-level concepts are characterized by even higher semantic abstraction and signal complexity. Examples of such concepts are audio scenes like *action scene* in a movie, or a *game* in a tennis match. Typically, a high-level semantic concept can be characterized by a specific combination and sequence of mid-level concepts [Baillie and Jose 2003][Lu et al. 2005].

Content-based audio retrieval has the objective of providing access to a large data corpus based on an input query. The query is usually in a textual form (i.e. *query-by-text*). For example, a user may use the text term “applause” to search for all audio clips containing the corresponding sound. Clearly, this retrieval strategy is directly enabled by the results obtained from audio classification, as the text-based search can be performed on the labels assigned to the audio segments contained in the collection. An alternative paradigm is *query-by-example*, with an audio clip as a query. For instance, one can search for applauses by providing an “applause” sound as example to the system, or search for a song by simply singing or humming its melody.

In the remainder of this section, we will address each block in Fig. 2.1 in more detail regarding its realization possibilities and in view of the previous work related to it.

2.1.1 Audio Segmentation

Early works on audio segmentation (e.g. [Saunders 1996][Zhang and Kuo 1999]) were strongly related to audio classification. The proposed methods apply a sliding window of a pre-specified length to obtain a set of basic segments that can be further classified individually into predefined classes (e.g. speech, music). Then, the basic segments can be concatenated into longer segments of a particular class (e.g. speech segments, music segments), usually after smoothing out the outliers in the labeled segment sequence first. More complex modeling and classification strategies were applied in a number of methods aiming at dividing speech streams into segments corresponding to different speakers. If a speaker was pre-registered, traditional speaker identification algorithms [Brummer 1994] can be used for this purpose. However, in many applications, speakers are unknown *a priori*. To deal with this problem, several approaches were proposed dealing with unsupervised speaker segmentation and clustering. [Cohen and Lapidus 1996] studied the scenario of discriminating between speakers in a telephone-line signal. They approached the problem using Hidden Markov Model (HMM) and by

assuming that the number of speakers was limited to two. Opposed to this, [Wilcox et al. 1994] used no knowledge about the speakers when proposing an HMM-based speaker segmentation algorithm based on an agglomerative clustering method. [Mori and Nakagawa 2001] also addressed the problem of speaker segmentation without prior information on the speakers available. In [Gish et al. 1991][Siu et al. 1992], a system was proposed to separate the traffic control speech and pilot speech using a Gaussian Mixture Model (GMM). Further, [Chen and Gopalakrishnan 1998] presented an approach to detect changes in speaker identity, as well as in environmental and channel conditions, by using the Bayesian information criterion (BIC). While these approaches usually work offline and are computationally expensive, attempts were also made to design a real-time speaker segmentation approach [Lu and Zhang 2002].

An alternative class of approaches relied on a direct online analysis of features, where longer audio segments were defined to coincide with a consistent feature behavior. Examples are the method of [Venugopal et al. 1999] to segment an audio stream in terms of gender, speech, music and speaker, and of [Sundaram and Chang 2000] for segmentation into “computable audio scenes”.

The abovementioned approaches to audio segmentation is effective in identifying basic audio categories (e.g. speech, music, and noise). However, these approaches are not suitable for inferring higher-level semantic descriptors, such as those marking the *logical story units* [Hanjalic et al. 1999], which may be characterized by complex and strongly varying combinations of basic audio categories. Since the mentioned approaches are sensitive to such content diversity, their deployment for segmentation at a higher abstraction level typically results in an over-segmentation.

2.1.2 Audio Classification

Fig. 2.2 shows a general classification scheme, which is typically composed of two main steps: *supervised learning* and *inference*. In the supervised learning step, a model of each semantic class is built based on a set of training data, and with a specific learning scheme. Then, in the inference step, a new, unseen collection of data is associated with a semantic label, the model of which best resembles the properties of the data. Various schemes and mechanisms have been employed so far for realizing both the learning and inference steps. These schemes include sets of heuristic rules, vector quantization (VQ), k-nearest neighbor (kNN), decision tree, Bayesian network, artificial neural network (ANN), Gaussian mixture model (GMM), support vector machine (SVM), and hidden Markov model (HMM). More information about these schemes can be found in [Duda et al. 2000][Hastie et al. 2001].

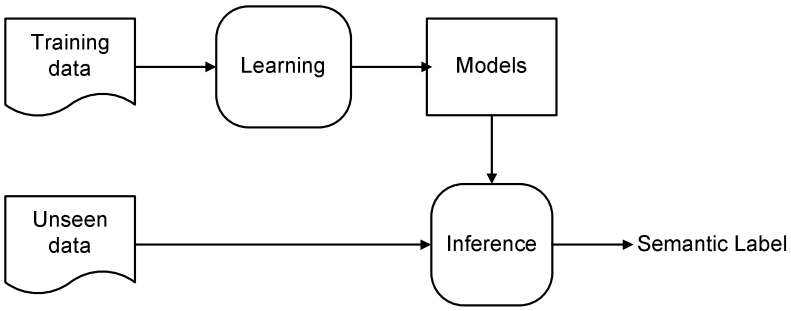


Fig.2.2. An illustration of a general classification scheme

In the following, we briefly discuss the results of the efforts invested so far in the field of audio classification to realize the scheme in Fig. 2.2 and for each of the three abstraction and complexity levels identified in the previous section to characterize a semantic concept.

2.1.2.1 Basic Audio Classification

Earliest audio classification attempts considered the distinction among basic audio types, such as speech, music and noise. In [Saunders 1996], a sliding-window based speech/music classifier for radio broadcast was presented. The authors reported the classification accuracy of up to 98%, obtained with a wide sliding window (2.4s). Working on the same problem, [Scheirer and Slaney 1997] introduced more features for audio representation and performed experiments with different classification methods, including GMM, kNN, and ANN. When using the same basic setting as in [Saunders 1996], the reported error rate was 1.4%. Then, [Kimber and Wilcox 1996] increased the number of classes and proposed a methodology based on HMM to classify audio recordings of meeting discussions into speech, silence, laughter, and non-speech sounds. In [Zhang and Kuo 1999], pitch tracking methods were introduced to divide audio recordings into songs and speech, based on a heuristic model that reached the accuracy of above 90%. [Srinivasan et al. 1999] proposed an approach to classify audio signals that consist of mixtures of speech, music and environmental sounds. The reported classification accuracy was above 80%. More recently, [Lu et al. 2001] presented a hybrid method which combines VQ and a rule-based method with multiple classifying steps to distinguish among speech, music, environmental sound, and silence. The accuracy of above 96% was reported. [Lu et al. 2003] further expanded this work to consider more classes, such as pure-speech and noisy speech.

2.1.2.2 Mid-level Audio Classification

One of the first attempts to audio classification at a higher abstraction level than the basic audio types was made by [Pfeiffer et al. 1996]. There, a method was presented to detect audio effects, such as *gunshot*, *explosion* and *cry*, in a given 30ms audio segment. [Xiong et al. 2003] also presented an approach to detecting both the basic audio types and some audio effects, such as *applause*, *cheering*, *music*, *speech*, and *speech with music*, for the purpose of highlights extraction from baseball, golf and soccer games. Other examples of methods targeting the extraction of various audio effects can be found in [Moncrieff et al. 2001][Xu et al. 2003][Cheng et al. 2003].

Several issues play a role in audio effect detection in audio signals, and need to be resolved in order to secure reliable classification. The most important issues can be described as follows:

(1) Audio effect detection in a long, continuous audio signal is typically approached by applying a sliding window of a given length (e.g. 0.5 seconds) to the signal. The audio segment captured by the window at a given time stamp is then used as the basic unit to be associated with an audio effect. An important implicit assumption here is that each segment corresponds to one and only one semantic class. However, a sliding window is often either too short to capture one complete audio effect, which leads to over-segmentation, or too long and captures several audio effects within one segment.

(2) The targeted audio effects are usually sparsely distributed over the signal, and there are plenty of non-target sounds that are to be rejected. Most existing approaches assume having a complete set of semantic classes available, and classify each audio segment into one of these classes. Other methods use thresholds to discard the sounds with low classification confidence [Cheng et al. 2003]. However, the setting of thresholds required in such an approach becomes troublesome for a large number of effects.

(3) Audio effects are usually related to each other. For example, some audio effects such as *applause* and *laughter* are likely to occur together in a sequence, while others are not. Taking into account the transition (co-occurrence) relationships between audio effects is therefore likely to improve the detection of each individual sound.

To investigate the possibilities for effectively resolving and exploiting the abovementioned issues when designing algorithms for audio effect detection, we elaborate on our previous approach proposed in [Cai et al. 2006] as an example. In this hierarchical probabilistic framework, *key audio effects* are searched for.

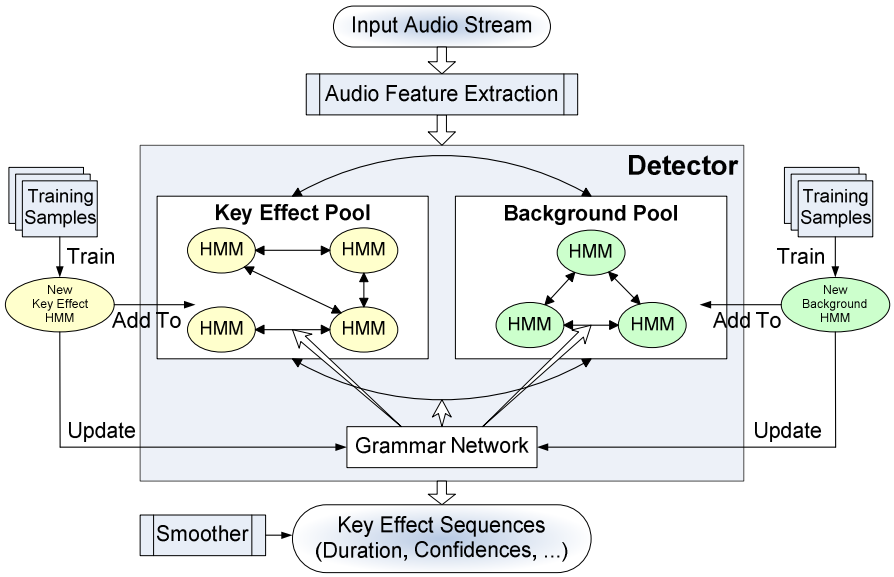


Fig. 2.3 The hierarchical probabilistic framework for key audio effect detection, consisting of three main parts: key audio effect pool, background sound pool, and grammar network [Cai et al. 2006]

As illustrated in Fig. 2.3, an HMM model is first built for each key audio effect based on a complete set of audio samples, and the defined models compose the *Key Audio Effect Pool*. Then, comprehensive *background* models are also established to cover all non-target sounds that complement the targeted key effects. Thus, the non-target sounds would be detected as background sounds and excluded from the target audio effect sequence. Moreover, a higher-level probabilistic model is used to connect these individual models with a *Grammar Network*, in which the transition probabilities among various audio effects and background sounds are taken into account for finding the optimal audio effect sequence. Thus, for a given input audio stream, the optimal audio effect sequence is found among the candidate paths using the Viterbi algorithm, and the location and duration of each key audio effect in the stream are determined simultaneously, without the need for an initial pre-segmentation of the audio data stream. In the following, both the learning and inference step are discussed in more detail.

A. Classifier Learning

We modeled each key audio effect and background sound using HMMs, since HMM provides a natural and flexible way for modeling time-varying process [Rabiner 1989]. The main issue that needs to be resolved for an HMM is the parameter selection, which includes i) the optimal model size (the number of states), ii) the number of Gaussian mixtures for each state, and iii) the topology of the model.

To select the model size, one needs to balance the number of hidden states in the HMM and the computational complexity in the learning and inference processes. In general, a sufficient number of states are required to describe all the significant behavioral characteristics of a signal over time. However, when the number of states increases, the computational complexity grows dramatically and more training samples are required. Unlike speech modeling, in which the basic units such as tri-phones could be adopted to specify the number of states, general key audio effects lack such basic units. A clustering-based method was proposed in [Zhang and Kuo 1998][Reyes-Gomez and Ellis 2003] to estimate a reasonable number of states (model size) per audio effect. The clustering step was realized through an improved, unsupervised *k*-means algorithm, and the obtained number of clusters is taken as the model size.

The number of Gaussian mixtures per state is usually determined experimentally. We adopt 32 Gaussian mixtures for each state in the HMM. This number is larger than those used in other related methods in order to secure a sufficient discriminative ability of the models to identify a large diversity of audio effects in general audio streams.

The most popular HMM topology is the left-to-right or the fully connected one. The left-to-right structure only permits transitions between adjacent states, while the fully connected structure allows transitions between any states in the model. Different topologies can be used to model audio effects with different properties. For instance, for key audio effects with obvious time-progressive signal behavior, such as *car-crash* and *explosion*, the left-to-right structure should be adopted, while for audio effects without distinct evolution phases, such as *applause* and *cheering*, the fully connected structure is more suitable.

Regarding the background sound modeling, a straightforward approach is to build a large HMM, and train it with as many samples as possible. However, background sounds are very complex and diverse, and their features are typically widely scattered in the feature space, so that both the number of states and the Gaussian mixtures per state of such a HMM must be exceptionally large to secure a representation of all possible background sounds. As an alternative, we modeled the background sounds as a set of subsets of basic audio classes, including speech, music, and noise, with 10 states and 128 Gaussian mixtures per state for each subset model. In this way, the

training data for each subset model would be relatively limited, and the training time would be reduced. Another advantage of building these subset models is that they could provide additional useful information for semantic inference at higher abstraction levels. For example, *music* is usually used in the background of movies, and *speech* is the most dominant component in talk shows.

The Grammar Network in Fig. 2.3 is an analogy to a language model in speech processing. It organizes all the HMM models for continuous recognition. Two models are connected in the Grammar Network if the corresponding sounds are likely to occur after each other, both within and between the key audio effect pool and the background sound pool. For each connection, the corresponding transition probability is taken into account when finding the optimal effect sequence from the input stream.

The transition probabilities between two models can be statistically learned from a set of training data. If no sufficient training data are available, a heuristic approach can be deployed as an alternative. For instance, our approach in [Cai et al. 2006] is based on the concept of *Audio Effect Groups*, where an audio effect group can be seen as a set of audio effects that usually occur together. The approach is based on the assumptions that 1) only audio effects in the same group can occur subsequently, 2) there should be background sounds between any two key audio effects belonging to different groups, and 3) the transition probability is uniformly distributed per group. An example Grammar Network with audio effect groups indicated as G_1-G_k is illustrated in Fig. 2.4

B. Probabilistic Inference

Based on the learned classification framework, the *Viterbi* algorithm can be used to obtain the optimal state sequence from the continuous audio stream, as:

$$s_{optimal} = \arg\max_{s} \Pr(s \mid M, O) . \tag{2.1}$$

Here, s is the candidate state sequence, M represents the hierarchical framework, and O is the observation vector sequence. In terms of practical realization of this classification scheme, the corresponding state and its log-probability are obtained first for each audio frame. Then, a complete audio effect or background sound can be detected by merging adjacent frames belonging to the same sound model. Before this merging step, a smoothing filter is applied to remove the classification outliers in the sequences of consecutive frames. The final classification confidence can be measured by averaging the log-probabilities of the classified audio frames. In addition, the starting time stamp and duration of each sound occurrence are obtained, by taking the starting and ending time stamp of the first and last audio frame in the sequence, respectively.

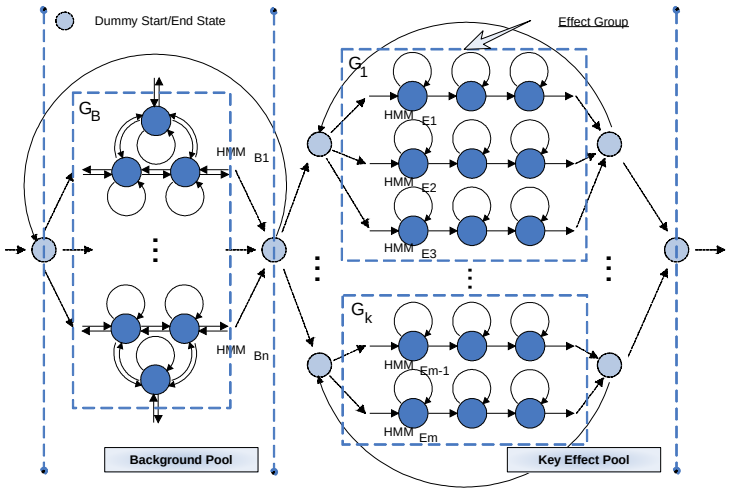


Fig. 2.4. An illustration of the *Grammar Network* with *Audio Effect Groups*, where G_k is the k^{th} *Effect Group* and G_B is the *Background Sound Pool*. For convenience, all key audio effect models are presented as 3-state left-to-right HMMs, and all the background models are denoted as 3-state fully connected HMMs. The dummy start and end states are used to link models [Cai et al. 2006]

2.1.2.3 Towards a Hierarchy of Semantic Concepts

Based on the obtained audio effect sequence, methods can be developed to perform audio classification at a higher abstraction level. While high-level semantic concepts can generally also be detected by directly working with the features as mentioned above, it has been shown that using audio effects as an intermediate classification level can lead to more effective indexing at higher abstraction levels [Lu et al. 2005].

Some example methods for inferring high-level audio semantics directly from the features include [Peltonen et al. 2002] and [Liu et al. 1998]. [Peltonen et al. 2002] built kNN and GMM classifiers to classify audio scenes into 26 pre-defined semantic categories. In [Liu et al. 1998], an ANN is developed to classify TV programs into five categories, namely commercials, basketball games, football games, news reports, and weather forecasts. However, the features may vary significantly among various audio samples belonging to the same semantic class, and thus may lead to unsatisfying detection/classification performance in practice.

The usability of audio effects to facilitate the semantic inference at the level of audio scenes can be derived from the fact that audio scenes with similar semantics are typically characterized by a number of same or similar audio effects. For instance, *cheering* and *laughter* are usually associated with humor scenes in comedies, and *explosion* and *gun-shots* often indicate violence scenes in action movies. Working in this direction, [Baillie and Jose 2003] presented an approach to event indexing in soccer broadcasts that first detects six content classes capturing various types and levels of crowd response during a soccer match. The audio patterns associated with each content class are modeled using a HMM model. Given a classified audio sequence, a simple rule-based decision process is developed to detect an event in each 'event window'. [Rui et al. 2000] presented an approach to extract highlights from the sound track of a baseball match. To deal with a high complexity of such an audio track, first the speech endpoint detection in noisy environment was developed. Then, energy and pitch statistics are computed for each speech segment. Gaussian fitting, kNN, and SVM were applied to detect portions of excited speech. Finally, some sports-specific sounds, e.g., baseball hits, are also detected by developing a directional template matching approach based on the characteristics of sub-band energy features. The detected mid-level results are further probabilistically fused to obtain final highlighting segments. [Xu et al. 2003] also worked on soccer game indexing. In this work, SVMs are first built to detect audio effects, such as *whistling* and *ball-hit*, based on audio frames of 20 ms. Then, a set of heuristic rules are used to infer the events in soccer games. An example of such rules is "if double whistling, then Foul or Offside". [Moncrieff et al. 2001] presents an approach to movie indexing. They first detect several key audio effects, e.g., *sirens*, *gun shots*, etc., using classifiers like decision tree and SVM. Then they concentrate on the extraction of complex audio scenes that are meant to coincide with dramatic movie segments, such as *car chase* and *violence*. Experimental results on movie audio tracks showed a classification accuracy of 88.9%. Another approach with a similar objective was presented in [Cheng et al. 2003]. In this work, sounds like *car-racing*, *siren*, *gun-shot*, and *explosion* are first identified using HMMs. Then, GMMs are used to learn the relationships between the audio effects and the higher-level semantics of audio scenes, and so to identify violent scenes in action movies.

Although the abovementioned and other related approaches actively employ audio effects as intermediate results for high-level semantic inference, the employed inference schemes usually do not reach beyond a set of relatively simple heuristic rules [Xu et al. 2003][Baillie and Jose 2003], or statistical classification [Moncrieff et al. 2001] [Cheng et al. 2003]. Heuristic inference is straightforward and can be easily applied in practice. However, it is usually laborious to find a proper rule set if the situation is complex. For example, the rules usually involve many thresholds which are

difficult to set, some rules may be in conflict with others, and some cases may not be well-covered. People are used to designing rules from a positive view but ignoring negative instances, so that many false alarms are introduced although high recall can be achieved. In the classification-based methods that apply statistical learning, the inference performance relies highly on the completeness and the size of the training samples. Without sufficient data, a positive instance not included in the training set will usually be misclassified. Thus these approaches are usually prone to high precision but low recall. Further, it is inconvenient to combine prior knowledge into the classification process in these algorithms.

To integrate the advantages of heuristic and statistical learning methods, we proposed a Bayesian network-based approach in [Cai et al. 2006]. A Bayesian network [Heckerman 1995] is a directed acyclic graphical model that encodes probabilistic relationships among nodes which denote random variables related to semantic concepts. A Bayesian network can handle situations where some data entries are missing, as well as avoid the overfitting of training data [Heckerman 1995]. Thus, it weakens the influence from unbalanced training samples. Furthermore, a Bayesian network can also integrate prior knowledge by specifying its graph structure.

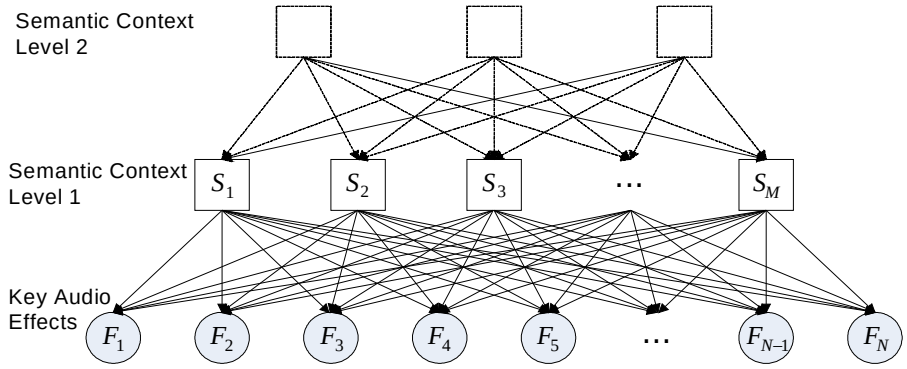


Fig. 2.5. An example of a Bayesian network for audio context inference: arcs are drawn from cause to effect. Following the convention, discrete variables are represented as squares while continuous variables are indicated as circles. Furthermore, observed variables are shaded, while hidden variables are not [Cai et al. 2006]

Fig. 2.5 illustrates the topology of an example Bayesian network with three layers. Nodes in the bottom layer are the observed audio effects. Nodes in higher layers denote high-level semantic categories, such as audio scenes, with increasing abstraction and complexity. In Fig. 2.5, the nodes in adjacent layers can be fully connected, or partially connected based on the prior knowledge of the application domain. For instance, if it is known *a priori* that some audio effects have no relationships with a semantic class the links from that class node to those effect nodes could be removed. A Bayesian network with a manually specified topology utilizes human knowledge in representing the conditional dependencies among nodes, thus it can describe some cases that are not covered in the training samples.

The nodes in the upper layers are usually assumed to be discrete binaries, which represent the presence or absence of a corresponding semantic class, while the nodes in the bottom layer produce continuous values of a Gaussian distribution

$$p(F_i \mid \mathbf{pa}_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \qquad (1 \le i \le N) \tag{2.2}$$

where F_i is a 2-dimensional observation vector of the i^{th} audio effect and is composed of its normalized duration and confidence in a given audio scene. The conditional argument $\mathbf{pa}_i$ denotes a possible assignment of values to the parent nodes of F_i, while $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean and covariance of the corresponding Gaussian distribution. In the training phase, all these conditional probability distributions are uniformly initialized and then updated by maximum likelihood estimation using the EM algorithm. In the inference process, the junction tree algorithm [Huang and Darwiche 1996] can be used to calculate the occurrence probability of each semantic class. Here, given the information on audio effects in each audio scene in the form of posterior probabilities (2.2), an audio scene, being at a higher abstraction level, can be classified into the c^{th} semantic class using the MAP criterion:

$$c = \arg\max_j \Pr(S_j \mid \mathbf{F}) \quad 1 \le j \le M \quad \textit{where} \quad \mathbf{F} = \{F_1, F_2, \cdots, F_N\} \tag{2.3}$$

With this scheme, human knowledge and machine learning are effectively combined to perform high-level semantic inference. In other words, the topology of the network can be designed according to the prior knowledge of an application domain, and the optimized model parameters can then be estimated by statistical learning.

2.1.3 Audio Retrieval

Audio retrieval aims at retrieving sound samples from a large corpus based on their relation to an input query. Depending on the type of a query, two general audio retrieval strategies can be distinguished. Using the first strategy based on a text query, a text label (e.g. “applause”) is submitted to find all audio clips in the corpus that are associated with that label. Clearly, this strategy relies on audio classification as the initial step to label the corpus and enable text-based audio retrieval. The other strategy, also referred to as *query-by-example*, uses an example sound as an input query. Then, based on the features extracted from the sounds in the corpus and from the query sound, as well as the similarity criteria employed, sounds from the corpus can be retrieved that best match the query sound. Since the realization possibilities of the first retrieval strategy have already been covered by the algorithms for audio classification in the previous section, we concentrate in this section on the query-by-example strategy and briefly review the most relevant previous work addressing this type of audio retrieval.

In one of the earliest content-based audio retrieval systems called ‘Muscle Fish’ [Wold et al. 1996], a statistical model including a Gaussian and a histogram model is employed to build a feature-based representation of a sound clip, using which the similarity between two clips can efficiently be measured. In order to speed up the search in a large database, the authors also built an index of the sounds based on acoustic features. It allows to quickly retrieve the desired sounds by requesting all the sounds whose feature values fall in the corresponding range.

In the audio search engine proposed in [Foote 1997], to be able to separate different sounds while remaining insensitive to unimportant variations, a tree-structured vector quantizer is built to divide the feature space into partitions (bins), optimally in the information-theoretical sense. Then, a “template” for each audio clip is built, which is actually a histogram indicating the vector counts in each bin. Euclidean or Cosine distances between the query template and corpus templates are employed, so that the audio clips in the corpus can be ranked correspondingly. Retrieval performance was evaluated on a corpus of simple sounds as well as a corpus of music excerpts. The best result is obtained with a supervised quantization tree with 500 bins and a cosine distance measure.

[Smith et al. 1998] presented a new search scheme - “active search” - to quickly search through broadcast audio data and to retrieve known sounds using 120 reference templates. Active search reduces the number of candidate matches between a reference and the test template, while still providing optimal retrieval performance. The template is built based on a histogram of zero-crossing features, which is claimed to be robust

against digitization noise and white noise addition down to 20 dB SNR (signal-to-noise ratio).

[Li 2000] presented a new method for audio classification and audio retrieval, called *Nearest Feature Line* (NFL). NFL interpolates or extrapolates each pair of prototypes (audio samples) belonging to the same class by using a linear model. The feature line that passes through two prototypes provides generalized information about the variants between these two sounds, i.e. possible sounds derived from the two prototypes. Opposite to the commonly used NN approach, in which classification is performed by comparing the query to each prototype individually, the NFL makes use of information provided by multiple prototypes per class. An evaluation on Muscle Fish audio database of 409 sounds shows that NFL outperforms both kNN and Nearest Center.

[Li and Khokhar 2000] presented another approach to content-based audio information retrieval, which is based on the multi-resolution decomposition property of the discrete wavelet transform. The wavelet decomposition of an audio signal highly resembles its decomposition in sound octaves. A hierarchical indexing scheme is constructed using statistical properties of the wavelet coefficients at multiple scales. A variant of B-tree data structure is used as an indexing structure, where the height of the tree corresponds to number of sub-bands and the nodes of each level corresponds to clusters in the corresponding sub-band. The performance of the proposed systems is experimentally evaluated on 418 audio clips. The prototype system yields high recall ratios (higher than 70%) for sample queries with diverse audio characteristics.

2.1.4 Other Relevant Previous Work

Next to the approaches presented in the previous sections, a large number of other ideas and methods have been proposed in recent literature that do not directly fall into the scope of audio segmentation, classification or retrieval, but are closely related to them. A good example of such work is the one on computational audio scene analysis (CASA), which attempts to separate and represent a continuous sound mixture (a composite audio) as a set of independent sources, or to estimate a number of distinct events therein. As a fundamental work in this direction, [Bregman 1990] first reports a number of theoretical foundations and experimental investigations that addressed the psychoacoustic aspects of the human listening behavior. These experiments have inspired numerous efforts to build computational models for audio scene analysis mentioned before. These modeling approaches can be conceptually divided into two groups, namely the “data-driven” and “prediction-driven” ones. The “data-driven” approach is more frequently used. There, specific features (e.g. instantaneous

frequency, amplitude modulation, onsets and offsets) in the sound signal are extracted and then grouped into larger entities of perceptual events or sources, such as in [Westner 1998][Casey 1998]. However, data-driven approach usually interprets a given sound regardless of the context. As an alternative, a "prediction-driven" approach [Ellis 1996] sees the analysis as a process of reconciliation between the observed features and the predictions of the sound in the future. More recent work on computational audio scene analysis can be found in [Wang and Brown 2006].

Other previous work related to the material presented in this thesis include [Hanjalic and Xu 2005] and [Ma et al. 2002]. [Hanjalic and Xu 2005] present a computational framework for affective audiovisual content representation and modeling, where the expected transitions from one human affective state to another are represented by a curve in a two-dimensional (intensity-valence) affect space. Audiovisual content is treated here in an integral fashion by combining the features from both the visual and audio track together into a joint affect model. [Ma et al. 2002] proposes a computational attention model, which models a viewer's attention on a video sequence by integrating a set of visual, audio, and linguistic attention values, and subsequently assign an overall attention value to each video frame. With an application to video summarization, the video shots with high attention value, which are most likely to attract the viewer's attention, are chosen. Moreover, speech is further segmented into sentences, so that each segment of the video summary contains one or several complete sentences without any interruption within a sentence.

2.2 What Can We Learn From The Past?

While the ideas and methods described in the previous sections have been invaluable for the rapid development of the theory and practice of content-based audio (and multimedia in general) analysis and retrieval in the past years, there are a number of issues that have not been sufficiently addressed yet, and that can be identified as an obstacle for a broad deployment of the obtained research results in real-life applications. We identify and briefly explain these issues in the following paragraphs.

Insufficient scalability and narrow application scope: While the observable dominance of supervised learning approaches in the field has led to many exciting results of automatic semantic audio classification (Section 2.1.2), the applicability and scalability of such approaches in a realistic application scenario will likely be limited, not only because one has to work with pre-defined (and pre-trained) semantic concepts, but also because the upper performance limit of such approaches is defined by the capability of the training data to capture the entire content diversity of a particular

semantic concept. Regarding the former, in most real-life applications, it is difficult to list all audio elements and semantic categories that are possible to be found in data. For example, in the applications like pervasive computing and audio-supported surveillance, relevant audio effects are generally unknown in advance. Thus it is impossible to collect training data and learn proper statistical models in these cases. Regarding the latter, due to the high diversity and insufficient training data, the upper performance limits are regularly not high enough to provide a solution usable in general application domains.

In view of the disadvantages of supervised methods, a few recent works introduced unsupervised approaches into multimedia content analysis. For example, an approach based on time series clustering is presented in [Radhakrishnan et al. 2004] to discover "unusual" events in audio streams. In [Ellis and Lee 2004], an unsupervised analysis of a personal audio archive is performed to create an "automatic diary". However, these existing methods are in general either designed for some specific applications [Ngo et al. 2001][Xie et al. 2003], or only address some isolated components of the content-based audio analysis process chain [Ellis and Lee 2004][Radhakrishnan et al. 2004].

In view of the above, there is a lack of approaches capable of addressing the full processing chain (illustrated in Fig. 2.6, Section 2.3), and also capable of dealing with a wide (and unpredictable) range of (composite) audio signals and related applications. The need for such complete and generic approaches is considerable due to a typically high variety of audio content and search/retrieval applications in a general consumer or professional context. To work well in such a context, a content-based audio analysis mechanism needs to be based on solid generic principles and show constant high robustness over the entire broad application scope. This is in contrast to the current sub-optimal and impractical possibilities that rely on combining together a large number of dedicated narrow-scope solutions in different ways to address different content types and search/retrieval scenarios.

Insufficient coverage of the semantic space: The abovementioned dominance of supervised classification methods has led to an abundance of solutions targeting basic and mid-level semantic concepts, and most of them mainly established the more-or-less straightforward link between the features and the corresponding concepts. For instance, the audio effects such as *explosion*, *cheering* and *laughter* are directly modeled from the temporal and spectral signal properties [Cheng et al. 2003]. However, due to the difficulty of modeling higher-level concepts caused by high content diversity, these more challenging tasks have typically been approached in a rather simplistic fashion, like for instance, audio scene characterization solely based on the detection of a particular low- or mid-level semantic concept contained therein, without considering other effects present there and their relationship with respect to each other. As an

example, a highlight scene is usually detected if it contains sounds of excited speech, while a scene of foul or offside in soccer is detected if the sound of double whistling is found in the signal. More sophistication in inferring higher-level semantics can be introduced by applying hierarchical probabilistic approaches, like the one based on multijects and multinets [Naphade et al. 2001]. However, while this approach allows expansion of the analysis scope and compensates for detection uncertainties of individual semantic concepts, the links between the concepts still need to be modeled and trained *a priori* using supervised learning methods, which again brings us back to the issues discussed before.

Furthermore, previous approaches to content-based audio analysis target the detection of pre-selected semantic concepts only, and do not provide the possibility to obtain a complete description of the entire audio track. However, in some applications like pervasive computing, we may want to have the entire description (or a brief outline) of the audio track, and not just some selected temporal segments. In principle, complete information about the content of a given audio stream including the scene partitioning, their content semantics and interrelations, could be obtained by applying various supervised concept detectors together, provided that it is known *a priori* which concepts are likely to be found in the content. However, as discussed above, this approach is not practical and the required information is not available in a general case.

Pre-segmentation issue: In order to apply semantic inference techniques on audio data, the data often needs to be divided into segments of consistent signal-level and semantic properties. The existing work on content-based audio analysis is largely based on the assumption that audio data is pre-segmented prior to applying classification and other semantic inference approaches. For example, [Saunders 1996] performed audio classification on audio segments of 2.4 seconds; and [Liu et al. 1998][Cai et al. 2005] also assumed that audio scenes are manually pre-segmented. The pre-segmentation assumption reduces the practical applicability of content-based analysis considerably as manual segmentation is expensive and inflexible. On the other hand, there are hardly any robust automated audio segmentation mechanisms available (see Section 2.1.1 for an overview).

2.3 Audio Content Discovery: An Unsupervised Approach

In view of the discussions in previous sections, we define in this section a framework for content-based analysis of composite audio, in which we take into account the deficiencies of the existing works in the field. Different components of the framework will be introduced and explained in more detail in the following chapters.

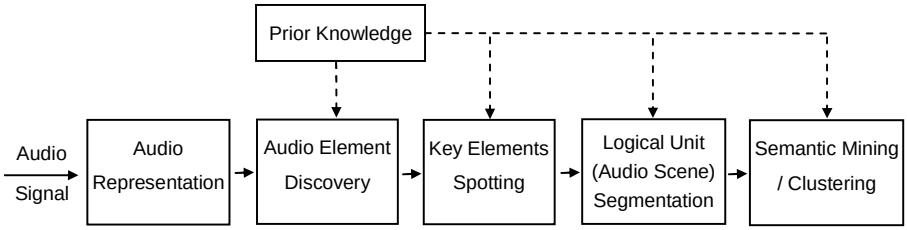


Fig. 2.6 Our proposed framework for content discovery from composite audio

2.3.1 Overview of the Proposed Framework

Recalling the discussion in Chapter 1, as opposed to a single-modal audio (e.g. pure music or speech), composite audio contains multiple audio categories such as speech, music and various noise sounds, which are either mixed together, or follow each other in a sequence. Because most of the audio data appearing in multimedia applications are composite, building a system for content-based composite audio analysis is likely to provide generic methods for semantic inference from audio data and support a wide variety of multimedia applications where this data plays a role.

Based on the discussion in Section 2.2 about the disadvantages of previous approaches, the framework we aim at developing in this thesis should satisfy the following conditions:

1) generalization and extensibility to support a wide variety of applications,
2) effective semantic inference from composite audio, and
3) sufficient coverage of the addressed semantic space to provide a complete description of the analyzed audio content.

With this in mind, we propose a framework for content-based analysis of composite audio as illustrated in Fig. 2.6.

In this framework, the input audio is first decomposed into *audio elements*. An audio element is a short temporal segment with coherent signal properties, such as speech, music, various audio effects and any combination of these. Then, *key audio elements* are selected, being the audio elements that are most indicative of the semantics (main underlying content) of the analyzed audio data segment. As an example, we could consider the audio clips containing the sounds of *laughter* and *applause* the key audio elements representing a *humor* scene in a typical situation comedy. Audio elements can be seen as analogies to words in text documents, and key audio elements are analog to

the keywords. According to the terminology convention we adopted in Chapter 1, we will refer to them as mid-level semantic descriptors.

Also based on the initial explanation of the thesis scope and related definitions provided in Chapter 1, once the (key) audio elements are discovered, we will employ them to divide an audio document into audio scenes and group these scenes into meaningful clusters. We show in this thesis that these scenes can effectively be characterized, detected, and grouped based on the audio elements they contain, just as the paragraphs of a text document can be characterized, detected and grouped using a vector of words and their weights. As it will be shown later in the thesis, introducing these mid-level descriptors enables us to split the semantics inference process into two steps, which leads to more robustness compared to inferring the semantics from features directly.

The semantic inference process described above is realized through the following main algorithmic modules:

- **Audio representation:** In this module, features representing the temporal and spectral properties of audio signals are extracted. Audio features are usually required to have enough discrimination capability regarding audio element extraction and deployment in subsequent analysis steps. The possibilities and guidelines for realizing the audio representation module are explained in more detail in Chapter 3.
- **Audio element discovery:** In this step, the input audio stream is decomposed into different audio elements. The data mining techniques and approaches deployed for this purpose are explained in detail in Chapter 4.
- **Key elements spotting:** Using the pool of the detected audio elements as input, we develop a mechanism deployed in this module to select the key audio elements.. Combined with audio element discovery, the details on the realization of this module are given in Chapter 4.
- **Audio scene segmentation:** The objective of this module is to detect boundaries between audio scenes. Compared to the mid-level semantic descriptors in the form of (key) audio elements, we consider the pointers to audio scene boundaries as high-level semantic descriptors. The theoretical fundamentals and realization details for the mechanism we developed for this module can be found in Chapter 5.
- **Semantic mining/clustering:** In this final step, the audio scenes are clustered together based on the audio elements they contain. This step can be seen as an unsupervised counterpart of the supervised approaches, such as those proposed in [Moncrieff et al. 2001][Cheng et al. 2003], where audio segments are classified as

humor and *violence* scenes based on the fact that they contained the sounds classified as *laughter*, *gun-shot*, and *explosion*. In this step, we will also investigate the grouping tendency and semantic affinity between audio elements, in order to obtain a better similarity measure between two audio scenes. The theory and algorithms used to develop this module are also explained in Chapter 5.

While the framework in Fig. 2.6 can be implemented in a supervised fashion, we choose in this thesis for an unsupervised approach to such realization, searching for the possibilities to compensate for the deficiencies of supervised methods and to provide a generic set of methodologies to support a variety of applications, as discussed in Section 1.3.3 and 2.2. Although there are a few unsupervised approaches to content-based audio analysis proposed in recent literature [Ellis and Lee 2004][Radhakrishnan et al. 2004], these existing methods are not meant to provide generic content analysis solutions, as they are either designed for specific applications, or only address some aspects of the scheme in Fig. 2.6.

2.3.2 Unsupervised Framework Implementation

Aiming at an unsupervised realization of the generic framework in Fig. 2.6, a novel unsupervised approach to content discovery of composite audio is proposed in this thesis, to automatically mine the audio elements, audio scenes and the relationship between them. The detailed flowchart of the proposed approach is given in Fig. 2.7(a). It consists of two major steps: I) audio elements discovery and key audio element spotting, and II) audio scenes detection and clustering. Both steps are unsupervised and domain- or application- independent. The approach also facilitates audio content discovery at different semantic levels, such as (mid-level) audio elements and (high-level) audio scenes.

The illustration of the proposed framework as given in Fig. 2.7(a) indicates the analogy to the standard scheme for topic-based text document categorization [Baeza-Yates and Ribeiro-Neto 1999] illustrated in Fig. 2.7 (b). In text analysis, a text document is first parsed to the sequence of words or phrases, which is similar to audio element discovery decomposing an audio document into audio elements. To indicate which words are more indicative of the semantics of the text document, the words are weighted based on their *term frequency* (TF) and *inverse document frequency* (IDF). Similarly, key audio elements can be detected in an audio signal by computing their importance relative to other audio elements detected in the signal. We therefore further refer to (key) audio elements also as *audio (key)words*. Subsequently, a text document can be segmented into smaller units (paragraphs) of consistent but unique content. This

step is similar to audio scene segmentation, where the discovered audio scenes can be seen as analogies to text paragraphs. Finally, each text document or each topic section can be represented by the words and keywords it contains, and the documents and sections can be clustered together based on their topics. This is again a direct analogy to audio scene clustering we aim at realizing in this thesis.

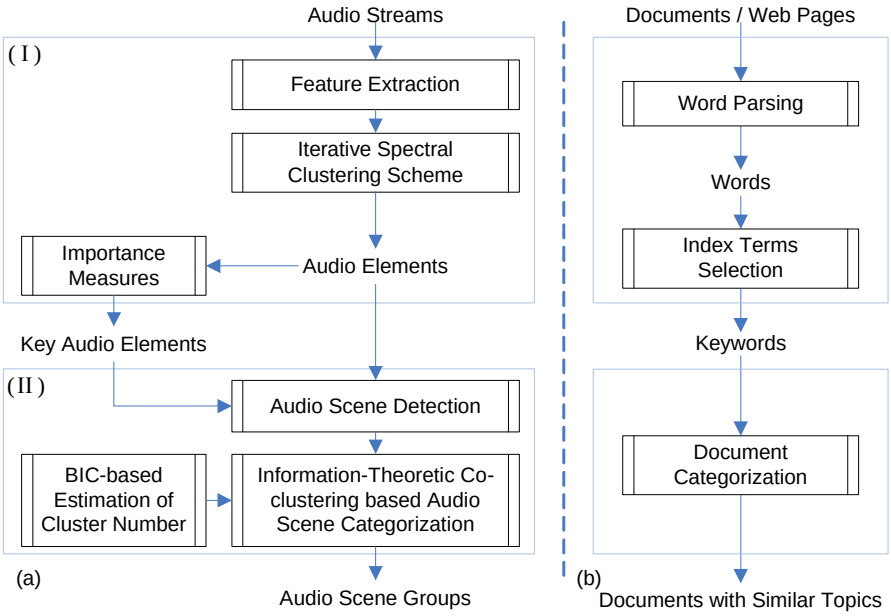


Fig. 2.7 (a) The flowchart of the proposed approach to unsupervised content-based audio analysis, which consists of two major parts: (I) audio element discovery and key element spotting; and (II) audio scene segmentation and clustering. (b) A comparable process of the topic-based text document categorization.

Regarding the technical implementation of the scheme in Fig. 2.7(a), we start with the assumption that the input into the scheme is a general composite audio stream. After feature extraction, an iterative spectral clustering method is proposed to decompose the composite audio into audio elements. Using this method, the segments with similar features in the audio stream are grouped into clusters that we adopt as audio elements. Then, following the same rationale underlying the TF and IDF definitions in text document analysis, we introduce a number of importance measures

and employ them to filter the obtained set of audio elements and select the key audio elements.

In the next step, the audio scenes are first detected by investigating the semantic affinity among various audio elements in the input audio. For this purpose, a novel semantic affinity measure is introduced. Then, the detected audio scenes are grouped into clusters by using an information-theoretic co-clustering algorithm, which exploits the relationships among various audio elements and audio scenes. Moreover, we propose a strategy based on the Bayesian Information Criterion (BIC) for selecting the optimal number of clusters for the co-clustering.

2.4 Summary

In this chapter we first discussed previous work related to composite audio analysis, and analyzed the issues that have not been sufficiently addressed yet. These issues are briefly summarized in the first column in Table 2.1. Based on these considerations, we proposed a framework for unsupervised content-based analysis of composite audio, which consists of five main components: audio representation, audio element discovery, key elements spotting, audio scene segmentation and scene clustering. While each of these components will be described in detail in the remaining chapters of this thesis, we summarize in the second column in Table 2.1 the main aspects of our approach helping us to optimally resolve the issues from the first column.

Table 2.1 Disadvantages of previous approaches together with our proposed solutions

Issues	Approach
insufficient scalability and narrow application scope	no application-/domain-specific prior knowledge considered, unsupervised approach
insufficient coverage of the semantic space	two-step semantic inference approach via mid-level semantic descriptors (audio elements), application of co-clustering to optimally exploit co-occurrence statistics among audio elements
pre-segmentation issue	robust automated segmentation at the audio element and audio scene level

Chapter 3

Feature Extraction

A fundamental step in content-based audio analysis is to obtain a representation of an audio signal in a feature space. In the context of this thesis, a feature set is considered suitable if it captures the temporal and spectral characteristics of individual elementary audio segments with a sufficient discriminative power to enable grouping of all segments belonging to a particular audio element, as well as the content discovery operations performed on the obtained audio elements, as described in Chapter 2.

In this chapter we give an overview of the typical audio features proposed in literature, select the features that we consider suitable for the content-based audio analysis approach presented in this thesis, and also propose some new features to be likely to cover as much as possible of the semantic content variance of composite audio signals in a general case. Finally, we also address the feature normalization and selection steps that are necessary to form a reliable feature vector serving as input into subsequent audio content discovery steps.

Parts of this chapter are based on the following publications (also to be found in the list of references):

- Lu, L., Zhang, H.-J., and Jiang, H. "Content Analysis for Audio Classification and Segmentation," *IEEE Trans. Speech Audio Processing*, 10(7), 504-516, 2002
- Lu, L., Zhang, H.-J., and Li, S., "Content-based Audio Classification and Segmentation by Using Support Vector Machines," *ACM Multimedia Systems Journal*, 8(6), 482-492, 2003
- Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. "Improve Audio Representation by Using Feature Structure Patterns," *Proc. 29th IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, 345-348, 2004.

3.1 An Overview of Audio Features

Table 3.1 gives an overview of the typical features employed in literature, and is based on 15 representative papers widely covering the fields of audio classification and audio retrieval. It should be noted that a number of these features are frequently used in various papers, and that regarding the terminology used there, the names of some features may be different in different papers. For example, the terms *spectral centroid* and *brightness* stand for one and the same feature, as well as the terms *sub-band power* and *band-energy ratio*.

In general, the audio features listed in Table 3.1 can be divided into *temporal* and *spectral* features that capture the temporal and spectral characteristics of an audio signal, respectively. Good examples of temporal features are zero-crossing rate (ZCR) and energy, while mel-frequency cepstrum coefficients (MFCC) and spectral centroid are typical spectral features.

Regarding the length of the audio segment from which features are extracted, a division into *frame-level* and *window-level* features can also be made. The features from the first class are extracted from individual audio frames. Some examples of frame-level features are ZCR, MFCC, and spectral centroid. The window-level features are extracted from a longer audio segment, comprising a number of consecutive frames and usually marked by applying a sliding window to the signal. While most audio features are extracted at the frame level, window-level features are mainly derived from the frame-level features by investigating their variation along the frames within the window, e.g., the mean, standard deviation, or other statistics derived from frame-level features. This expansion of the frame-level feature consideration from an individual frame to a series of consecutive frames proved to be useful in many applications, which indicates the importance of window-level features. Examples of this feature class are spectral flux and the features based on vocal contour and pitch contour [Liu et al. 1998].

Since the popularity of each feature may be seen as a first indication of its quality, and because the classification into frame-level and window-level features is particularly relevant for different steps in the content-based audio analysis approach discussed in this thesis, we reorganized the list of the features introduced in the previous table as shown in Table 3.2, and ranked them in a descending order depending on the number of times they are used in the representative set of 15 papers considered. There, for each feature we provide a “use score” indicating its popularity and a classification as either a frame-level or window-level feature. Multiple names indicated per row of the table stand for one and the same feature and/or its variants.

Table 3.1 An overview of typical audio features used in literature. The information in the table is based on 15 representative papers covering the most important aspects of content-based audio analysis and being indicative of the features typically employed for various approaches in the field.

Representative Reference	Features
[Saunders 1996]	zero crossing rate (ZCR), energy
[Scheirer and Slaney 1997]	4Hz modulation energy, percentage of low-energy frames, spectral rolloff, spectral centroid, spectral flux, ZCR, cepstrum resynthesis residual magnitude, pulse metric
[Zhang and Kuo 1999]	short time energy (STE), short time ZCR, short time fundamental frequency, spectral peak
[Tzanetakis and Cook 2000]	ZCR, root mean square (RMS), spectral rolloff, linear predictive coding (LPC), MFCC, harmonicity, pitch, spectral flux, spectral moments, spectral centroid
[Srinivasan et al. 1999]	ZCR, energy, sub-band energy, harmonic frequency
[Wold et al. 1996]	loudness, pitch, tone (brightness and bandwidth), cepstrum and derivatives
[Foote 1997]	MFCC, energy
[Li 2000]	total spectrum power, sub-band powers, brightness, bandwidth, pitch frequency, MFCC
[Li and Khokhar 2000]	wavelet decomposition
[Peltonen et al. 2002]	ZCR, energy, band-energy ratio (BER), spectral centroid, bandwidth, spectral rolloff, spectral flux, LPC, MFCC LPC-derived Cepstral coefficients (LPCC)
[Liu et al. 1998]	(1) nonsilence ratio, (2) volume standard deviation, (3) volume dynamic range, (4) frequency component of the volume contour around 4Hz, (5) pitch standard deviation, (6) voice-or-music ratio (VMR), (7) noise or unvoice ratio, (8) frequency centroid, (9) frequency bandwidth, (10–12) energy ratios of subbands 1–3
[Baillie and Jose 2003]	MFCC
[Xu et al. 2003]	ZCR, spectral power, MFCC, LPC, LPC-derived Cepstral coefficients (LPCC)
[Cheng et al. 2003]	volume, band-energy ratio, ZCR, frequency centroid, bandwidth, MFCC
[Xiong et al. 2003]	MFCC, MPEG-7 audio features

Table 3.2 An overview of features used in literature. The features from Table 3.1 are ranked according to their usage and are classified as either frame- or window level features. The last column indicates which of the features we considered in our approach, directly ("√") or indirectly ("•"). Different names of the same feature are put together.

Features	#Usage	Level	Used
short time energy, RMS, spectrum power, volume, loudness	10	frame-level	√
ZCR	8	frame-level	√
MFCC	8	frame-level	√
spectral centroid, brightness, frequency centroid	7	frame-level	√
bandwidth	5	frame-level	√
sub-band energy (distribution), sub-band power, band-energy ratio	5	frame-level	√
short time fundamental frequency, pitch, harmonic frequency	5	frame-level	•
LPCC or cepstrum	4	frame-level	
LPC	3	frame-level	
spectral rolloff	3	frame-level	
spectral peak	1	frame-level	•
spectral moments	1	frame-level	
harmonicity	1	frame-level	•
wavelet decomposition	1	frame-level	
MPEG-7 audio features	1	frame-level	
spectral flux	3	window-level	√
percentage of low-energy frames	1	window-level	√
clip-level features [Liu et al,1998]	1	window-level	•
4Hz modulation energy	1	window-level	
pulse metric	1	window-level	

To select the features to be used in the methods introduced in this thesis, an ideal approach would be through a general experiment investigating a suitability of a given feature in a general composite audio context. This approach is, however, not realistic without selecting a number of representative use cases, which would bring us back to a supervised approach. Instead, we choose to collect those robust, proven features, the effectiveness of which was shown in many different noisy, ambiguous use cases characteristic for unconstrained composite audio signals we address in this thesis.

To generate a good feature set, we first took the feature list from Table 3.2 as the basis and either eliminated those features which may be ineffective or inefficient to extract in our use context, or used them directly or indirectly, as we will illustrate by an example in the next section. The features we used directly or indirectly are indicated by a corresponding mark in the last column in Table 3.2. Then, we also added a number of features that complement those listed in Table 3.2.

In Section 3.2 and 3.3, we present all features used in this thesis organized as either the frame- or window-level features. Then, in Section 3.4, we briefly explain a standard unsupervised method for selecting optimal feature set per use case based on the Principle Component Analysis.

3.2 Frame-level Features

Referring to the feature list from Table 3.2, our set of temporal frame-level features include short-time energy (STE) and zero-crossing rate (ZCR), while the spectral features include sub-band energy ratio (BER), brightness, bandwidth, and Mel-frequency cepstral coefficients (MFCC). Regarding other features listed in Table 3.2, although LPC provides a good model for voiced speech and gives a good approximation to the vocal tract spectral envelope, it is less effective on those sounds which are not vocally produced, like music, noise, and various audio effects. Therefore, we did not use them in our approach. Moreover, there are also some features (as indicated by "•" in Table 3.2), which are not considered directly, but indirectly via a set of alternative features. For example, we do not use pitch or fundamental frequency directly, due to a large variation of pitch values within a given audio semantic class and the difficulty of multiple pitch detection in a polyphonic sound. Instead, we use the sub-band partial prominence and harmonicity prominence as alternative pitch-related features, for which the rationale and extraction method are explained in detail later in this section. Table 3.3 summarizes all temporal and spectral frame-level features used in our approach.

Table 3.3 The list of frame-level features used in this thesis

Feature Kind		**Feature list**
common features	temporal	ZCR, STE,
	spectral	BER, brightness, bandwidth, MFCC
proposed	spectral	sub-band partial prominence, harmonicity prominence

Prior to frame-level feature extraction, a composite audio signal is first converted into a general format, described by the following parameters: 16 KHz, 16-bit, and mono-channel. Then, it is pre-emphasized with parameter 0.98 (i.e. $H(z) = 1 - 0.98z^{-1}$) to equalize the inherent spectral tilt, and is further divided into frames of 25ms with 50% overlap. For extracting spectral features, the spectral domain is equally divided into 8 sub-bands in Mel-scale and then the sub-band features are extracted, including BER, MFCC, and sub-band partial prominence. All the above features are collected into a 29-dimensional feature vector per audio frame. The following paragraphs provide a detailed description of each frame-level feature used in this thesis.

3.2.1 Zero-Crossing Rate

Zero-Crossing Rate (ZCR) is defined as the relative number of times the audio signal crosses the zero-line within a frame:

$$ZCR = \frac{1}{2(N-1)} \sum_{m=1}^{N-1} \left| \mathrm{sgn}[x(m+1)] - \mathrm{sgn}[x(m)] \right| \tag{3.1}$$

where sgn[] is a sign function, $x(m)$ is the discrete audio signal, $m = 1 \ldots N$, and N is the frame length.

The ZCR is a computationally simple measure of the frequency content of a signal, and as such it is particularly useful in characterizing audio signals in terms of the *voiced* and *unvoiced* sound categories. For example, as speech signals are generally composed of alternating voiced and unvoiced sounds at the syllable rate, which is not the case in music signals, the variation in the ZCR values is expected to be larger for speech signals than for music signals. Due to its discriminative power in separating speech, music and various audio effects, ZCR is often employed in content-based audio analysis algorithms.

3.2.2 Short Time Energy and Sub-Band Energy Distribution

Short Time Energy (STE) is the total spectral power of a frame. In our approach, it is computed from the Discrete Fourier Transform (*DFT*) coefficients, as

$$STE = \sum_{k=0}^{K/2} |F(k)|^2 \tag{3.2}$$

Here, $F(k)$ denotes the *DFT* coefficients, $|F(k)|^2$ is the signal power at the discrete frequency k, and K is the order of *DFT*. In our approach, the logarithmic value of this power is computed to get a measure in (or similar to) decibels.

Similar to ZCR, STE is also an effective feature for discriminating between speech and music signals. For example, there are more silence (or unvoiced) frames in speech than in music. As a result, the variation of STE in speech is in general much higher than in music. However, STE considers only the overall energy of one audio frame. To further exploit the energy information, the spectral energy distribution (SED) (i.e. band energy ratio (BER)) is computed. This distribution can be obtained by dividing the frequency spectrum into sub-bands, and by computing for each sub-band j the ratio D_j between the energy contained in that sub-band and the total spectral power of the frame,

$$D_j = \frac{1}{STE}\sum_{L_j}^{H_j} |F(k)|^2 \tag{3.3}$$

where L_j and H_j are the lower and upper bound of sub-band j respectively.

Since the spectrum characteristics are rather different for sounds produced by different sources (e.g. human voice, music, environmental noise), the STE and SED features have often been used for audio classification [Saunders 1996][Srinivasan et al. 1999][Liu et al. 1998], and, in particular, for discriminating between different audio effects [Wold et al. 1996][Cai et al. 2003].

3.2.3 Brightness and Bandwidth

Brightness and bandwidth are related to the first- and second-order statistics of the spectrum, respectively. The brightness is the centroid of the spectrum of a frame, and can be defined as:

$$w_c = \frac{\sum_{k=0}^{K/2} k\,|F(k)|^2}{\sum_{k=0}^{K/2} |F(k)|^2} \tag{3.4}$$

Bandwidth is the square root of the power-weighted average of the squared difference between the spectral components and the frequency centroid:

$$B = \sqrt{\frac{\sum_{k=0}^{K/2}(k - w_c)^2 \left|F(k)\right|^2}{\sum_{k=0}^{K/2}\left|F(k)\right|^2}} \tag{3.5}$$

Brightness and Bandwidth characterize the shape of the spectrum, and roughly indicate the timbre quality of a sound. From this perspective, brightness and bandwidth can serve as good indicators for audio element discrimination, as already shown in many audio classification processes [Scheirer and Slaney 1997][Li 2000][Wold et al. 1996][Fujinaga 1998][Rossignol et al. 1998].

3.2.4 Mel-Frequency Cepstral Coefficient (MFCC)

The set of *Mel-Frequency Cepstral Coefficients* (MFCC) [Rabiner and Juang 1993] is a cepstral representation of the audio signal obtained based on the mel-scaled spectrum. The log spectral amplitudes are first mapped onto the perceptual, logarithmic *mel-scale*, using a triangular band-pass filter bank. Then, the output of the filter bank is transformed into MFCC using the discrete Cosine transform (*DCT*).

$$c_n = \sqrt{\frac{2}{K}}\sum_{k=1}^{K}(\log S_k)\cos[n(k-0.5)\pi / K] \qquad n=1, \ldots, L \tag{3.6}$$

where c_n is the n-th MFCC, K is the number of band-pass filters, S_k is the Mel-scaled spectrum after passing the k-th triangular band-pass filter, and L is the order of the cepstrum.

MFCC is commonly used in speech recognition and speaker recognition systems. However, MFCC also proved to be useful in discriminating between speech and other sound classes, which explains its wide usage in the audio analysis and processing literature [Foote 1997][Moreno and Rifkin 2000][Kimber and Wilcox 1996][Pye 2000]. Based on the suggestions made in literature, we use 8-order MFCC in our approach.

3.2.5 Sub-band Partial Prominence and Harmonicity Prominence

We now consider two further spectral characteristics of audio signals that can be associated with human identification of sounds [Gygi 2001]: i) presence of a prominent

harmonic frequency (i.e. a *partial*) at a certain spectral sub-band, and ii) the harmonicity of the sound. For example, a distinct difference between *cheering* and *laughter* is that *laughter* usually has prominent harmonic partials but *cheering* does not. The features described in previous sections are incapable of describing these characteristics. Brightness and bandwidth can only measure the global energy center and the deviation of the whole spectrum. Although BER and MFCC calculate the average energy in sub-bands, it is still hard to specify whether there are salient components in some sub-bands.

Based on our previous works on audio representation [Cai et al. 2004], we propose two new spectral features, *sub-band partial prominence* (SBPP) and *harmonicity prominence* (HP), to address the abovementioned audio characteristics. The sub-band partial prominence (SBPP)[1] is used to measure whether there are salient frequency components in a sub-band. In other words, the SBPP estimates the existence of prominent partials in sub-bands. It is computed by accumulating the variation between adjacent frequency bins in each sub-band, that is

$$S_p(i) = \frac{1}{H_i - L_i} \sum_{j=L_i}^{H_i-1} \left| \hat{F}(j+1) - \hat{F}(j) \right| \tag{3.7}$$

Here, L_i and H_i are the lower and upper boundaries of the i^{th} sub-band respectively, and the value of $S_p(i)$ indicates the corresponding prominence of salient partial components. The SBPP value for sub-bands containing salient partial components is expected to be large. To reduce the impact induced by the energy variation over time, the original *DFT* spectral coefficient vector $\boldsymbol{F}$ is first converted to the decibel scale and then constrained to the unit L_2-norm, as suggested in [Casey 2001]:

$$\hat{\boldsymbol{F}} = \frac{10\log_{10}(\boldsymbol{F})}{\left\| 10\log_{10}(\boldsymbol{F}) \right\|} \tag{3.8}$$

If we now consider the property of an ideally harmonic sound (with one dominant fundamental frequency f_0), its full spectrum energy is highly concentrated and precisely located at those predicted harmonic positions, which are the multiples of the fundamental frequency f_0. To detect this situation, the following three factors could be

[1] In our previous work [Cai et al. 2004] we referred to this as *sub-band spectral flux*. We rename it here since *sub-band partial prominence* is more suitable and straightforward to represent the meaning of the extracted feature. *Sub-band spectral flux* may be a little confusing in this context, since s*pectral flux*, according to its traditional definition, is usually computed from two neighboring frames.

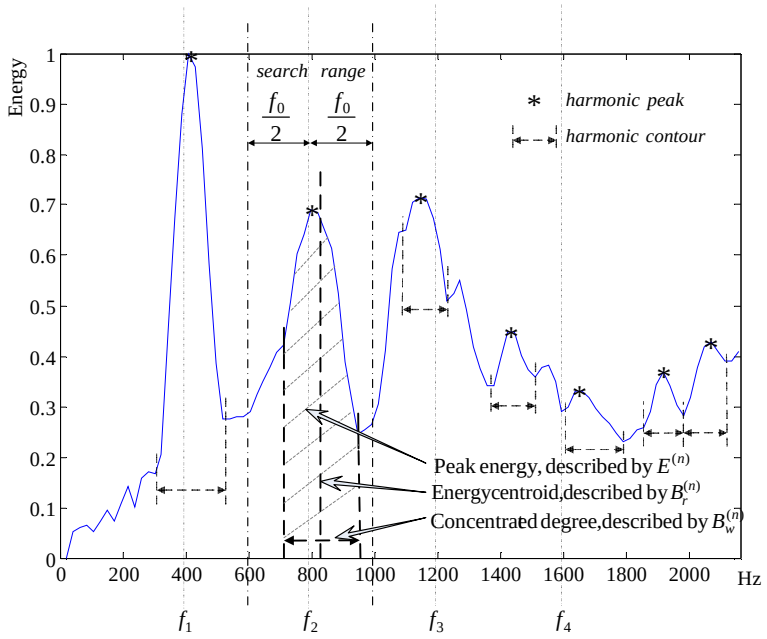


Fig. 3.1. Definition of *harmonicity prominence*. The horizontal axis represents the frequency, and the vertical axis denotes the energy. The harmonic contour is the segment between the adjacent valleys separating the harmonic peaks. Based on the harmonic contour, three factors, that is, the peak energy, energy centroid (brightness) and degree of concentration (bandwidth), are computed to estimate the *harmonicity prominence*, as illustrated at the second harmonic in this example.

measured: i) the energy ratio between the detected harmonics and the whole spectrum, ii) the deviation between the detected harmonics and predicted positions, and iii) the concentration degree of the harmonic energy. The *harmonicity prominence* (HP) is proposed to take into account the above three factors and can be defined as

$$H_p = \frac{\sum_{n=1}^{N} E^{(n)}(1-|B_r^{(n)} - f_n|/0.5 f_0)(1-B_w^{(n)}/B)}{E} \tag{3.9}$$

Here, $E^{(n)}$ is the energy of the detected n^{th} harmonic contour in the range of $[f_n\text{-}f_0/2, f_n\text{+}f_0/2]$ and the denominator E is the total spectral energy. The ratio between $E^{(n)}$ and E stands for the first of the three factors identified above. Further, f_n is the n^{th} predicted harmonic position and is defined as

$$f_n = nf_0\sqrt{1+\beta(n^2-1)} \tag{3.10}$$

where β is the *inharmonicity modification factor* set to 0.0005 following the discussions in [Fletcher and Rossing 1998]. $B_r^{(n)}$ and $B_w^{(n)}$ are the *brightness* and *bandwidth* of the n^{th} harmonic contour, respectively. The brightness $B_r^{(n)}$ is used instead of the detected harmonic peak in order to estimate the frequency center more accurately. The bandwidth $B_w^{(n)}$ describes the concentration degree of the n^{th} harmonic. It is normalized by the constant B, which is defined as the bandwidth of an instance where the energy is uniformly distributed in the search range. Thus, the components $(1-|B_r^{(n)}-f_n|/0.5f_0)$ and $(1-B_w^{(n)}/B)$ in the numerator of (3.9) represent the second and the third factor defined above, respectively. An illustration of the definition of *harmonicity prominence* is given in Fig. 3.1.

In our implementation, f_0 is estimated by an autocorrelation-based approach. Only the first 4 harmonic partials are considered in the computation, since only these harmonic partials are sufficiently prominent in most cases. Furthermore, in the case where the fundamental frequency cannot be precisely predicted, f_0 is varied in a pre-defined range first, and then the corresponding HP values are calculated, the maximum of which is chosen as the HP value for the frame. For a sound without pitch, H_p is set to zero.

3.3 Window-level Features

While the features from the previous section are extracted from one audio frame, in this section we choose to group audio frames into longer temporal audio segments of the length *t*, extract features at the segment level, and use these longer segments as the basis for the subsequent audio processing steps. This step will not only reduce the computational complexity in subsequent steps of content-based audio analysis, but also result in additional useful features, other than those extracted at the frame level. For example, as pointed out above, the variation (e.g. standard deviation) of ZCR and STE is more discriminative if measured over a longer audio interval than per frame. For this purpose, in our approach, a sliding window of 1.0 second with 0.5 seconds overlap is applied to the frame sequence. Future reference to an *audio segment* will relate to this one-second-long segment that will serve as the basic unit in further audio processing steps described in this thesis. The window and step length are selected to balance the detection resolution and the computational complexity. At each window position, the mean and standard deviation of the frame-level features are computed and used to represent the corresponding audio segment.

Table 3.4 The list of window-based features used in our approach

Feature Kind	Feature list
basic statistics derived	mean and standard deviation of the frame-based features
window-level features	HZCRR, LSTER, spectrum flux, noise frame ratio

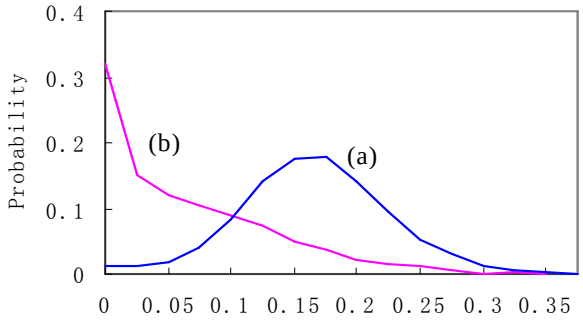


Fig. 3.2. An illustration of a distribution of HZCRR values: (a) speech, and (b) music.

In addition to computing the mean and standard deviation of the frame-level features, specific window-level features can be considered as well, such as those listed in Table 3.4, which have already shown their effectiveness in various audio analysis approaches. In the following sections we describe the models and computation of these features in more detail.

3.3.1 High ZCR Ratio

As mentioned above, the variation of ZCR is more discriminative than the exact value of ZCR. Although this variation is frequently modeled using the standard deviation of ZCR, the high zero-crossing rate ratio (HZCRR) can also be used for this purpose.

HZCRR is defined as the fraction of frames in the analysis window, whose ZCR are at least 50% higher than the average *ZCR* computed in the window, that is

$$HZCRR = \frac{1}{2N}\sum_{n=0}^{N-1}[\mathrm{sgn}(ZCR(n) - 1.5 avZCR) + 1] \tag{3.11}$$

Here, n is the frame index, $ZCR(n)$ is the zero-crossing rate at the n-th frame, N is the total number of frames, $avZCR$ is the average ZCR in the analysis window, and sgn[] is a sign function. Using similar reasoning as in Section 3.2.1, the value of HZCRR is expected to be higher in speech signals than in music.

Fig. 3.2 shows the distributions of HZCRR values computed for a large number of speech and music signals. It can be seen that the center of HZCRR distribution of speech segment is around 0.15, while HZCRR values of music segments mostly fall below 0.1, though there are significant overlaps between these two curves. If we use the cross-point of two displayed HZCRR curves as the threshold to discriminate speech from music, the expected classification error would be 19.36%.

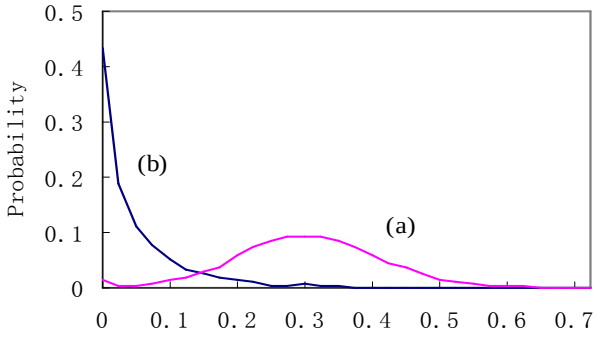


Fig. 3.3. An illustration of a distribution of LSTER values: (a) speech, and (b) music

3.3.2 Low Short-time Energy Ratio

As an analogy for selecting HZCRR to model the variations of the ZCR within the analysis window, the low short-time energy ratio (LSTER) can be defined to model the variation of the STE in this window, as proposed in [Scheirer and Slaney 1997]. LSTER is the fraction of the frames within the analysis window, whose STE values are less than a half of the average STE in the window, that is,

$$LSTER = \frac{1}{2N}\sum_{n=0}^{N-1}[\mathrm{sgn}(0.5avSTE - STE(n)) + 1] \tag{3.12}$$

where *N* is the total number of frames in the analysis window, STE(*n*) is the short time energy at the *n*-th frame, and *avSTE* is the average STE in the window.

Similar to HZCRR, the LSTER measure of speech is expected to be much higher than that of music. This can be seen clearly from the distributions of LSTER values obtained for a large number of speech and music signals, as illustrated in the Fig. 3.3. It is shown that LSTER value of speech is around 0.15 to 0.5, while that of music is mostly less than 0.15. Based on Fig. 3.3, if we use the cross-point of two displayed LSTER curves as a threshold to discriminate between speech and music, the expected error rate would be only 8.27%.

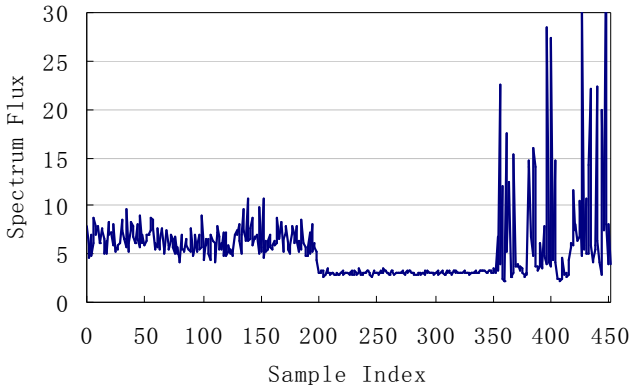


Fig. 3.4. The spectrum flux curve of speech (0-200 seconds), music (201-350 seconds) and environmental sound (351-450 seconds)

3.3.3 Spectrum Flux

Spectrum Flux (SF) is defined as the average variation of the spectrum between adjacent two frames in the analysis window, that is,

$$SF = \frac{1}{(N-1)(K-1)} \sum_{n=1}^{N-1} \sum_{k=1}^{K-1} [\log(A(n,k)+\delta) - \log(A(n-1,k)+\delta)]^2 \qquad (3.13)$$

where *A(n, k)* is the absolute value of the *k*-th *DFT* coefficient of the *n*-th frame, *K* is the order of DFT, δ is a very small value to avoid computation overflow, and *N* is the total frame number in the analysis window.

Similar to HZCRR and LSTER, the SF of speech is expected to be larger than that of music. We also found that the spectrum flux of environmental sounds (or audio effects) is generally very high and changes more dynamically than for speech and music. To illustrate this, Fig. 3.4 shows the SF computed for an audio segment consisting of speech (0 to 200 seconds), music (201 to 350 seconds) and environmental sounds (351 to 450 seconds). From this figure, it can be seen that the SF is a promising feature to discriminate between audio elements including speech, audio effects, and music.

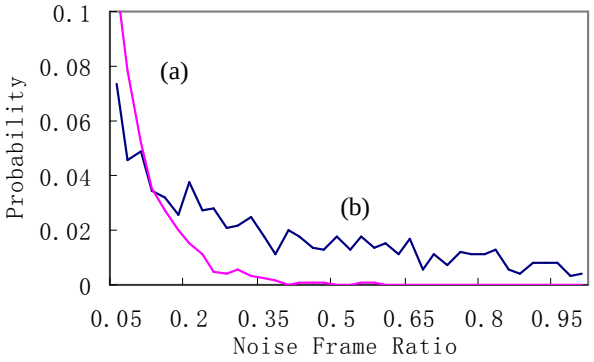


Fig. 3.5. An illustration of a distribution of NFR values; (a) music and (b) environment sound.

3.3.4 Noise Frame Ratio

*Noise frame ratio (*NFR) is defined as the ratio of noise frames in a given audio clip. A frame is considered as a noise frame if the maximum local peak of its normalized correlation function is lower than a pre-set threshold. The NFR is usually used to discriminate environmental sounds from music and speech, and to detect noisy sounds. For example, the NFR value of a noise-like environmental sound is higher than that for music, because it contains many more noise frames. As can be observed in Fig. 3.5, considering NFR values can be helpful in separating these two classes of audio.

3.4 Feature Vector Generation

Following up on the discussion from Section 3.1, the broad set of features that we select based on the previous work in the field of content-based audio analysis needs to be tuned for a given use case to form a suitable feature vector serving as input into further audio analysis steps. We briefly explain in this section how we proceeded with creating such case-optimal feature vectors.

Since the values and dynamics of various extracted features may vary considerably over the feature set, simply concatenating them all into a long feature vector is not likely to lead to good results. Therefore, a *normalization* process needs to be performed on the features first to equalize their scales. The normalization (also called standardization or z-scores) is typically performed using the mean and standard deviation per feature, as

$$x_i^{'} = (x_i - \mu_i) / \sigma_i \tag{3.14}$$

where x_i is the *i*-th feature, and where the corresponding mean μ_i and standard deviation σ_i can be obtained from the analyzed data set.

In addition to normalization, we follow a standard approach and employ the *principle component analysis* (*PCA*) to improve the effectiveness of the feature vector while minimizing its dimension. Technically, PCA is an orthogonal linear transformation that transforms the data to a new coordinate system, to reveal the main characteristics (*principal components*) of the data that contribute most to the variance in data, and therefore best explain the data. To perform PCA, we apply *singular value decomposition* (*SVD*) [Wall et al. 2003] to the $M \times N$ matrix X' containing N-dimensional normalized feature vectors collected from M segments (usually $M >> N$). Each row corresponds to a feature vector of one audio segment. By applying the SVD, the matrix X' can be written as

$$X' = USV^T \tag{3.15}$$

In terms of SVD, V and U are, respectively, an $N \times N$ and $M \times N$ matrix containing the right and left singular vectors, while the diagonal $N \times N$ matrix S = diag$\{\lambda_1, \ldots, \lambda_N,\}$ contains singular values, with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. In terms of PCA, singular vectors (columns) of the matrix V can be seen as principal components of X', each of which has its corresponding singular value. The larger the singular value is, the more principal (or more important) the component is. Assuming that V_m is a matrix keeping the first m

principal components (by keeping the first *m* columns from *V*), the original feature set *X'* can be replaced by a reduced, *PCA*-transformed feature set

$$X'' = X'V_m \tag{3.16}$$

which only preserves those features that are relevant to subsequent audio signal analysis, while leaving out the redundant and irrelevant (noisy) features. In our approach, the number *m* of principal components is determined using the following equation:

$$m = \arg\min_k \{\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n} \lambda_i > \eta\} \tag{3.17}$$

Based on initial experiments, we set the threshold η to 0.9, which means that 90% principle components are kept. After normalization and PCA, the resulting feature set is used to form the feature vector serving as input into the subsequent audio content discovery steps, as presented in the next chapters.

Chapter 4

Audio Element Discovery and Key Audio Element Spotting

By building on the feature-based audio representation discussed in the previous chapter, we now aim at developing a methodology for extracting mid-level semantic descriptors from audio signals in the form of (key) audio elements. Referring to the discussion from Chapter 1, deployment of audio elements in content-based audio analysis divides the process of semantic inference into two steps: 1) audio elements discovery from features, and 2) semantic inference from audio elements. Our recent studies [Lu et al. 2005] have shown that the approaches to semantic parsing and classification of audio based on (key) audio elements outperform the "plain" feature-based approaches. This can be observed in particular on the increased precision in the obtained results. For instance, as will be shown on the case of audio scene segmentation in Chapter 5, audio-element based analysis inherently searches for high-level content breaks only, and neglects irrelevant variations in audio data due to which the feature-based approaches usually produce an over-segmentation.

The scheme we propose for automatic audio element discovery builds on an iterative spectral clustering method. Using this method, we group audio segments (as defined in Chapter 3) with similar signal properties into clusters, and these obtained clusters are adopted as audio elements. To detect key audio elements from the obtained clusters, two cases are considered. In the first case, we assume that only one audio document is

---

This chapter is based on the following publications (also to be found in the list of references):

- Lu, L., and Hanjalic, A. "Towards Optimal Audio Keywords Detection for Audio Content Analysis and Discovery, " *Proc. 14th ACM Int'l Conf. on Multimedia*, 825-834, 2006
- Lu, L., and Hanjalic, A. "Audio Keywords Discovery for Text-Like Audio Content Analysis and Retrieval," *IEEE Trans. on Multimedia*, vol. 10, no. 1, 74-85, 2008

available for analysis. Then, a number of heuristic importance indicators are defined and deployed to select the key audio elements. In the second case, multiple audio documents are available. There, inspired by the effectiveness of the concepts of TF and IDF from text document analysis, and some similar measures used in video content analysis [Uchihashi et al. 1999], we see the possibility to apply these measures (or their equivalents) to audio documents to help the key audio elements detection in terms of robustness and level of automation. In particular, *expected term frequency* (*ETF*) and *expected inverse document frequency* (*EIDF*), which are equivalents to TF and IDF, respectively, are proposed. In addition, *expected term duration* (*ETD*) and *expected inverse document duration* (*EIDD*) are computed as well, which take into account the discriminative power of the overall duration of a particular audio element in an audio document in characterizing the semantics of that document.

4.1 Audio Element Discovery

Audio elements to be found in complex composite audio documents, such as sound tracks of movies, usually show complicated and irregular distributions in the feature space. However, traditional clustering algorithms such as K-means, are based on the assumption that the cluster distributions in the feature space are Gaussians [Duda 2000], which is usually not satisfied in complex cases. Furthermore, the clustering results are usually affected by the initially selected centroids so that multiple restarts are needed to obtain the optimal results. As a promising alternative, spectral clustering [Ng et al. 2001] showed its effectiveness in a variety of complex applications, such as image segmentation [Yu and Shi 2003][Zelnik-Manor and Perona 2004] and the multimedia signal clustering [Ngo et al. 2001][Radhakrishnan et al. 2004]. We therefore choose to employ spectral clustering to decompose audio documents into audio elements. To further improve the robustness of the clustering process, we adopt the self-tuning strategy [Zelnik-Manor and Perona 2004] to set context-based scaling factors for different data densities, and build an iterative scheme to perform a hierarchical clustering of input data.

4.1.1 Spectral Clustering

Spectral clustering can be seen as an optimization problem of grouping together similar data samples based on eigenvectors of a (possibly normalized) affinity matrix that contains the similarity values measured between each pair of data samples. Like other

clustering methods, its basic idea is to keep together the data with similar features while separating the data with different features. For this purpose, and using the concepts from the graph theory, a complete undirected graph *G(V,E)* is first constructed from a data set. Here, *V* is the node set where each node represents one data sample, and *E* is the edge set. The weight of edge e_{uv} defines the similarity between data *u* and data *v*. This weight can be defined as

$$w(u,v) = e^{-d(u,v)/2\sigma^2} \tag{4.1}$$

where *d(u,v)* is a distance measure between data *u* and *v*, and σ is a scaling factor. *W*=[*w*] forms an affinity matrix for the graph *G*.

Grouping the data into *N* clusters is now identical to partitioning the graph *G(V,E)* into *N* disjoint sets, simply by removing the edges connecting the sets. Taking the case *N*=2 as an example, the degree of dissimilarity between two sets, *A* and *B*, can be computed as the total weight of the edges that have been removed, that is,

$$cut(A,B) = \sum_{u\in A, v\in B} w(u,v) \tag{4.2}$$

The notation "*cut*" in (4.2) is adopted from the graph theory. From this perspective, obtaining the optimal partitioning of a graph could be seen as an optimization problem of minimizing the *cut* value. However, as shown in [Wu and Leahy 1993], this *minimum cut* approach tends to be biased towards cutting out unnaturally small sets of isolated graph nodes. To avoid this, [Shi and Malik 1997] proposed an alternative dissimilarity measure, referred to as *normalized cut* and defined as

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \tag{4.3}$$

Here, *assoc(A,V)* and *assoc(B,V)* are the total weights between each of the nodes in *A* or B, respectively, and all nodes in the graph.

[Shi and Malik 1997] indicated that normalized cut can minimize the disassociation between the groups (inter-class distance) and simultaneously maximize the association within the groups (intra-class distance). This joint optimization of inter- and intra-class distance can be obtained by solving the generalized eigenvector system,

$$(D-W)\boldsymbol{y} = \lambda D\boldsymbol{y} \tag{4.4}$$

where $\boldsymbol{y}$ are the generalized eigenvectors, and *D* is a diagonal matrix with $D(i,i) = \sum_j W(i,j)$. This generalized eigenvector system can further be transformed into

a standard eigensystem as

$$D^{-1/2}(D-W)D^{-1/2}\boldsymbol{z} = \lambda \boldsymbol{z} \tag{4.5}$$

with $\boldsymbol{z} = D^{1/2}\boldsymbol{y}$.

The obtained eigenvectors can be used to do the partition. Each eigenvector indicates a segmentation possibility based on the value of its components, and multiple (k) eigenvectors stands for multiple (k) splits. While [Shi and Malik 1997] worked on 2-cluster partition, [Ng et al. 2001] further proposed a spectral clustering method to use k eigenvectors simultaneously to partition the data into k clusters, following related ideas from [Shi and Malik 1997] and [Weiss 1999]. Spectral clustering has been successfully applied to a number of applications, and we therefore considered this algorithm as the basis of our clustering approach.

We first assume that for a given audio document a set $U = \{u_1, \ldots, u_n\}$ of feature vectors is obtained through the feature extraction process described in Chapter 3. There, each sample u_i represents the feature vector of one audio segment, and n is the total number of audio segments in the audio document being analyzed. After specifying the search range $[k_{min}, k_{max}]$ for the most likely number of audio elements existing in the document, the spectral clustering algorithm can be carried out as the following series of steps:

Algorithm: Spectral_Clustering (U, k_{min}, k_{max})

1. Form an affinity matrix A defined by $A_{ij} = \exp(-d(u_i, u_j)^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$. Here, $d(u_i, u_j) = \|u_i - u_j\|$ is the Euclidean distance between the feature vectors u_i and u_j, and σ is the scaling factor. The selection of σ will be discussed in the next Section.

2. Obtain the diagonal matrix D, whose (i, i) element is the sum of A's i-th row, and construct the normalized affinity matrix $L = D^{-1/2}AD^{-1/2}$.

3. If $(x_1, \ldots, x_{k_{max}+1})$ are the $k_{max}+1$ largest eigenvectors of L, and $(\lambda_1, \ldots, \lambda_{k_{max}+1})$ are the corresponding eigenvalues, then the optimal cluster number k is estimated based on the eigen-gaps between adjacent eigenvalues as:

$$k = \arg\max_{i \in [k_{\min}, k_{\max}]}(1 - \lambda_{i+1} / \lambda_i) \tag{4.6}$$

   Then, form the matrix $X = [x_1 x_2 \ldots x_k] \in \boldsymbol{R}^{n \times k}$ by stacking the first k eigenvectors in columns.

4. Form the matrix Y by renormalizing each of X's rows to obtain unit length, that is:

$$Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2} \tag{4.7}$$

5. Treat the rows of Y as points in $\boldsymbol{R}^k$, and cluster them into k clusters via the cosine-distance based K-means algorithm. The initial centers in the K-means are selected to be as orthogonal to each other as possible [Yu and Shi 2003].

6. Assign the original data point u_i to cluster c_j if and only if the row i of the matrix Y is assigned to c_j.

4.1.2 Context-based Scaling Factors

Although reasonable results can be obtained based on the algorithm described above, the clustering performance is likely to improve if the scaling factor selection is considered more carefully. In the spectral clustering algorithm, the scaling factor σ affects how rapidly the similarity measure A_{ij} decreases when the Euclidean distance $d(u_i, u_j)$ increases. In this way, it actually controls the value of A_{ij} at which two audio segments are considered similar. In the algorithm from [Ng et al. 2001], σ is set uniformly for all data points (for example, σ is set to the average Euclidean distance in the data), based on the assumption that each cluster in the input data has a similar distribution density in the feature space. However, such assumption is usually not satisfied in composite audio data, which often contain clusters with different cluster densities. Suppose there are two clusters, a dense and a sparse one, and the data of the sparse cluster is sparsely distributed around the dense cluster, the algorithm tends to either merge these two clusters into one, or split the cluster with sparse density into many smaller clusters.

Fig. 4.1(a) illustrates an example affinity matrix of a 30-second audio clip composed of *music* (0-10s), *music with applause* (10-20s), and *speech* (20-30s), using a uniform scaling factor. From the figure, it can be noticed that the density of *speech* is sparser than the densities of other elements, while *music* and *music with applause* are close to each other and hard to separate. Thus, the "standard" spectral clustering cannot properly estimate the number of clusters using (4.6) and based on the eigenvalues and eigen-gaps shown at the bottom of Fig. 4.1(a). Actually, from the Fig. 4.1(a), the estimated number of clusters would be one.

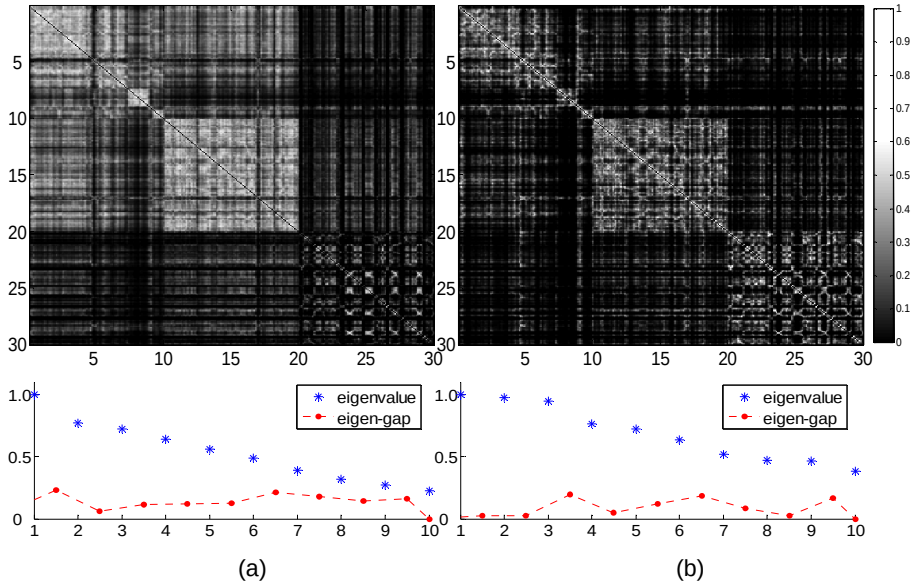


Fig. 4.1. The affinity matrices with top 10 eigenvalues and the eigengaps, computed for a 30-second audio clip consisting of *music* (0-10s), *music with applause* (10-20s), and *speech* (20-30s): (a) using a uniform scaling factor, (b) using the context-based scaling factors.

To obtain a more reliable similarity measure and improve the clustering robustness, the scaling factor needs to be set in a more sophisticated (adaptive) way. An intuitive idea is that, if a cluster has a sparse density, the scaling factor should be large. Otherwise, the scaling factor could be set to a smaller value. According to this idea, in our approach, the self-tuning strategy [Zelnik-Manor and Perona 2004] is employed to select context-based scaling factors. That is, for each data point u_i, the scaling factor is set adaptively based on its context data density as:

$$\sigma_i = \sum_{u_j \in close(u_i)} d(u_i, u_j) / n_b \tag{4.8}$$

where $close(u_i)$ denotes the set containing n_b nearest neighbors of u_i. In our approach we experimentally set n_b to 5. Accordingly, the affinity matrix can now be re-defined as:

$$A_{ij} = \exp(-d(u_i, u_j)^2 / (2\sigma_i \sigma_j)) \tag{4.9}$$

Fig. 4.1(b) shows the corresponding affinity matrix computed using the context-based scaling factors. It can be noticed that the three blocks on the diagonal are more distinct than those in Fig. 4.1(a). In Fig. 4.1(b), the *speech* segment appears more concentrated in the affinity matrix, while better separation is achieved between *music* and *music with applause*. It can also be noted that the prominent eigen-gap between the 3rd and 4th eigenvalue predicts the correct number of clusters.

4.1.3 Iterative Clustering

Another thing we need to consider is the purity of the obtained audio elements. In other words, we need to prevent audio segments belonging to different audio elements to be grouped into the same cluster. Impure audio elements are insufficiently representative (discriminative) with respect to the semantic content, and can be considered bad input into the semantic inference processes.

In view of the above, we propose an iterative clustering scheme to verify whether a cluster can be partitioned any further. That is, at each iteration, every cluster obtained from the previous iteration is submitted again to the spectral clustering scheme. Although a cluster is inseparable in the (large scale) affinity matrix in the previous iteration, it may become separable in a new affinity matrix (small scale, only considering the cluster's own data) during the next iteration. A cluster is considered inseparable if spectral clustering returns only one cluster. The iterative scheme can be described by the following pseudo code.

Iterative_Clustering(U, k_{min}, k_{max})
{
    [k, {c_1, …, c_k}] = Spectral_Clustering(U, k_{min}, k_{max});
    if (k is equal to 1) return;
    for (j = 1; $j \leq k$; j++)
        Iterative_Clustering(c_j, 1, k_{max});
}

It is important to note that iterative clustering may introduce over-segmentation, that is, one actual audio element can be spread over several clusters, each of which is then adopted as a different audio element. As this is typical for audio elements that appear with small variations at various time instances of an audio document, such

over-segmentation could generally be considered an analogy to distinguishing between the variations in text words, e.g., caused by different endings.

Another type of over-segmentation may also be caused, however, due to the fact that we apply spectral clustering to a large audio document collection. In our approach, audio elements are extracted independently from different audio documents by applying iterative spectral clustering to each available audio document separately. Since some parts of different audio documents may have similar audio properties, it is expected that many audio elements obtained for different documents are actually belonging together into one and the same cluster. Intuitively, one could choose to combine all available audio documents together first, and then apply spectral clustering to the entire collection. However, combining audio tracks would make the affinity matrix too large and the SVD applied to this matrix computationally unaffordable. Ideas on how to deal with this type of over-segmentation will be introduced and explained in different contexts in later sections of this thesis.

4.1.4 Smoothing

The clustering process groups the audio segments together into clusters based on their feature similarity, but it does not take into account temporal sequencing of the segments. In order to avoid unrealistic discontinuities in the cluster assignment between consecutive audio segments, an extra smoothing step involving a median filter is performed after the clustering process. For example, if consecutive audio segments are assigned to clusters A and B as "A-A-B-A-A", this series of segments will be smoothed to "A-A-A-A-A" to remove unlikely discontinuities in the semantic content flow.

4.1.5 Terminology

Fig. 4.2 shows an example of an audio elements sequence after smoothing, where an audio clip is decomposed into 3 audio elements, *e1*, *e2* and *e3*, as indicated by different gray-level values. Based on Fig 4.2, we now introduce the terminology that will be used in the subsequent sections. Each audio element has several *occurrences* along the data stream. Each occurrence of an audio element is actually a smoothed series of continuous audio segments that belong to the corresponding audio element cluster. For example, the blocks marked with 1-5 are five occurrences of audio element *e1*. Correspondingly, we refer to the duration of an audio element occurrence as the *length* of that occurrence. For example, l_3 is the length of the 3rd occurrence of audio element

e1. We also refer to the sum of the lengths of all occurrences of an audio element as the (overall) *duration* of this audio element. Moreover, since we already realized that an over-segmentation may result in a number of semantically related audio elements, we refer to these audio elements jointly as an *audio term*.

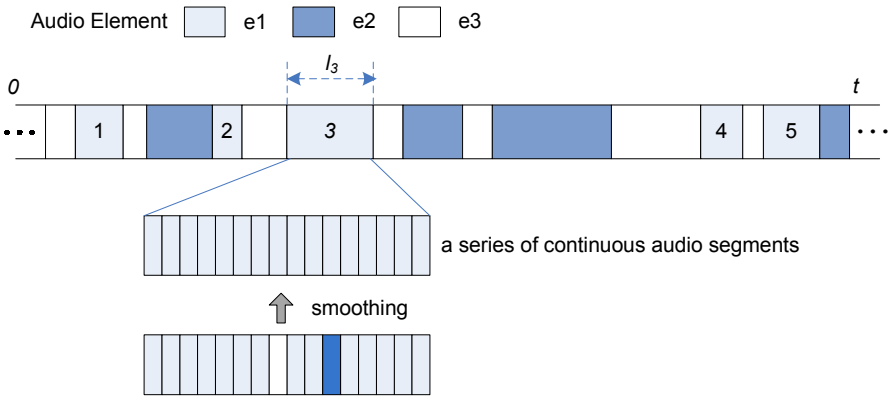


Fig. 4.2. An illustration of an audio element sequence. All blocks indicated by the same grey value represent the same audio element. The larger numbered blocks corresponding to the element *e1* represent different occurrences of that audio element. An occurrence consists of a smoothed series of audio segments belonging to the same audio element cluster.

4.2 Key Audio Element Spotting: Single Document Case

To this end, we have discovered audio elements in audio documents, which we consider being analog to the words in text documents. In the next step, we aim at spotting those audio elements – the key audio elements - that are most representative for the behavior of an audio data stream at various time instances. Such audio elements would play similar role as the keywords in text, and could help us further perform content-based analysis and retrieval of audio documents using the proven theories and methods of text document analysis. Just like the words in text, different audio elements may have different importance in content-based audio analysis. For example, while an award ceremony typically contains the sounds like *speech*, *music*, *applause*, *cheering* and their different combinations, *applause* and *cheering* can be considered good representatives of the actual content of the ceremony.

To spot key audio elements, we can draw an analogy to keyword extraction in text document analysis, where the most commonly used criteria are TF and IDF. However, in content-based audio analysis, we may only have a single audio document available (the one to be analyzed) which prevent us from estimating the IDF values as normally done in text analysis based on a large training corpus. In this section, we propose some heuristic importance indicators for audio elements based on the analysis of a single audio document. Then, in the next section we assume that multiple audio documents are available, and present an audio element weighting scheme that follows the analogy to the TFIDF concept more closely than in the single document case.

As a first heuristic importance indicator, we consider the *occurrence frequency* of an audio element, which is a direct analogy to TF in text analysis. However, just like in the text analysis, it is not necessarily the case that the higher occurrence frequency of an audio element implies its higher importance. This can be drawn from the following analysis. For example, the major part of the sound track of a typical action movie segment consists of "usual" audio elements, such as *speech*, *music*, speech mixed with music, etc., while the remaining smaller part includes audio elements that are typical for action, like *gun-shots* or *explosions*. As the usual audio elements can be found in any other (e.g. romantic) movie segment as well, it is clear that only this small set of specific audio elements is the most important to characterize the content of a particular movie segment. To compensate for this in text analysis, another measure, IDF, emphasizing relative uniqueness of a word in one document compared to other documents, is usually combined with TF to obtain a reliable weight for each word. However, in the case of single document, IDF cannot be calculated. To compensate for this, we apply a heuristic constraint to the occurrence frequency in the form of a naive (normalized) Gaussian model to compute the *Element Frequency* indicator *efrq*, that is,

$$efrq(e_i, D) = \exp(-(n_i - \alpha \cdot n_{avg})^2 / (2n_{std}^2)) \tag{4.10}$$

Here, e_i is an audio element in audio document D, n_i is the number of occurrences of e_i, and n_{avg} and n_{std} are the corresponding mean and standard deviation of the numbers of occurrences of all audio elements. The factor α adjusts the expectation of how often the key elements will likely occur. Using this indicator, the audio elements that appear far more or far less frequently than the expectation $\alpha \cdot n_{avg}$ are punished. In terms of TF and IDF, it can be said that the *efrq* indicator combines both in one measure.

An important difference to the text case is in the fact that an audio element has duration information attached to each of its occurrences. In order to detect key audio elements accurately and robustly, we apply a similar reasoning as above to extend the "importance" measure by other two relevant indicators, the total duration and the

average occurrence length of an audio element, which are typically very different for various sounds in an audio document. Background sounds are usually majorities while key audio elements are minorities. For instance, in a situation comedy, both the total duration and the average length of the *speech* are considerably longer than that of the *laughter* or *applause*. Based on the above observations, another two heuristic importance indicators are designed to capture the observations made regarding the element duration. These indicators are defined as follows:

Element Duration takes into account the total duration of audio element e_i in the document:

$$edur(e_i, D) = \exp(-(d_i - \beta \cdot d_{avg})^2 / (2d_{std}^2)) \tag{4.11}$$

where d_i is the total duration of e_i, and d_{avg} and d_{std} are the corresponding mean and standard deviation. The factor β adjusts the expectation of the duration of key audio elements, and has a similar effect as α.

Average Element Length takes into account the average length of e_i over all its occurrences, as:

$$elen(e_i, D) = \exp(-(l_i - \gamma \cdot l_{avg})^2 / (2l_{std}^2)) \tag{4.12}$$

where l_i is the average occurrence length of e_i, and l_{avg} and l_{std} are the corresponding mean and standard deviation. The factor γ is similar to α and β and adjusts the expectation of the average occurrence length of key audio elements.

The heuristic importance indicators defined above can be tuned adaptively for different applications, based on the available domain knowledge. For example, to detect unusual sounds in surveillance videos, factors α, β, and γ could be set relatively small, if such sounds are not expected to occur frequently and are of a relatively short duration.

Based on these importance indicators and by realistically assuming that the above indicators are independent of each other, we measure the importance (or weight) of each audio element as,

$$W(e_i, D) = efrq(e_i, D) \cdot edur(e_i, D) \cdot elen(e_i, D) \tag{4.13}$$

To better explain the underlying idea of the product in (4.13), the weight $W(e_i, D)$ can be seen as an analogy to the posterior probability that e_i is a key audio element, given the observations regarding the n_i, d_i and l_i. Further, each of the equations (4-10)-(4.12) can be seen as an analogy to the likelihood for each observation type (n_i,

d_i or l_i), given the key audio element hypothesis at e_i. If the three observations are considered independent, then the posterior probability that e_i is a key audio element is proportional to the product of the likelihoods.

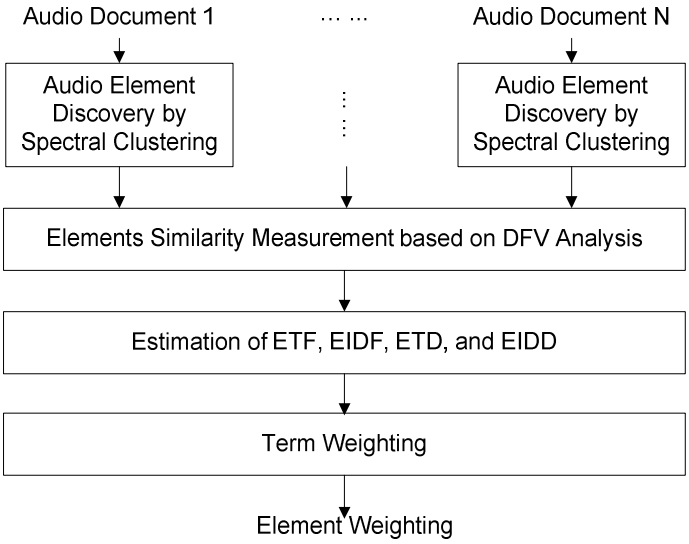


Fig. 4.3 The flowchart of the TFIDF-based audio elements weighting

4.3 Key Audio Element Spotting: Multiple Document Case

Inspired by the effectiveness of term frequency and inverse document frequency in text analysis, we see the possibility to apply these measures (or their equivalents) to improve key audio elements detection when multiple audio documents are available. To do this, the number of occurrences of a particular audio *term* in one document needs to be computed to estimate its TF value, while for computing its IDF value simultaneous analysis of multiple audio documents needs to be performed. Here, recalling the discussion on over-segmentation in Section 4.1.3 and the definition of audio term in Section 4.1.5, we first need to estimate the probability that an audio element belongs to a particular audio term.

Fig 4.3 illustrates our TFIDF-based audio element weighting scheme. In this approach, the similarity between audio elements found in multiple audio documents is first computed based on *dominant feature vectors* (*DFV*), and then the similarity values

are used to compute the probability of the occurrence of one audio term in one and across multiple documents. Evaluating the *DFV*-based audio elements similarity can be considered an equivalent to identifying the matches between words in text that are semantically the same but, for instance, have different endings

The obtained probability is further used to compute the equivalents of the standard *TF* and *IDF* measures, namely, the *expected term frequency* (*ETF*) and the *expected inverse document frequency* (*EIDF*). In addition, the *expected term duration* (*ETD*) and *expected inverse document duration* (*EIDD*) are computed as well, which again take into account the discriminative power of the duration of a particular audio element in characterizing the semantics of an audio document. Finally, the four measures are combined to give the final importance weight of an audio term, which is then assigned to all audio elements corresponding to this term.

4.3.1 Evaluating Similarity of Audio Elements

To take into account possible high-level variations of one and the same audio term, and judge which audio elements correspond to the same audio term, we introduce a procedure for measuring the similarity $S(e_i,e_j)$ between audio elements e_i and e_j, which will be further used to get a reliable indication of audio term occurrence. To measure this similarity, a possible approach would be to represent each audio element using a standard method involving a Gaussian mixture model (*GMM*). However, as no assumptions about covariance matrices of *GMMs* can be made for a general case, computing the distance between *GMMs* is not likely to be easy. Besides, compared to the similarity computation between audio segments in the spectral clustering step, searching for similarity between audio elements needs to be done with respect to high-level signal descriptors, which will eliminate the influence of irrelevant (low-level) signal variations. We therefore choose for an alternative approach that employs *Dominant Feature Vectors (DFVs).*

4.3.1.1 Dominant Feature Vectors

Each audio element usually stands for a number of audio segments and thus a number of feature vectors, which typically have complex distributions and multiple salient characteristics. To represent the salient characteristics of an audio element we employ *DFVs*, which are the principle components in the feature space. Following the same general procedure that we already defined for feature selection in Chapter 3, the *DFVs*

are also computed via the singular value decomposition (*SVD*) on the feature space of an audio element, or in other words, on the $N \times M$ matrix X containing in each of its columns the N-dimensional feature vector of one of M audio segments belonging to the audio element considered (usually $M >> N$). Using *SVD*, the decomposition of X can be written as

$$X = USV^T \tag{4.14}$$

where in this case $U=\{u_1, \ldots, u_N\}$ is an $N \times N$ orthogonal matrix containing the spectral principle components, $S = \text{diag}\{\lambda_1, \ldots, \lambda_N\}$ is an $N \times N$ diagonal matrix of singular values with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, and V is an $N \times M$ matrix containing the temporal principle components. Those spectral principal components in U associated with large singular values represent the primary distributions of the audio element in the feature space, and can therefore be adopted as *DFVs*. The required number m of *DFVs* describing an audio element is related to the amount of feature variation, and in our approach, it is chosen using the similar expression (3.17) with the threshold η set again to 0.9.

It should be noted that our approach to *DFV* extraction is different from traditional *PCA* applications. While *PCA* is traditionally used to remove the noisy feature dimensions, our method removes the noisy feature vectors, but preserves the dimension of each feature vector. Moreover, dominant feature vectors are extracted to form a *signal subspace*, which represents the most salient characteristics of an audio element. In contrast to this, *PCA* usually maps feature vectors into the principle *feature subspace*.

4.3.1.2 Definition of Audio Element Similarity

We now assume to have two audio elements e_1 and e_2, which contain m_1 and m_2 *DFVs* respectively. We denote their *i-th* and *j-th DFV* as $q_{e1,i}$ and $q_{e2,j}$, and the corresponding singular values as $\lambda_{e1,i}$ and $\lambda_{e2,j}$, respectively. To measure the similarity between e_1 and e_2, we first consider the similarity between each pair of their *DFVs*, $q_{e1,i}$ and $q_{e2,j}$, which is usually defined as their inner-product, that is $s_{i,j} = \| q_{e1,i}^T q_{e2,j} \|$.

Since different *DFVs* have different importance, which is determined by their corresponding singular values, they should contribute differently to the audio element similarity measure. In order to take this into account, we define the similarity between two audio elements as the weighted sum of the similarity between every pair of their *DFVs*, that is

$$S = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} w_{i,j} s_{i,j} \tag{4.15}$$

where the weight $w_{i,j}$ is determined based on the corresponding singular values, as

$$w_{i,j} = \lambda_{e1,i} \lambda_{e2,j} / \sqrt{\sum_{i=1}^{m_1} \lambda_{e1,i}^2 \sum_{j=1}^{m_2} \lambda_{e2,j}^2} \tag{4.16}$$

The weight is selected as such for the following two reasons:

1. it needs to be proportional to the contributions of the corresponding *DFVs*, which are related to the singular values, $\lambda_{e1,i}$ and $\lambda_{e2,j}$;
2. the weighted sum should be equal to one, when two audio elements are the same.

Based on the above, the similarity between two audio elements can now be defined as:

$$S_{dfv}(e_1, e_2) = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \lambda_{e1,i} \lambda_{e2,j} \| q_{e1,i}^T \cdot q_{e2,j} \| / \sqrt{\sum_{i=1}^{m_1} \lambda_{e1,i}^2 \sum_{j=1}^{m_2} \lambda_{e2,j}^2} \tag{4.17}$$

This similarity is symmetric as $S_{dfv}(e_1, e_2) = S_{dfv}(e_2, e_1)$, and its value is in the range of [0, 1]. When the subspaces of e_1 and e_2 are aligned, their similarity is 1, and when they are orthogonal to each other, the value is 0.

4.3.2 Audio Element Weighting Scheme

To estimate TF and IDF of a given audio term, we have to check all reoccurrences of an audio term. We do this by searching for audio elements that are sufficiently similar to each other in terms of (4.17) and that can therefore be said to correspond to one and the same audio term. Due to the missing exact match between audio elements, we can only speak about the probability for reoccurrence of the term, where this probability depends on the value of the similarity measure (4.17). Based on this probability, the equivalents of standard TF and IDF, namely *expected term frequency* (*ETF*) and *expected inverse document frequency* (*EIDF*), can be computed.

As mentioned in the previous section, the duration of the audio elements, which defines the amount of presence of the corresponding term in an audio document, is also a parameter that should be taken into account when computing the weight of the term. Further, it can realistically be assumed that the overall duration of a key term is larger

in its "own" document than in other documents. Therefore, we extend the weight computation scheme to include two additional indicators of term importance, namely *expected term duration* (*ETD*) and *expected inverse document duration* (*EIDD*).

4.3.2.1 ETF and ETD

ETF and *ETD* define the expected occurrence frequency and duration of an audio element in an audio document, respectively. Thus, to calculate *ETF* of audio element e_i in audio document D_k, we first need to compute the probability $P(e_i = e_j)$ for all audio elements e_j obtained from D_k. Then, the *ETF* can be obtained as the normalized weighted sum of the occurrence frequencies of all the audio elements e_j in D_k, where the abovementioned probabilities serve as the weights:

$$ETF(e_i, D_k) = \frac{\sum_j n_j P(e_i = e_j \mid e_j \in D_k)}{\sum_j n_j} = \frac{\sum_{e_j \in D_k} n_j S_{dfv}(e_i, e_j)}{\sum_{e_j \in D_k} n_j} \tag{4.18}$$

Here, $ETF(e_i, D_k)$ is the expected term frequency of audio element e_i in the audio document D_k. It is noted that D_k is not necessarily the document that e_i is obtained from. Further, $P(e_i = e_j \mid e_j \in D_k)$ is the probability that e_i represents the same audio term as the audio element e_j, and is computed using the similarity (4.17). Finally, n_j is the number of occurrences of e_j in the document D_k.

Similarly, $ETD(e_i, D_k)$ can be defined as,

$$ETD(e_i, D_k) = \frac{\sum_j d_j P(e_i = e_j \mid e_j \in D_k)}{\sum_j d_j} = \frac{\sum_{e_j \in D_k} d_j S_{dfv}(e_i, e_j)}{\sum_{e_j \in D_k} d_j} \tag{4.19}$$

where d_j is the total duration of e_j in the document D_k.

4.3.2.2 EIDF and EIDD

Similar to *IDF* in text document analysis, *EIDF* of an audio element e_i can be computed as the log of the number of all documents divided by the expected number of documents containing the audio element e_i. That is,

$$EIDF(e_i) = \log \frac{|D|}{\sum_k P(e_i \in D_k)} \tag{4.20}$$

where |D| is the number of documents, and where $P(e_i \in D_k)$ is the probability that e_i appears in the document D_k. This probability can be calculated as

$$\begin{aligned} & P(e_i \in D_k) \\ & = P(e_i = e_{j1} \quad \cup \quad e_i = e_{j2} \quad \cup \quad ... \quad \cup \quad e_i = e_{jN} \mid e_{j1}, ...e_{jN} \in D_k) \\ & = 1 - \prod_j (1 - P(e_i = e_j \mid e_j \in D_k)) \\ & = 1 - \prod_{e_j \in D_k} (1 - S_{dfv}(e_i, e_j)) \end{aligned} \tag{4.21}$$

It is easy to verify that $P(e_i \in D) = 1$ if the audio element e_i is obtained from the document D.

Similarly, the *EIDD* of audio element e_i can be calculated as the log of the duration of all documents divided by the expected duration of audio element e_i in all documents. As the expected duration of audio element e_i in document D_k is obtained by $ETD(e_i, D_k)$, the *EIDD* can be approximated as,

$$EIDD(e_i) = \log \frac{\sum_k d_{D_k}}{\sum_k ETD(e_i, D_k)} \tag{4.22}$$

where d_{D_k} is the total duration of audio document D_k.

4.3.2.3 Final Weighting

To integrate the defined four importance indicators into the definitive importance weight of an audio term, we realistically assume that four indicators are independent of each other. Also, following the analogy to the text document analysis case, where TF and IDF indicators are simply combined into a product, we follow the same procedure here to compute the overall weight of an audio element e_i in the document D_k:

$$W(e_i, D_k) = ETF(e_i, D_k) \cdot EIDF(e_i) \cdot ETD(e_i, D_k) \cdot EIDD(e_i) \tag{4.23}$$

4.3.3 Number of Key Audio Elements

To this end, the first K audio elements with the highest weight (4.13) or (4.23) could be selected as key audio elements, and used to characterize an audio document in further content-based audio analysis steps. To determine the number K of key audio elements, an intuitive idea would be to set a threshold, and choose an audio element as a key audio element if its weight is larger than the threshold. An alternative method is presented in [Cai et al. 2005], where the number of key audio elements is chosen based on the total duration of the selected key audio elements, as:

$$K = \arg\max_k \{\sum_{i=1}^{k} d_i^{'} \le \eta \cdot L_D\} \tag{4.24}$$

where $d_i^{'}$ denotes the duration of the i-th audio element on the list of audio elements ranked in the descending order based on the weighting score, L_D is the total duration of audio document D, and η is a tuning parameter set experimentally to 0.25. This setting is based on the assumption that the key audio elements will not cover more than 25% of the entire audio document.

However, these methods used thresholds which are usually hard to set and depend on specific applications. In this thesis, we follow the practice of text document analysis and decide not to set a heuristic threshold and make a hard decision on the number of key audio elements to be selected. Instead, we choose to consider all obtained audio elements and their corresponding weights in the further semantic inference steps.

4.4 Experimental Evaluation

In this section, the performance of the proposed unsupervised approach to audio element discovery and key element spotting is evaluated experimentally using a manually annotated representative test audio data set. As no wide benchmarking effort (e.g. a counterpart of TRECVID) exists in the field of content-based audio analysis, we invested considerable effort in optimizing the preparation of our data collection to maximize the reliability of the insights obtained through the experiments. The test audio documents (sound tracks) are extracted from various types of video, including sports, situation comedy, award ceremony and war/action movies, and in the total length of about 5 hours. These sound tracks contain an abundance of different audio elements, and are of different complexity, both in terms of content dynamics and the composite nature of audio signals, in order to provide a reliable base for evaluating the proposed approach under different conditions. For example, in the test dataset, the

sound track of the tennis game is relatively simple, as compared to far more complex sound tracks from the war/action movies “Band of Brothers – Carentan” and “Sword Fish”.

Table 4.1. Information of the experimental audio data

No.	Video	Category	duration
A_1	*Friends*	situation comedy	0:25:08
A_2	*Tennis Game*	sports	0:59:41
A_3	*59th Annual Golden Globe Awards*	award ceremony	1:39:47
A_4	*Band of Brothers - Carentan*	war movie	1:05:19
A_5	*Sword Fish*	Action movie	1:00:00

Detailed information on the sound tracks we used is listed in Table 4.1. As already indicated in Chapter 3 and earlier in this chapter, all audio streams are in 16 KHz, 16-bit and mono channel format, and are divided into frames of 25ms with 50% overlap for feature extraction. To balance the detection resolution and the computational complexity, audio frames are grouped into one-second-long audio segments with 0.5 seconds overlap, which are further used as basic units for audio element discovery.

4.4.1 Audio Element Discovery

In our spectral clustering approach to audio element discovery, the search range for selecting the cluster number is set experimentally as k_{min}=2 and k_{max}=20 for all sound tracks. Moreover, to illustrate the effectiveness of the utilized spectral clustering scheme with context-based scaling factors, we compare this scheme with the “standard” spectral clustering from [Ng et al. 2001].

Table 4.2 shows the detailed comparison results of the two spectral clustering algorithms on the example sound track of "Friends" (A_1). In this sound track, we obtained 7 audio elements using the spectral clustering with context-based scaling factors, and only 5 audio elements using the “standard” spectral clustering. To enable a quantitative evaluation of the clustering performance, we established the ground truth by combining the results obtained by three unbiased persons who analyzed the content of the sound track and the obtained audio elements. This process resulted in 6 sound classes that we labeled as noise (N), speech (S), applause (A), laughter (L), music (M), and laughter with music (L&M). In Table 4.2, each row represents one discovered

Table 4.2. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "Friends" (A_1) (unit: second)

	No.	N	S	A	L	L&M	M	precision
Spectral clustering with context-based scaling factors	1	42	2		0.5			0.944
	2	7	1132.5	1	8			0.986
	3			5				1.000
	4	1	2		215			0.986
	5	3			8	31.5		0.741
	6	0.5					46.5	0.980
	7	0.5					2.5	
	recall	0.778	0.996	0.833	0.929	1.000	1.000	**0.978**
Standard spectral clustering	1	50.5	43.5					0.537
	2	1.5	527.5		4	2		0.977
	3		290	6	2	1	7	
	4		267		1.5			
	5	2	8.5		224	28.5	42	0.734
	recall	0.935	0.954	0.000	0.968	0.000	0.000	**0.901**

Abbr. noise (**N**), speech (**S**), applause (**A**), laughter (**L**), and music (**M**)

audio element and contains the durations (in seconds) of its occurrences in view of the ground truth sound classes. We manually grouped those audio element occurrences associated to the same ground truth class (indicated by shaded fields in the table), and then calculated the precision, recall and accuracy (the duration percentage of the correctly assigned audio segments in the stream) based on the grouping results. These measures roughly represent the overall clustering performance. As shown in Table 4.2, the accuracies of the two algorithms for the sound track of "Friends" (A_1) are in average 97.8% and 90.1%, respectively. We like to emphasize that these performance figures were obtained for the case where all audio segments are treated equally. While one may choose to compute the costs related to the clustering errors in a different way, like for instance weighting the clustering errors of more important (longer) audio elements stronger than those of the less important ones, we considered such adaptation application/domain specific and therefore beyond the scope of this thesis.

Table 4.2 also shows that each class in the ground truth can be covered by the audio elements discovered using the spectral clustering with context-based scaling factors. In the standard spectral clustering, the sounds of applause (A), music (M) and laughter with music (L&M) were missed and falsely included into other clusters, while speech (S) is divided over three discovered audio elements. As demonstrated in Section 4.1.2,

this phenomenon is likely caused by the unharmonious distributions of various sound classes in the feature space. For instance, the feature distribution of speech (S) is relatively sparse and has large divergence (weaker cluster density), while those of music (M) and laughter with music (L&M) are more "tight". The influence of unharmonious sound distributions can be reduced by setting different scaling factors for different data densities, as done in our approach.

Table 4.3. Performance comparison between the spectral clustering with and without context-based scaling factors for all test sound tracks

No.	***#gc***	**Standard spectral clustering**		**Spectral clustering with context-based scaling factors**	
		#nc / #miss	***accuracy***	***#nc / #miss***	***accuracy***
A_1	6	7 / 3	0.747	7 / 0	0.951
A_2	6	5 / 3	0.901	7 / 0	0.978
A_3	7	8 / 2	0.814	11 / 0	0.928
A_4	6	5 / 3	0.621	16 / 0	0.930
A_5	6	2 / 4	0.332	17 / 0	0.494
Avr.	6.2	5.4 / 3	0.683	11.6 / 0	0.856

Table 4.3 summarizes the performance of audio element discovery on all test sound tracks. The table shows the number of ground truth sounds (***#gc***), the number of discovered audio elements (***#nc***), the number of missed ground truth audio elements (***#miss***), and the overall accuracy. It can be seen that by using the standard spectral clustering algorithm, around 48% of sound classes in the ground truth are not properly discovered, and the average accuracy is only around 68%. The table also shows that the spectral clustering with context-based scaling factors performs better on all test sound tracks, and achieves an average accuracy of around 86%. In particular, no sound classes in the ground truth are missed in the obtained set of audio elements. Hence, the use of context-based scaling factors in spectral clustering of complex audio streams can notably improve the clustering performance.

Detailed comparison results for the sound tracks A_2-A_5 are shown in Table 4.4-4.7, where we also manually grouped those audio elements associated with the same ground truth class and represented them by shaded fields. The results reported in these tables confirm that spectral clustering with context-based scaling factors can obtain better results, including a better estimate of the cluster number, better cluster purity, less missed clusters, and a higher recall and precision.

Table 4.4. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "*Tennis*" (A_2) (unit: second)

	No.	S	A+S	A	Sil	B	M	precision
Spectral clustering with context-based scaling factors	1	1594	42	4	14	4		0.961
	2	21	279.5	40	0.5			0.820
	3				1.5		20.5	0.932
	4		9.5	298	12			0.933
	5				834.5	3		0.997
	6				96.5			
	7			3	22.5	282		0.917
	recall	0.987	0.844	0.864	0.949	0.976	1	**0.951**
Standard spectral clustering	1	0	7	19	29.5	35	3	0.534
	2	0.5	15	7	47.5	76	3.5	
	3	90	84	319	893.5	174	14	
	4	692	25		7	4		0.969
	5	751			4			
	6	69.5	8					
	7	12	192					0.941
	recall	0.937	0.580	0	0.989	0	0	**0.747**

Abbr. speech (**S**), applause (**A**), applause with speech (**A+S**), silence (**Sil**), ball-hit (**B**), and music (**M**)

Table 4.5. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "*59th Annual Golden Globe Awards*" (A_3) (unit: second)

	No.	S	S+M	M	A+M	A+S	A	N	precision
Spectral clustering with context-based scaling factors	1	195.5	1				6		0.953
	2	27.5							
	3	2755	18	4	8	38.5	52	18	
	4		6	9	320	4	5	2	0.898
	5		1.5	2	155.5	7		2.5	
	6		2	21	220	11		6	
	7	6	807.5	15	32	3	28	9	0.897
	8		3	38.5	2			7	0.762
	9				2.5	475	8		0.978
	10	32			2	8.5	186	1	0.811

	11	2.5				29	18	374	0.883
	recall	0.987	0.9625	0.43	0.937	0.825	0.614	0.892	**0.928**
Standard spectral clustering	1	122.5	9.5	0.5		0.5	2	5.5	0.193
	2	1091.5	36	2		5		44.5	
	3	640.5	51	9.5	10	12	8	32	
	4	1083				9		41.5	
	5	57	11	56.5	138	526.5	148	227.5	0.452
	6	7	12	14	580	7	24	43	0.844
	7	5	9		2	13	118	21	0.702
	8	12	710.5	7	12	3	3	4.5	0.945
	recall	0.973	0.8468	0	0.782	0.914	0.389	0	**0.814**

Abbr. speech (**S**), music (**M**), speech with music (**S+M**), applause (**A**), applause with music (**A+M**), applause with speech (**A+S**), and noise (**N)**

Table 4.6. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "*Band of Brothers*" (A_4) (unit: second)

	No.	**S**	**NS**	**Sil**	**N**	**G**	**M**	**precision**
Spectral clustering with context-based scaling factors	1	569	8					0.963
	2	11.5			1.5			
	3	181	8			3		
	4	320	13.5				8	
	5		55.5		4	2.5		0.767
	6	1	15		0.5			
	7	12	107	16	11	7		
	8			123	4			0.934
	9	1.5		161	14	1		
	10			21.5			1	
	11	1	2	0.5	17.5	2	1.5	0.902
	12	5		21	461.5	12	7	
	13	4	8	12	38	652		0.916
	14		7		7.5	198	1	
	15	1		8	15.5		710	0.966
	16			1	1		43	
	recall	0.977	0.792	0.839	0.832	0.969	0.976	**0.930**

Standard spectral clustering	1			5		1	545.5	0.987
	2		1.5				40.5	
	3				12	628		0.976
	4	3		3		115		
	5	1104	222.5	356	564	133.5	185.5	0.430
	recall	0.9973	0	0	0	0.847	0.76	**0.621**

Abbr. speech (**S**), noisy speech (**NS**), Silence (**Sil**), noise (**N)**, gun-shot **(G),** and music (**M**)

Table 4.7. Comparison of the results of the standard spectral clustering and the spectral clustering with context-based scaling factors on the sound track of "*Sword Fish*" (A_5) (unit: second)

	No.	**S**	**M**	**B**	**SBM**	**F**	**precision**
Spectral clustering with context-based scaling factors	1	223.5	20	111	129	19	0.519
	2	15	6.5	14	7.5	1	
	3	567.5	18	183	226	11	
	4	0	50.5	14.5	1	2.5	0.626
	5	0	169	128.5	6	8	
	6	0.5	90.5	14.5	4	5.5	
	7	70	47	218.5	92.5	21.5	0.462
	8	28	41.5	91	53	28.5	
	9	175.5	203	652	184.5	51.5	
	10	13.5	37	102	60	5	
	11	81.5	61.5	147.5	140.5	14	
	12	25.5	9	37.5	40.5	2	0.357
	13	21	26	67.5	75	19.5	
	14	8.5	52	43	16.5	108	0.516
	15	0	88	95.5	29	240.5	
	16	2	7	19	27	74.5	
	17	8.5	19	36	16	76	
	recall	0.650	0.328	0.613	0.104	0.725	**0.494**
	1	1238	944.5	1964	1096.5	674.5	0.332
	2	2.5	1	11	11.5	13.5	0.342
	recall	0	0	0.994	0	0.020	**0.332**

Abbr. speech (**S**), speech with background sound or music (**SBM**), Background sounds (**B)**, fighting sounds **(F),** and music (**M**)

It can also be seen that the iterative clustering scheme produces more audio elements than the number of the ground-truth clusters. In other words, over-segmentation is introduced. However, as mentioned before, related audio elements can be recognized as such using the DFV-based similarity metric; and furthermore, groups of related audio elements can also emerge from higher-level content analysis, such as the co-clustering process that we will elaborate on in Chapter 5.

4.4.2 Single Document based Key Audio Element Spotting

Single document based audio element weighting scheme (4.13) is employed when only one audio document is available, that is, we consider each test sound track independently. If we assume that a key audio element has somewhat average occurrence frequency and duration, the model parameters in (4.10)-(4.12) can simply be set all to 1. Our experiments showed that this assumption worked well in a general case. However, in order to investigate the effect of different parameter values, we also tuned the parameters differently for different soundtracks. This parameter tuning was not sophisticated, but simply realized through a rough sampling of the relevant parameter space using the parameter options from the set (0.5, 1.0, 1.5, 2.0). The results reported in this section are the best ones obtained for different parameter settings. The rationale behind this approach is that if our method is applied in a given domain or use case, general domain knowledge can be used to roughly set the model parameters. We wanted to provide an indication regarding the expected performance in such a case.

The results of key elements spotting are listed for each test sound track in Table 4.8 - 4.12, respectively, and summarized in Table 4.13. The semantic labels provided in these tables (and also in tables 4.14-4.19) are assigned manually to audio elements after these elements are obtained. These labels serve only to roughly describe the major content of an audio element, in order to be able to evaluate the meaningfulness of the obtained audio elements regarding the part of our test data set from which they are extracted. Furthermore, all audio segments belonging to one audio element cluster are characterized by one label only. If the composite nature of the sounds in this cluster is complex, we did our best to reveal this complexity when defining the label. For example, next to the audio elements simply defined as “speech”, we also identified audio elements that can best be described as “laughter with music” or “speech with background music”. Similarly, different audio elements may be characterized using variations of one and the same label (due to over-segmentation). For instance, we found that a large portion of speech segments is indeed likely to be grouped into the cluster adopted as the “speech” audio element. However, due to the fact that different speech

segments may have rather different signal characteristics, what we would consider speech might get spread over several audio element clusters that we then refer to as “speech 1” or “speech 2”. Each of these clusters is treated further as separate audio elements. Finally, the difference between “speech X” elements and the element “speech (with gunshot background)” is in the purity of the speech component, which is lower in the latter case due to the gunshot noise in the background. This is why we also labeled it in a different way.

As an example, Table 4.9 shows the results of sound track "Tennis" (A_2), with the best parameters α, β, and γ set to 1. For 7 discovered audio elements in the audio document, the table lists their total duration (***dur***), the occurrence times (***occu***), the average occurrence length (***avgl***), and the final importance score. Based on these scores, an "educated guess" can be made for the most likely key audio elements. For example, in this tennis soundtrack, the audio elements indicated by the shaded fields, including *applause with speech*, *applause*, and *ball-hit*, have the highest importance scores, and therefore can be taken as key audio elements.

Table 4.8 Single document based audio element score on the track of "Friends" (A_1)

No.	Description	*occu*	*dur*	*avgl*	*score*
1	speech + noise	27	44.5	1.6481	0.642
2	laughter	102	218.0	2.1373	0.890
3	theme music	1	47.0	47	0.015
4	laughter + music	9	42.5	4.7222	0.503
5	speech	124	1148.5	9.2621	0.039
6	applause + cheering	1	5.0	5	0.413
7	TV music	1	3.0	3	0.407

Table 4.9 Single document based audio element score on the track of "Tennis" (A_2)

No.	Description	*occu*	*dur*	*avgl*	*score*
1	clean speech	250	1658.0	6.632	0.020
2	speech + applause	108	341.0	3.157	0.928
3	music	1	22.0	22.00	0.008
4	applause	106	319.5	3.014	0.908
5	silence	173	837.5	4.841	0.633
6	noisy silence	32	96.5	3.016	0.399
7	ball-hit	145	307.5	2.121	0.820

Table 4.10 Single document based audio element score on the track of "Golden Global Awards" (A_3)

No	Description	*occu*	*dur*	*avgl*	*score*
1	speech 1	132	202.5	1.534	0.380
2	speech 2	26	27.5	1.058	0.150
3	music + applause 1	110	346	3.146	0.880
4	music + applause 2	72	168.5	2.340	0.544
5	music + speech	161	900.5	5.593	0.510
6	music	22	50.5	2.296	0.366
7	applause	143	485.5	3.395	0.959
8	speech + applause	109	229.5	2.106	0.553
9	background noise	211	423.5	2.007	0.470
10	(dense) music + applause	68	260.0	3.824	0.819
11	speech 3	487	2893.5	5.942	0.000

Table 4.11 Single document based audio element score on the track of "Band of Brother" (A_4)

No	Description	*occu*	*dur*	*avgl*	*score*
1	speech	187	577	3.086	0.463
2	speech (gun background)	25	62.0	2.48	0.122
3	speech	1	13.0	13.0	0.069
4	speech	72	192.0	2.667	0.316
5	heavy noise	11	24.5	2.227	0.081
6	silence (some noise)	44	127.0	2.886	0.212
7	noise	143	506.5	3.542	0.662
8	speech	122	341.5	2.799	0.529
9	gunshot + speech 1	128	714.0	5.578	0.731
10	gunshot + speech 2	85	213.5	2.512	0.35
11	background sounds	51	177.5	3.480	0.297
12	applause	3	16.5	5.50	0.120
13	music	48	734.5	15.30	0.141
14	music + speech	4	45.0	11.25	0.116
15	noise + speech	86	153.0	1.780	0.251
16	silence (with HF noise)	3	22.5	7.50	0.137

Table 4.12 Single document based audio element score on the track of "Sword Fish" (A_5)

No	Description	*occu*	*dur*	*avgl*	*score*
1	speech+ backgrounds	235	449.5	3.826	0.215
2	fighting sounds 1	57	228.0	8.0	0.654
3	fighting sounds 2	155	453.0	5.845	0.561
4	speech + backgrounds	86	242.0	5.628	0.499
5	speech	241	502.5	4.170	0.225
6	mixed backgrounds	21	44.0	4.191	0.203
7	speech	363	1005.5	5.540	0.026
8	speech + backgrounds	73	114.5	3.137	0.212
9	speech + backgrounds	121	209.0	3.455	0.275
10	backgrounds	404	1266.5	6.270	0.004
11	speech in repressive env.	67	129.5	3.866	0.265
12	music	13	69.0	10.612	0.421
13	fighting sounds	76	155.5	4.092	0.300
14	backgrounds	102	217.5	4.265	0.355
15	speech + backgrounds	247	445.0	3.603	0.182
16	music	45	311.5	13.844	0.654
17	music	23	115.0	10.0	0.503

The importance scores obtained for all audio elements from all test sound tracks are summarized in Table 4.13. For each sound track, the number of audio elements (***#ele***), the parameter setting, and the description of each audio element with corresponding weighting score are listed in the descending order. The audio elements indicated in bold correspond to ground truth, which is established here again by combining the results obtained by three unbiased persons who analyzed the content of the test sound tracks and selected the most characteristic sounds or sound combinations.

From the table, it can be noted that the performance on audio documents A_2 and A_3 is satisfying. All the key elements manually picked are among the highest-ranked elements. On the other hand, in audio document A_1, A_4 and A_5, some audio elements not included in the ground truth are also ranked high (that is, false alarms are introduced). For example, the *speech with noise* in A_1 is falsely ranked as second important, since it has similar occurrence frequency and duration as the expected key elements. Similar cases are also found for the audio elements *speech* in A_4 and *music* in A_5. Also in A_4, some key audio elements such as the *gunshot with speech* is not ranked high enough, since the characteristics of key elements in complex audio documents vary too much. These problems indicate that the proposed heuristic rules do not

perform entirely as expected in complex audio documents. However, the overall performance of key element spotting using the proposed rules on our test set is still acceptable. If we take the first four audio elements as key audio elements in each audio document, more than 85% (12 out of 14) of the key audio elements in the ground truth can be properly recalled.

Table 4.13. Single document based weighting for audio elements obtained in all the sound tracks

No.	*#ele.*	*(α,β,γ)*	Discovered audio elements and corresponding weight
A_1	7	(2,1,1)	**laughter**(0.89), speech + noise(0.642), **laughter + music**(0.503), **applause + cheering**(0.413), TV music(0.407), speech(0.039), theme music(0.015)
A_2	7	(1,1,1)	**speech + applause**(0.928), **applause**(0.908), **ball-hit**(0.82), silence(0.633), noisy silence(0.399), clean speech(0.02), music(0.008)
A_3	11	(1,1,1)	**applause**(0.959), **music + applause 1**(0.88), **(dense) music + applause**(0.819), **speech + applause**(0.553), **music + applause 2**(0.544), music + speech(0.51), background noise(0.47), speech 1(0.38), music(0.366), speech 2(0.15), speech 3 (0.0)
A_4	16	(2,2,2)	**gunshot + speech 1**(0.731), noise(0.662), speech(0.529), speech(0.463), **gunshot + speech 2**(0.35), speech(0.316), background sounds(0.297), noise + speech(0.251), silence (some noise)(0.212), music(0.141), silence (with HF noise)(0.137), speech (gunshot background)(0.122), applause (0.12), music + speech(0.116), heavy noise(0.081), speech(0.069)
A_5	17	(0.5, 1.5,2)	**fighting sounds 1**(0.654), music(0.654), **fighting sounds 2**(0.561), music(0.503), speech + backgrounds(0.499), music(0.421). backgrounds(0.355), fighting sounds (0.3), speech + backgrounds (0.275), speech in repressive env.(0.265), speech(0.225), speech+ backgrounds (0.215), speech + backgrounds(0.212), mixed backgrounds (0.203), speech + backgrounds(0.182), speech(0.026), backgrounds(0.004)

4.4.3 TFIDF-based Audio Element Weighting

In this experiment, we employ the whole test audio set to estimate the importance indicators from Section 4.3, and then use these indicators for audio element weighting. Table 4.14 - 4.18 show the results for each test sound track, respectively. In these tables, we not only list the total number of occurrences (***occu***) and total duration (***dur***)

of each audio element, but also the derived *ETF*, *EIDF*, *ETD*, *EIDD* values and the final importance weight.

Based on the data collected in these tables, situations can be analyzed that led to a particular weight. For example, the 6th audio element *applause with cheering* in Table 4.14, although occurring only once and lasting only 5 seconds in this track, occurs statistically even less in other audio tracks. This makes its *EIDF* (2.15), *EIDD* (2.135) and the final weight high. On the other hand, the 5th audio element *music with speech* and the 11th audio element *speech* in Table 4.16, although appearing many times and having long durations (161 times / 900.5 seconds, and 487 times / 2893.5 seconds, respectively), seem to appear often in other soundtracks as well. Thus, their *EIDF*, *EIDD* and the final weight are low. These results show that the *TF* and *IDF* concepts from text analysis are indeed applicable to general audio signals.

Table 4.14. TFIDF based audio element weighting on the track of "Friends" (A_1)

No	Description	*occu.*	*dur.*	*ETF*	*EIDF*	*ETD*	*EIDD*	*weight*
1	speech + noise	27	44.5	0.59	0.588	0.691	1.046	0.251
2	laughter	102	218.0	0.699	1.411	0.61	1.597	0.96
3	theme music	1	47.0	0.236	1.466	0.501	1.582	0.274
4	laughter + music	9	42.5	0.515	1.234	0.525	1.421	0.474
5	speech	124	1148.5	0.785	0.674	0.897	0.967	0.459
6	applause+cheering	1	5.0	0.496	2.15	0.392	2.135	0.892
7	TV music	1	3.0	0.036	1.711	0.038	2.587	0.006

Table 4.15. TFIDF based audio element weighting on the track of "Tennis" (A_2)

No	Description	*occu.*	*dur.*	*EDF*	*IDF*	*EDD*	*IDD*	*weight*
1	clear speech	250	1658.0	0.576	0.177	0.66	0.639	0.043
2	speech + applause	108	341.0	0.555	0.304	0.571	0.779	0.075
3	music	1	22.0	0.431	0.651	0.409	1.135	0.13
4	applause	106	319.5	0.42	1.194	0.358	1.464	0.262
5	silence	173	837.5	0.533	0.934	0.491	1.262	0.308
6	noisy silence	32	96.5	0.465	1.117	0.404	1.452	0.304
7	ball-hit	145	307.5	0.641	0.54	0.598	0.92	0.19

Table 4.16. TFIDF based audio element weighting on the track of "Golden Global Awards" (A_3)

No	Description	*occu.*	*dur.*	*EDF*	*IDF*	*EDD*	*IDD*	*weight*
1	speech 1	132	202.5	0.646	0.249	0.705	0.81	0.092
2	speech 2	26	27.5	0.607	0.304	0.669	0.91	0.112
3	music + applause 1	110	346.0	0.713	0.395	0.691	0.767	0.149
4	music + applause 2	72	168.5	0.681	0.45	0.654	0.832	0.167
5	music + speech	161	900.5	0.752	0.158	0.795	0.543	0.051
6	music	22	50.5	0.512	0.349	0.544	0.784	0.076
7	applause	143	485.5	0.506	1.043	0.458	1.374	0.332
8	speech + applause	109	229.5	0.705	0.358	0.708	0.802	0.143
9	background noise	211	423.5	0.747	0.216	0.739	0.622	0.074
10	(dense) music + applause	68	260.0	0.622	0.363	0.623	0.814	0.114
11	speech 3	487	2893.5	0.776	0.161	0.829	0.581	0.06

Table 4.17. TFIDF based audio element weighting on the track of "Band of Brother" (A_4)

No.	Description	*occu.*	*dur.*	*EDF*	*IDF*	*EDD*	*IDD*	*weight*
1	speech	187	577	0.684	0.132	0.621	0.568	0.032
2	speech (gun back-ground)	25	62.0	0.624	0.188	0.588	0.591	0.041
3	speech	1	13.0	0.157	1.446	0.151	2.108	0.072
4	speech	72	192	0.662	0.119	0.601	0.529	0.025
5	heavy noise	11	24.5	0.501	0.396	0.447	0.921	0.082
6	silence (some noise)	44	127.0	0.438	0.636	0.37	1.173	0.121
7	noise	143	506.5	0.648	0.218	0.579	0.609	0.05
8	speech	122	341.5	0.667	0.125	0.611	0.541	0.027
9	gunshot + speech 1	128	714.0	0.4	0.704	0.402	1.069	0.121
10	gunshot + speech 2	85	213.5	0.279	1.225	0.291	1.381	0.137
11	background sounds	51	177.5	0.452	0.607	0.384	1.208	0.127
12	applause	3	16.5	0.263	0.912	0.239	1.493	0.085
13	music	48	734.5	0.42	0.517	0.482	1.022	0.107
14	music + speech	4	45.0	0.423	0.475	0.447	1.014	0.091
15	noise + speech	86	153.0	0.706	0.158	0.658	0.544	0.04
16	silence (w/ HF noise)	3	22.5	0.23	0.926	0.225	1.615	0.077

Table 4.18. TFIDF based audio element weighting on the track of "Sword Fish" (A_5)

No	Description	*occu.*	*dur.*	*EDF*	*IDF*	*EDD*	*IDD*	*weight*
1	speech+backgrounds	235	449.5	0.811	0.176	0.795	0.514	0.058
2	fighting sounds 1	57	228.0	0.31	1.017	0.334	1.557	0.164
3	fighting sounds 2	155	453.0	0.645	0.242	0.645	0.72	0.073
4	speech+backgrounds	86	242.0	0.593	0.413	0.573	0.982	0.138
5	speech	241	502.5	0.759	0.107	0.74	0.507	0.031
6	mixed backgrounds	21	44.0	0.528	0.562	0.524	1.107	0.172
7	speech	363	1005.5	0.793	0.115	0.771	0.531	0.037
8	speech+backgrounds	73	114.5	0.782	0.101	0.763	0.534	0.032
9	speech+backgrounds	121	209.0	0.692	0.197	0.67	0.585	0.053
10	backgrounds	404	1266.5	0.737	0.222	0.725	0.587	0.07
11	speech in repr. env.	67	129.5	0.413	0.566	0.401	1.042	0.098
12	music	13	69.0	0.506	0.468	0.497	0.962	0.113
13	fighting sounds	76	155.5	0.345	0.877	0.34	1.347	0.138
14	backgrounds	102	217.5	0.604	0.448	0.588	0.929	0.148
15	speech+backgrounds	247	445.0	0.818	0.16	0.799	0.559	0.058
16	music	45	311.5	0.601	0.388	0.609	0.83	0.118
17	music	23	115.0	0.394	0.653	0.402	1.248	0.129

The final *TFIDF*-based importance scores for all test sound tracks are summarized in Table 4.19. This table also lists the number of audio elements (***#ele***), the description of each audio element and the corresponding weighting score sorted in the descending order, and with the collected ground truth indicated in bold. The table shows that most of key audio elements in the ground-truth are correctly ranked high, such as the *laughter*, *applause with cheering*, and *laughter with music* in A_1, and the *applause* and *music with applause* in A_3.

If we also take the first four audio elements as key audio elements in each audio document, 11 out of 14 can be properly recalled. At first sight, this performance seems not as good as that based on heuristic rules. However, after further analysis, we find that other audio elements ranked high by *TFIDF*-based scheme are also quite representative to the audio document, although they are not included in the ground-truth. For example, two *silence* elements found in *Tennis* soundtrack (A_2) are assigned the highest weights (the silence segments between every two ball-hits are clustered together). This is justifiable since *silence* periods are very representative for the game and also are not that pronounced in other sound tracks in the test set. Also, in the war and action movie (A_4 and A_5), some movie-specific *background sounds*, are

reasonably ranked high. These sounds are not selected as ground truth since test subjects tend to choose the highlights as representative sounds rather than background sounds.

Table 4.19. TFIDF based weighting for audio elements obtained in all sound tracks

No.	*#ele.*	Discovered audio elements and corresponding weight
A_1	7	**laughter**(0.96), **applause + cheering** (0.892), **laughter + music** (0.474), speech (0.459), theme music (0.274), speech + noise (0.251), TV music (0.006)
A_2	7	silence(0.308), noisy silence(0.304), **applause**(0.262), **ball-hit(**0.19), music(0.13), **speech + applause** (0.075), clean speech(0.043)
A_3	11	**applause**(0.332), **music + applause 2**(0.167), **music + applause 1(**0.149), **speech + applause(0.143)**, **(dense) music + applause**(0.114), speech 2 (0.112), speech 1 (0.092), music(0.076), background noise(0.074), speech 3 (0.06), music + speech(0.051)
A_4	16	**gunshot + speech 2**(0.137), background sounds(0.127), silence (some noise)(0.121), **gunshot + speech 1**(0.121), music(0.107), music + speech(0.091), applause(0.085), heavy noise(0.082), silence (with HF noise)(0.077), speech(0.072), noise(0.05), speech (gunshot background)(0.041), noise + speech(0.04), speech(0.032), speech(0.027), speech(0.025)
A_5	17	mixed backgrounds (0.172), **fighting sounds 1**(0.164), backgrounds(0.148), speech+backgrounds (0.138), **fighting sounds 2**(0.138), music(0.129), music(0.118), music(0.113), speech in repressive env.(0.098), fighting sounds 2(0.073), backgrounds(0.07), speech+backgrounds (0.058), speech+backgrounds(0.058), speech+backgrounds(0.053), speech(0.037), speech+backgrounds(0.032), speech(0.031)

4.4.4 Discussion

Based on the obtained results, we find that both approaches (scheme (4.13) and scheme (4.23)) can achieve reasonable results. As also confirmed by our test panel consisting of three subjects, most of the high-weighted audio elements indeed correspond to the most important or representative sounds in the test sound tracks.

The obtained results also indicate that two approaches of audio element weighting are biased in different way. The single document based weighting scheme (4.13) gives

high weights to those audio elements which satisfy some pre-defined criteria regarding signal behavior. It is therefore suitable for some specific applications relying on prior knowledge and contextual information. The *TFIDF*-based weighting scheme (4.23), on the other hand, usually gives high weights to document-specific elements, that is, the elements frequently appearing in their "own" document but hardly occurring in other documents. A good example here is the element *speech with noise*, which obtained the second highest score in "Friends" (A_1) based on (4.13), since it satisfies the expected occurrence frequency and duration of a key audio element. However, if (4.23) is used, the low *EIDF* value pulls its weight down, and reveals that this element also appears frequently in other documents. Compared to this, a document-specific sound, *applause with cheering* has received the second highest score based on (4.23). This sound occurs statistically much less in other audio documents, which makes its *EIDF*, *EIDD* and the final weight high. While the nature of the *TFIDF*-based scheme makes it suitable for more generic applications than the scheme based on single-document analysis, we see an interesting challenge in combining the two schemes to improve the results even further. This new scheme could namely make use of the available prior knowledge and optimally combine it with the reliable statistics on audio signal behavior derived from the available multiple audio documents.

Chapter 5

Audio Scene Detection and Clustering

Based on the discovered audio elements and their importance weights, we proceed in this chapter with the development of a method to parse an audio document into audio scenes and group these scenes into meaningful clusters. In our approach, audio scenes are characterized, detected, and grouped based on the audio elements they contain, just as the paragraphs of a text document can be characterized, detected and grouped using a vector of words and their weights. As we stated before, utilizing this mid-level audio content representation enables us to split the semantics inference process into two steps, which leads to more robustness compared to inferring the high-level semantics from the features directly.

5.1 Audio Scene Segmentation

5.1.1 Comparative Study

In order to optimally position our proposed audio scene segmentation approach with respect to the previous work on the subject, we start this section by a comparative study

---

This chapter is based on the following publications (also to be found in the list of references):

- Lu, L, Cai, R. and Hanjalic, A. "Audio Elements based Auditory Scene Segmentation". *Proc. 31th Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. V, 17-20, 14-19, 2006
- Lu, L, and Alan Hanjalic. "Text-Like Segmentation of General Audio for Content-Based Retrieval", *IEEE Trans. on Multimedia*, vol. 11, no.4, 658-669, 2009
- Cai, R. Lu, L, and Hanjalic, A. "Co-clustering for Auditory Scene Categorization," *IEEE Trans. on Multimedia*, vol. 10, no. 4,596-606, 2008

considering feature-based approaches (e.g. [Venugopal et al. 1999][Sundaram and Chang 2000][Chen and Gopalakrishnan 1998]), our previous work involving key audio elements ([Lu et al. 2005][Cai et al. 2005]), and some related methods addressing the problem of video segmentation [Kender and Yeo 1998][Hanjalic et al. 1999]. For this purpose, we illustrated different classes of audio segmentation approaches in Fig. 5.1.

Most previous works that aimed at extracting higher-level audio content semantics either assumed the audio scenes were manually pre-segmented [Cai et al. 2005][Cheng et al. 2003], or rely on a direct feature-based analysis to automate the segmentation step. There, audio segments are typically defined to coincide with a consistent feature behavior. For example, a method was introduced in [Venugopal et al. 1999] to segment an audio stream in terms of speech, music, speaker and gender based on the features like tonality, bandwidth, excitation patterns, tonal duration, and energy. In [Sundaram and Chang 2000], a segmentation method was presented that uses the features, such as cepstral and cochlear decomposition, combined with the listener model and various time scales. Finally, the method [Chen and Gopalakrishnan 1998], originally proposed for speaker segmentation in broadcast news speech and employing information-theoretic measurements of signal consistency across the control point within a sliding window, was adopted in [Ellis and Lee 2004] and applied to general audio data. A typical feature-based approach is illustrated in Fig. 5.1(a). There, the consistency of the feature behavior is measured within a sliding window W and across the control point at the time stamp t in the middle of the window. If the inconsistency is larger than a predefined threshold, a boundary is detected at the time stamp of the control point.

These and similar approaches to audio parsing have proved effective for many applications, and in particular for those where knowledge about the presence and distribution of the basic audio modalities (speech, music, and noise) is critical. However, for other applications, like those where higher-level semantic concepts (e.g. audio scenes) become interesting, the feature-based approaches usually can not handle large content diversity of such semantic concepts, and therefore typically result in a (heavy) over-segmentation of an audio document.

In [Lu et al. 2005], a simple segmentation scheme was presented that employs a pre-defined set of key audio elements. As shown in Fig. 5.1(b), two adjacent key audio elements are assumed to belong to different audio scenes if the time interval Δt between them is longer than a prespecified threshold T. In this way, the boundaries of an audio scene are marked by the first and the last key audio element in a series of adjacent key audio elements following each other tightly over time. Clearly, the algorithm is naive and does not fully exploit the relationship between audio elements and audio scenes. To improve the detection performance, [Cai et al. 2005] introduced

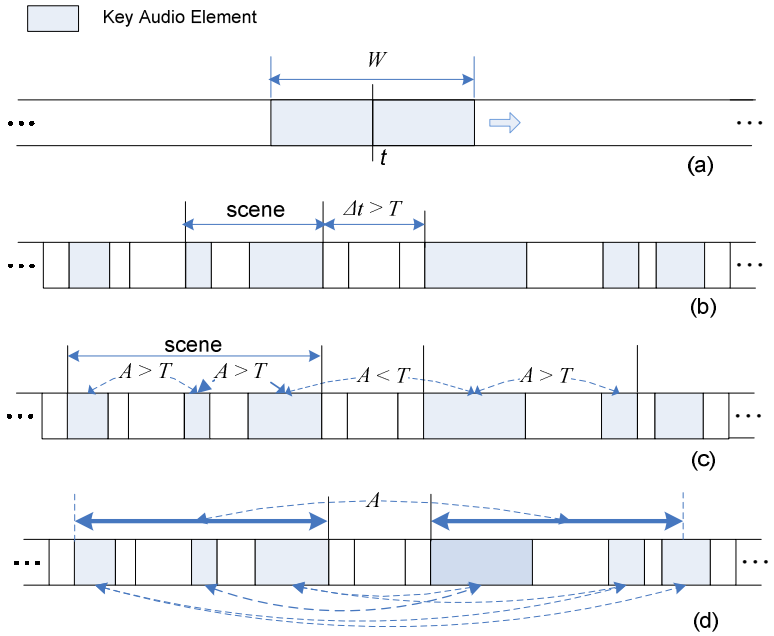


Fig. 5.1 An illustration of different classes of previous approaches to audio scene segmentation, where the vertical lines symbolize scene boundaries or candidate boundaries (a) using feature behavior consistency within a sliding window, (b) using time interval between key audio elements, (c) using semantic affinity between neighboring key audio elements, and (d) investigating the relationship among the audio elements on a larger temporal scale.

the notion of *semantic affinity A* between two contiguous key audio elements, as an exponential function of the time interval separating the elements. As shown in Fig. 5.1(c), an audio scene boundary is found between two adjacent key audio elements if their semantic affinity is below a predefined threshold *T*. However, in both cases the real audio scene boundary falls somewhere between the two limiting key audio elements, which makes precise audio document segmentation not possible in a general case.

It may be too strict to base the detection of audio scene boundaries on the comparison of two subsequent key audio elements only. A more intuitive approach would be to allow more flexibility in the ordering of key audio elements, as long as

their mutual distance remains acceptable, which is similar to some classical video scene segmentation approaches [Kender and Yeo 1998][Hanjalic et al. 1999]. As illustrated in Fig. 5.1(d), an approach in this direction would decide about the presence of a scene boundary at time stamp *t* based on an investigation of the semantic affinity *A* between (key) audio elements taken from a broader range and surrounding this time stamp.

5.1.2 Proposed Approach

The performance of the segmentation methods described above strongly depends on the definition of a key audio element and the reliability of its detection. Crisply defining key audio elements and detecting them in composite audio documents may be rather difficult due to various combinations of multiple superimposed audio modalities. Therefore, a more reliable solution would be to work with all audio elements instead, and rely on their importance weights obtained as explained in Chapter 4. In view of this and the abovementioned broader-range investigation, we propose a novel approach to audio scene segmentation, in which

- we first revise the basic concept of semantic affinity [Cai et al. 2005] by working with all audio elements and their importance weights, and by considering the co-occurrence information for each pair of audio elements, and
- we adapt the successful concept of *content coherence* known from video segmentation [Hanjalic 2004] to evaluate the semantic affinity values obtained along an audio document and to infer the presence of audio scene boundaries.

We illustrate our proposed approach in Fig. 5.2 that shows an example audio element sequence. There, each block belongs to an audio element and different audio elements are represented by different grey values. Each time stamp separating two audio elements can now be considered a potential audio scene boundary, and the confidence of having an audio scene boundary at the observed time stamp can be obtained by computing the semantic affinity between the audio segments drawn from the left and right audio element "buffers" (indicated as *L-Buf* and *R-Buf* in Fig. 5.2) surrounding that time stamp. The two buffers jointly form the sliding window in which the analysis for the observed time stamp (middle of the window) is performed.

In the following sections, we first define a new measure of semantic affinity between two audio segments. Then, an intuitive segmentation scheme is presented in which the proposed affinity measure is used to compute the confidence of having an audio scene boundary at a given time stamp in a composite audio data stream.

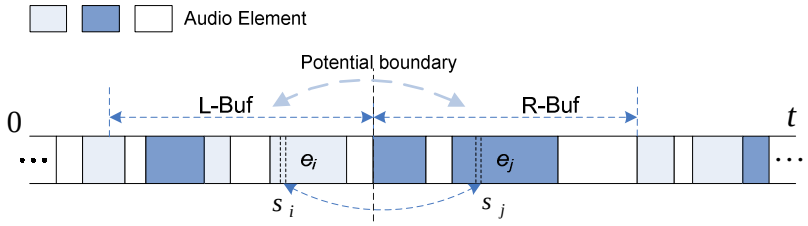


Fig. 5.2 An illustration of the proposed approach to audio element based audio scene segmentation, where s_i and s_j are two audio segments, and e_i and e_j are their corresponding audio element identities

5.1.3 Semantic Affinity Measure

Semantic affinity is introduced as a mean to determine whether two audio segments are likely to belong to the same semantic scene. Following the idea of text document analysis, the measure for semantic affinity should not be based on the low-level similarity between two audio segments, but on their joint ability to mark a meaningful (semantically coherent) piece of audio. We measure this joint ability for two audio segments by observing the co-occurrence statistics and the importance weights of the audio elements contained therein and the time interval separating the segments. Inspired by the video segmentation approach from [Kender and Yeo 1998], our definition of semantic affinity is based on the following intuitive assumptions:

1) there is a high affinity between two audio segments if the corresponding audio elements usually occur together;

2) the larger the time interval between two audio segments, the lower is their semantic affinity, and

3) the higher the importance weights of the corresponding audio elements, the more important is the role these elements will play in the segmentation process, and therefore the more significant the semantic affinity value computed between them will be.

In view of the above assumptions, the semantic affinity between audio segments s_i and s_j can be computed as a function consisting of three components, each of which reflects one of the assumptions stated above. We propose the following measure:

$$A(s_i, s_j) = Co(e_i, e_j) e^{-T(s_i, s_j)/T_m} W(e_i, D_k) W(e_j, D_k) \tag{5.1}$$

Here, the notation e_i and e_j is used to indicate the audio element identities of the elementary audio segments s_i and s_j, that is, to describe their content (e.g. speech, music, noise, or any combination of these). $W(e_i, D_k)$ and $W(e_j, D_k)$ are the importance weights of audio elements e_i and e_j in the audio document D_k. $T(s_i, s_j)$ is the time interval between the audio segments s_i and s_j, and T_m is a scaling factor, the value of which is selected based on the discussions on human memory limit [Sundaram, and Chang 2000]. The exponential expression for $T(s_i, s_j)$ is inspired by the content coherence computation formula introduced in [Kender and Yeo 1998] to model the "content recall" in the context of video segmentation. Finally, $Co(e_i, e_j)$ stands for the co-occurrence between two audio elements e_i and e_j in the audio document D_k.

To estimate the co-occurrence between two audio elements, we rely on the average time interval between them as a reference. The shorter this average time interval, the higher is the co-occurrence probability. Inspired by this, the value $Co(e_i, e_j)$ is estimated using the procedure that is summarized in the following three steps:

1) We first compute D_{ij}, the average time interval between audio elements e_i and e_j. This value is obtained by investigating the neighborhoods of the observed audio elements in the input audio stream. For each occurrence of audio element e_i, the nearest occurrence of e_j is found, and then D_{ij} is obtained as the average temporal distance between these two occurrences.

2) As an analogy to D_{ij}, we also compute D_{ji}. It is clear that D_{ij} may not be equal to D_{ji} in some cases;

3) We then compute the co-occurrence value as

$$Co(e_i, e_j) = e^{-\frac{D_{ij}+D_{ji}}{2\mu_D}} \tag{5.2}$$

where μ_D is the mean of all D_{ij} and D_{ji} values. The choice for an exponential formula in (5.2) is made to keep the influence of audio element co-occurrence on the overall semantic affinity comparable with the influence of the time interval between the audio segments (5.1).

Having defined the semantic affinity (5.1), we can now compute the confidence of being within an audio scene at the time stamp t. To do this, we adopt the general idea of *overlapping similarity links* [Hanjalic 2004] introduced in various forms in previous works on video segmentation (e.g. [Kender and Yeo 1998][Hanjalic et al. 1999]). Based on this idea, the more similarity links can be established between audio segments surrounding a given time stamp t, and the stronger these links are, the higher is the confidence that this time stamp is within an audio scene. Therefore, we choose to

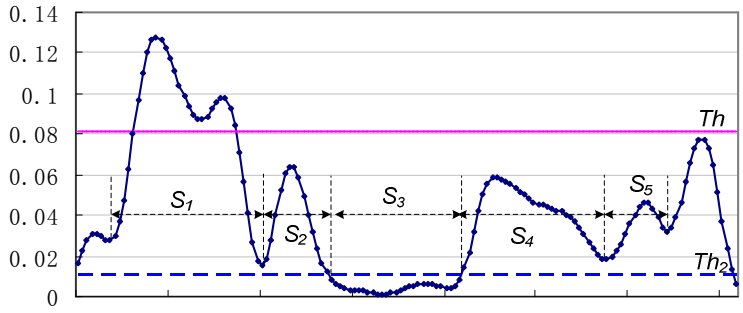


Fig. 5.3 An example of the smoothed confidence curve and the audio scene segmentation scheme, where S_1~S_5 are five obtained audio scenes and *Th* and Th_2 are two thresholds.

compute this confidence simply by averaging the semantic affinity values computed for all pairs of segments s_i and s_j surrounding the *t*, that is,

$$C(t) = \frac{1}{N_l N_r} \sum_{i=1}^{N_l} \sum_{j=1}^{N_r} A(s_i, s_j) \tag{5.3}$$

where N_l and N_r are the numbers of audio segments considered on the left and right from the potential boundary (as captured by the intervals *L-Buf* and *R-Buf* in Fig. 5.2).

5.1.4 Segmentation Scheme

By combining the expressions (5.1) and (5.3), the confidence measure is written as,

$$C(t) = \frac{1}{N_l N_r} \sum_{i=1}^{N_l} \sum_{j=1}^{N_r} Co(e_i, e_j) e^{-T(s_i, s_j)/T_m} W(e_i, D_k) W(e_j, D_k) \tag{5.4}$$

Using this expression, a confidence curve can be obtained over the timeslots of potential boundaries, as illustrated in Fig. 5.3. The boundaries of audio scenes can now be obtained simply by searching for local minima of the curve. In our approach, we first smooth the curve by using a median filter and then find the audio scene boundaries at the places where the following criteria are fulfilled:

$$C(t) < C(t+1); \quad C(t) < C(t-1); \quad C(t) < Th \tag{5.5}$$

Here, the first two conditions guarantee a local valley, while the last condition prevents high valleys from being detected. The threshold *Th* can be set experimentally (as will be discussed in the experimental section).

The obtained confidence curve is likely to contain long sequences of low confidence values, as shown by the segment S_3 in Fig. 5.3. These sequences typically consist of audio elements representing various background sounds, which are weakly related to each other and also have low importance weights. Since it is not reasonable to divide such a sequence into smaller segments, or to merge them into neighboring audio scenes, we choose to isolate these sequences by including all consecutive audio segments with low affinity values into a separate audio scene. Detecting such scenes is an analogy to detecting pauses in speech. Inspired by this, we set the corresponding threshold (Th_2 in Fig. 5.3) by using an approach similar to background noise level detection in speech analysis [Wang et al. 2003].

5.2 Audio Scene Clustering

Clustering theory [Jain and Dubes 1988] provides the most intuitive framework for grouping semantically similar scenes together in an unsupervised fashion. In traditional one-way (or one-directional) clustering algorithms such as K-means, the similarity between two scenes is estimated by measuring the distances among the relevant points in the feature space, and by assuming that each feature provides equal contribution to the distance measure. However, due to the likely grouping trends (co-occurrences) among the features, such assumption is not always satisfied in practice and usually leads to a suboptimal clustering performance,. In this section, we first investigate local grouping trends among audio elements, and then explain how such trends can positively affect the measurement of audio scene similarity. To employ these grouping trends to effectively group audio scenes, we propose an approach based on *co-clustering*. Co-clustering (also referred to as *bi-clustering*) is a simultaneously bidirectional clustering algorithm, which has already been employed successfully in other research fields like bioinformatics [Hanisch et al. 2002][Madeira and Oliveira 2004] and text analysis [Dhillon 2001][Dhillon and Guan 2003][Dhillon et al. 2003], mostly acting as a tool for generating co-occurrence statistics. In this chapter, we show that co-clustering can lead to better audio scene grouping results than the traditional one-way clustering approaches. Moreover, while the cluster number in the existing co-clustering algorithms is assumed to be known beforehand, we introduce a method to automatically select the optimal cluster number by applying the Bayesian information criterion.

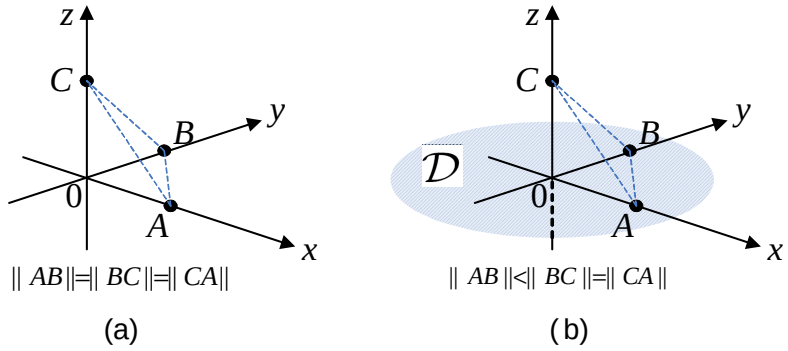


Fig. 5.4. Local feature grouping trend and its influence on the distance measure

5.2.1 On Local Grouping Trends

It is known from the field of pattern classification that different features (or feature combinations) may provide different contributions in distinguishing between the samples belonging to different classes. Various algorithms, such as the Bayesian structural EM [Friedman 1998], can be utilized to learn this information and estimate optimal models and parameters for improving classification accuracy. However, in the unsupervised case, the absence of training samples makes it difficult to discover such latent relations between features and data clusters, although these relations exist and also affect the clustering performance. Mapped onto the specific problem addressed in this section, this would mean that given the audio scenes and their representations in the form of audio elements, and with the absence of any deeper analysis, each individual audio element will contribute equally to the distance measure, what may not always be a reasonable approach in practice.

In order to improve the reliability of scene clustering in a general case, we investigate how the relations among audio elements change locally depending on various potential audio scene clusters. We will refer to such relations further as the *local grouping trends*. In the remainder of this section, we will go deeper into the analysis of these trends and their influence on distance/similarity computation. We will first discuss the grouping tendency among the features as a general case. Then, we will map this discussion onto our work context and expand it towards local grouping trends among audio elements and their influence on unsupervised audio scene grouping.

5.2.1.1 Grouping Tendency at the Feature Level

Fig. 5.4 illustrates an example of a local feature grouping trend and its influence on a distance measure. Fig. 5.4(a) shows three data points *A*, *B*, and *C* located on the ***x***, ***y***, and ***z*** axes of a 3-dimensional feature space, respectively, and all at the same distance from the origin ***o***. If ***x***, ***y***, and ***z*** are independent of each other, then we have $||AB||=||BC||=||CA||$. Now, we assume that it has been known that in the data set there is a cluster whose data points lay on the ***x*** - ***y*** surface, as indicated by the region $\mathcal{D}$ in Fig. 5.4(b), where *A* and *B* belong to cluster $\mathcal{D}$ while *C* does not. Then, in order to properly reveal the cluster $\mathcal{D}$ after clustering, the distance from the point *C* to each of the points *A* or *B* should be larger than the distance between *A* and *B*, i.e., $||AB||<||BC||=||CA||$. In other words, we can say that the dimensions ***x*** and ***y*** are apt to be *locally grouped* or correlated given the distribution of data cluster $\mathcal{D}$. Such local ***x*** - ***y*** grouping should be considered in the distance measure to properly reveal the cluster $\mathcal{D}$. In the clustering practice, however, such local grouping trends among features cannot be analyzed in advance since data clusters like $\mathcal{D}$ are unknown. As a matter of fact, revealing such clusters is the purpose of clustering.

5.2.1.2 Grouping Tendency among Audio Elements

Recognizing and interpreting the grouping trends among audio elements can even be more intuitive than for the abstract case involving arbitrary features as discussed above. To show this, we consider the example illustrated in Fig. 5.5. This case involves 8 audio scenes, each of which is described by the occurrence probabilities of four audio elements. The matrix of occurrence probabilities is normalized to 1. In a one-way clustering approach, all four audio elements are considered independently in computing the similarity measure between the scenes, which results in four scene categories indicated by (a)-(d). However, a manual analysis of the content in the scenes suggests that the above grouping leads to over-segmentation of the content, and that grouping the scenes into only two categories would be more appropriate. The first of these two categories, indicated as *A*, can be labeled as *war*, and consists of scene groups (a) and (c). The second one, the category *B*, represents *humor* and includes scene groups (b) and (d). Here, we can say that the dimensions of *cheering* and *laughter* are locally grouped (or co-occur) given the samples from the *humor* scene. To discover these relations automatically, an algorithm should learn that a *gun-shot* usually occurs together with *explosion* in *war* scenes, while *cheering* and *laughter* often co-occur in *humor* scenes. More generally, since there are combinations of audio elements that often explain the semantics of audio data much better than the elements taken individually, we need to develop an audio scene clustering mechanism that can effectively discover and exploit the grouping tendency among audio elements.

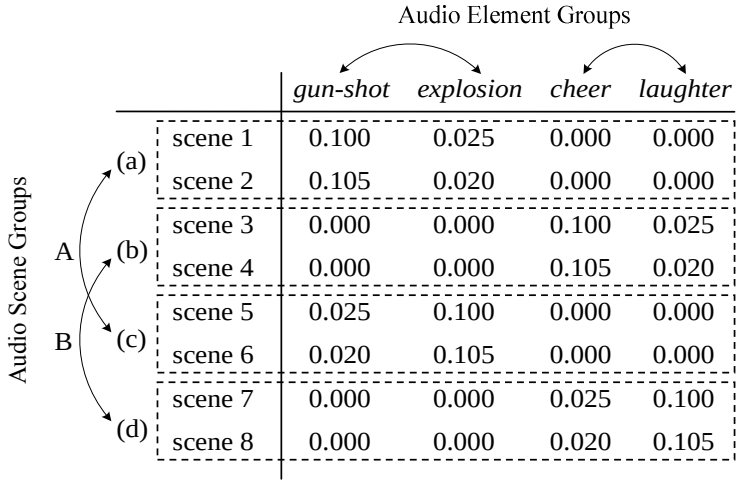


Fig. 5.5. An illustration of audio scene categorization based on audio elements

5.2.1.3 Discussion

Based on the analysis above, to achieve better clustering results, the local grouping trends among audio elements should be exploited to improve the audio scene similarity measurements. However, in order to reveal such local grouping trends from data, audio scene clusters should be known beforehand. In view of this, the grouping phenomena discussed above can optimally be exploited in the clustering process only by jointly pursuing the processes of clustering audio scenes and discovering the local grouping trends among audio elements. While this chicken-and-egg problem can be solved in a supervised learning context offline through an analysis of training data, finding a solution in an unsupervised context is difficult, especially if the traditional clustering mechanisms, like K-means, are deployed. Although one could think in the direction of some well-known statistical data pre-processing mechanisms, such as principal component analysis (PCA) [Shlens 2005] and independent component analysis (ICA) [Hyvarinen and Oja 2000] as possible means to approach a reasonable solution, we emphasize that PCA and ICA have an entirely different objective than the one defined in this chapter, and are not suitable for this purpose. They only search for global correlations among features and are used in general to reduce the dimensionality of the data set.

5.2.2 Co-clustering of Audio Scenes and Audio Elements

As the local grouping trends depend on the audio element co-occurrence in the samples from various audio scene clusters, an intuitive solution to reveal such trends is to cluster audio elements based on their co-occurrence first. The revealed relations among the elements are then used to measure similarities among scene samples and to form audio scene clusters. Based on the obtained initial scene clusters, the group relations among elements can be refined and then employed in a second round to provide better scene clusters. However, for the above iterative clustering process the convergence is difficult to prove theoretically [El-Yaniv and Souroujon 2001], and there is no plausible criterion, based on which such iterative process can be stopped.

In this thesis, we propose an alternative clustering approach based on the co-clustering idea. Co-clustering provides the possibility for a simultaneous clustering of audio elements and audio scenes, and it was already proved to converge toward a local minimum. Two different co-clustering approaches have been proposed in recent literature. One of them is based on spectral graph partitioning [Dhillon 2001], and the other one is an information-theoretic approach [Dhillon et al. 2003]. We choose to develop our co-clustering method based on the latter idea of *information-theoretic co-clustering* (ITCC), since it imposes fewer practical restrictions than the first approach[3]. While the cluster number in the ITCC approach are assumed to be known beforehand, in our approach we expand this approach by employing the Bayesian Information Criterion (BIC) [Kass and Wasserman 1995] to automatically select the optimal numbers of clusters for both audio scenes and audio elements.

5.2.2.1 Information-Theoretic Co-Clustering

The information-theoretic co-clustering [Dhillon et al. 2003] effectively exploits the relationships among various audio elements and audio scenes using the concept of *mutual information*. We now assume that there are m audio scenes to be clustered and that n audio elements are used to represent these scenes. Audio scenes can be considered as being generated by a discrete random variable $\boldsymbol{S}$, whose value is taken from the set $\{S_1, \ldots, S_m\}$. Similarly, audio elements can be assumed generated by another discrete random variable $\boldsymbol{E}$, whose value is taken from the set $\{e_1, \ldots, e_n\}$. Let $p(\boldsymbol{S}, \boldsymbol{E})$ denote the joint probability distribution between $\boldsymbol{S}$ and $\boldsymbol{E}$. As $\boldsymbol{S}$ and $\boldsymbol{E}$ are both

[3] In spectral graph partition-based co-clustering, the numbers of clusters in both feature dimensions and samples should be the same. Such assumption is too strict in the context of our work.

discrete, $p(\boldsymbol{S}, \boldsymbol{E})$ is an $m \times n$ matrix, whose elements can be represented by $p(S, e)$, which represents the co-occurrence probability of an audio element e and the audio scene S. Such a matrix is often called a two-dimensional *contingency table* or *co-occurrence table.* Fig. 5.5 shows an example of such a co-occurrence table. Now we also assume that $\boldsymbol{S}$ and $\boldsymbol{E}$ could be grouped into k and l disjoint clusters denoted as $\{S^*_1, \ldots, S^*_k\}$ and $\{e^*_1, \ldots, e^*_l\}$, respectively. These clusters could also be regarded as being generated by two discrete random variables $\boldsymbol{S}^*$ and $\boldsymbol{E}^*$.

We start our approach by measuring the amount of information shared between $\boldsymbol{S}$ and $\boldsymbol{E}$, that is, by computing the mutual information $I(\boldsymbol{S}; \boldsymbol{E})$:

$$I(\boldsymbol{S};\boldsymbol{E}) = \sum_S \sum_e p(S,e) \log_2 \frac{p(S,e)}{p(S)\,p(e)} \tag{5.6}$$

The mutual information is taken as a measurement of the original information of the data collection, controlled by latent relations between the variables $\boldsymbol{S}$ and $\boldsymbol{E}$. The co-clustering criterion states that the mutual information (5.6) should not change too much during the clustering, as the objective of the clustering is just to reveal the latent relations between the two variables. Based on such assumption, it was shown in [Dhillon et al 2003] that the optimal co-clustering method should target the minimization of the *loss of mutual information* after the clustering, i.e., for the optimal clusters we can write,

$$(\hat{\boldsymbol{S}}^*, \hat{\boldsymbol{E}}^*) = \min_{\boldsymbol{S}^*, \boldsymbol{E}^*} \{I(\boldsymbol{S};\boldsymbol{E}) - I(\boldsymbol{S}^*;\boldsymbol{E}^*)\} \tag{5.7}$$

The loss of mutual information can be represented as

$$I(\boldsymbol{S};\boldsymbol{E}) - I(\boldsymbol{S}^*;\boldsymbol{E}^*) = KL(p(\boldsymbol{S},\boldsymbol{E}), q(\boldsymbol{S},\boldsymbol{E})) \tag{5.8}$$

where $q(\boldsymbol{S},\boldsymbol{E})$ is also a distribution in the form of an $m \times n$ matrix, with each element defined as:

$$q(S,e) = p(S^*,e^*)\,p(S \mid S^*)\,p(e \mid e^*), \text{ where } S \in S^*, e \in e^* \tag{5.9}$$

and where $KL(f, g)$ denotes the *Kullback-Leibler* (*K-L*) divergence or *relative entropy* of two distributions $f(x)$ and $g(x)$:

$$KL(f,g) = \sum_x f(x) \log_2 \frac{f(x)}{g(x)} \tag{5.10}$$

As also shown in [Dhillon et al 2003], the *K-L* divergence in (5.8) can be further expressed in a symmetric manner:

$$KL(p,q) = \sum_{S^*} \sum_{S \in S^*} p(S) KL(p(\boldsymbol{E} \mid S), q(\boldsymbol{E} \mid S^*)) \tag{5.11}$$

$$KL(p,q) = \sum_{e^*} \sum_{e \in e^*} p(e) KL(p(\boldsymbol{S} \mid e), q(\boldsymbol{S} \mid e^*)) \tag{5.12}$$

From (5.11) and (5.12), we can see that the loss of mutual information can be minimized by minimizing the *K-L* divergence between $p(\boldsymbol{E} \mid S)$ and $q(\boldsymbol{E} \mid S^*)$, as well as the divergence between $p(\boldsymbol{S} \mid e)$ and $q(\boldsymbol{S} \mid e^*)$. This leads to the following iterative four-step co-clustering algorithm:

Algorithm Co_Clustering ($p(\boldsymbol{S}, \boldsymbol{E})$, k, l):

1. *Initialization*: Assign all audio scenes into k clusters, and audio elements into l clusters. Then compute the initial value of the q matrix.
2. *Updating audio scene clusters*: For each audio scene S, find its new cluster index i as:

   $$i = \arg\min_k KL(p(\boldsymbol{E} \mid S), q(\boldsymbol{E} \mid S_k^*)) \tag{5.13}$$

   Thus the *K-L* divergence of $p(\boldsymbol{E} \mid S)$ and $q(\boldsymbol{E} \mid S^*)$ decreases in this step. With new cluster indices of audio scenes, update the q matrix according to (5.10).
3. *Updating audio element clusters*: Based on the updated q matrix in step 2, find a new cluster index j for each audio element e as:

   $$j = \arg\min_l KL(p(\boldsymbol{S} \mid e), q(\boldsymbol{S} \mid e_l^*)) \tag{5.14}$$

   Thus the *K-L* divergence of $p(\boldsymbol{S} \mid e)$ and $q(\boldsymbol{S} \mid e^*)$ decreases in this step. With new cluster indices of audio elements, update the q matrix again.
4. *Re-calculating the loss of mutual information*. If the change in the loss of mutual information is smaller than a pre-defined threshold, stop the iteration process and return the clustering results; otherwise go to step 2 to start a new iteration.

[Dhillon et al. 2003] proved that the above iteration process results in a monotonic decrease in the loss of mutual information and always converges to a local minimum. In the implementation of the process, the "maximally-far-apart" criterion is used to

select the initial cluster centers, and the local search strategy is employed to increase the quality of the local optimum [Dhillon and Guan 2003]. The algorithm is computationally efficient and its complexity is $O(n \cdot \tau \cdot (k+l))$, where n is the number of non-zeros in $p(\boldsymbol{S}, \boldsymbol{E})$ and τ is the iteration number.

5.2.2.2 Estimating the Number of Clusters

In the co-clustering algorithm defined above, the numbers k and l of audio scene and audio element clusters, respectively, are assumed to be known. However, in an unsupervised approach, it is difficult to specify the cluster numbers beforehand. Thus, an effective approach to estimate these numbers automatically is required.

While the loss of mutual information is used as the criterion to evaluate the clustering results, it is also possible to use this criterion to choose optimal numbers of clusters. However, according to the definition, the loss of mutual information has its inherent variation trend with the change of cluster numbers, that is, more mutual information is reserved if more clusters are used. For example, when k and l are both one, there is 100% of mutual information loss. Compared to this, there is no mutual information loss if the cluster numbers are equal to the original numbers of samples. Therefore, we can not get reasonable numbers of clusters if we only rely on the loss of mutual information. However, from the viewpoint of statistics, clusters can be considered as a model describing the data distribution. Therefore, with more clusters, the model complexity (the number of parameters in the model) increases significantly. From this viewpoint, we can use the criteria like *Bayesian information criterion* (BIC) [Kass and Wasserman 1995] to select the optimal cluster numbers by balancing the loss of mutual information and model complexity. For instance, in K-means clustering [Pelleg and Moore 2000], the BIC trades off the data likelihood L with the model complexity $|\Theta|$. In practice, the former has a weighting factor λ, while the latter is modulated by the logarithm of the total number of samples T in the database. This leads to the BIC formulation as

$$BIC = \lambda L - \frac{1}{2} |\Theta| \log(T) \tag{5.15}$$

In our co-clustering scheme, the implementation of the BIC criterion is somewhat different from the one frequently used in one-way clustering. First, given the values of k and l, the data likelihood L could be approximated by the logarithm of the ratio between the mutual information $I(\boldsymbol{S}^*; \boldsymbol{E}^*)$ after clustering and the original mutual information $I(\boldsymbol{S}; \boldsymbol{E})$. It is assumed here that the model reserving more mutual

information would have a higher "probability" to fit the data. Second, as co-clustering is a two-way clustering, the model complexity here should consist of two parts: the size of audio scene clusters ($n \times k$: k cluster centers of dimensionality n), and the size of audio element clusters ($m \times l$: l cluster centers of dimensionality m). According to the definition of BIC, these two parts are further modulated by the logarithm of the numbers of audio scenes and audio elements, i.e. log$m$ and log$n$, respectively. This brings us the following definition of the BIC to be used in our co-clustering scheme:

$$BIC(k,l) = \lambda \log \frac{I(\boldsymbol{S}^*;\boldsymbol{E}^*)}{I(\boldsymbol{S};\boldsymbol{E})} - (\frac{nk}{2}\log m + \frac{ml}{2}\log n) \tag{5.16}$$

In the implementation, λ is set experimentally as $m \times n$, which is the size of the co-occurrence matrix. The algorithm searches over all (k, l) pairs in a pre-defined range, and the model with the highest BIC score is chosen as the optimal clustering result.

5.2.2.3 Construction of the Co-occurrence Matrix

To apply co-clustering on our obtained audio scenes, we first need to construct the co-occurrence matrix (or contingency table) linking the scene set and audio elements set. While previous related approaches to audio clustering mainly rely on key audio elements to infer the semantics of audio scenes, we follow the approach we already introduced for audio scene segmentation and use all audio elements to reveal the natural audio cluster structure. Further, what we know about the input audio track is the presence and duration of discovered audio elements per each detected audio scene. Therefore, the occurrence probability of the audio element e_j in the audio scene S_i can simply be approximated by the duration percentage $occr_{ij}$ of e_j in S_i. If an audio element does not occur in the scene, its duration percentage is set to zero. Finally, to satisfy the requirement that the sum of the co-occurrence distribution is equal to one, the co-occurrence matrix $p(\boldsymbol{S}, \boldsymbol{E})$ is normalized as:

$$p(S_i, e_j) = occr_{ij} / \sum_{i=1}^{m}\sum_{j=1}^{n} occr_{ij} \tag{5.17}$$

5.3 Experimental Evaluation

In this section, the performance of the proposed approach to audio scene segmentation and clustering is evaluated based on the data collection containing 5 hours of sound tracks listed in Table 4.1 (Chapter 4) and the corresponding audio elements discovered therein.

5.3.1 Audio Scene Segmentation

For evaluating the performance of audio scene segmentation, we first created the set of ground truth audio scene boundaries. For this task, we employed a number of test persons who are instructed on how to understand the concept of an audio scene. For the situation comedy (A_1) and movies (A_4 and A_5), we linked the audio scenes to the concept of video scenes, for which we adopted the definition of an episode or a logical story unit [Hanjalic et al. 1999], that is, a meaningful segment related to a particular action or event, location or time. As other two test sequences in our data set (award ceremony and tennis) show the events taking place on one location (i.e. the show stage or the tennis court, respectively) and do not follow a typical episode-based structure as sitcoms or movies, a slightly different understanding of an audio scene needed to be adopted there. For instance, in the award ceremony (A_3) we targeted the segments like those where the host announces the nominees and the winner, and where the winner approaches the stage while the audience is applauding. Similar event-based scene concept was targeted for the Tennis sequence (A_2), where, for instance, an event starting with a serve and ending by the score change can be considered a scene potentially interesting for retrieval, as well as the scenes of a break characterized by the speech of the anchorperson commenting the match.

As the instructions given to the annotators were not strict, but only meant to help them to approach the annotation problem at the right abstraction level, different audio scene sets could be expected from different annotators. While some of the audio scenes were obvious and were detected by all annotators, a number of boundaries were proposed only by some of them. We refer in this paper to these two sets of ground-truth audio scene boundaries as the *true* and *probable* ones, respectively. Probable boundaries appeared mainly at parts of our data set where the semantic content flow can be followed at different abstraction levels. For example, in the award ceremony, the turns between the played excerpts of nominated movies were often marked as additional audio scene boundaries. In total, we obtained 295 true boundaries and 186 probable boundaries from five sound tracks.

In the following, we present and discuss the results of two experiments that we performed to evaluate the performance of our audio scene segmentation approach. In the Experiment 1, we compare the segmentation performance of our approach based on audio elements with the performance of typical feature-based approaches. To make the results and the related discussions more complete, we experiment with several variants of our approach, in which we investigate the impact of different design choices that we introduced in Section 5.1.3, such as the choice of importance weights and the assumptions underlying our definition of semantic affinity (5.1). While in this

experiment we worked with the fixed length of the buffers L-Buf and R-Buf (see Fig. 5.2), in the Experiment 2, we investigate the influence of this length on the segmentation performance and justify the buffer length used in the first experiment.

5.2.1.1 Experiment 1: A Comparative Analysis

The feature-based approach implemented as a reference for the comparison follows the general idea illustrated in Fig. 5.1(a), where the segmentation is simply done by investigating the feature consistency within a sliding window. In the first variant of the feature-based approach, we simply use the mean of the feature vectors to represent the segments on the left and right from the candidate boundary, and then the feature consistency is measured by a cosine distance:

$$C(t) = \frac{f_l \cdot f_r}{\| f_l \| \cdot \| f_r \|} \tag{5.18}$$

where f_l and f_r are the average feature vectors on the left and right from the candidate boundary.

The other variant follows the approach proposed in [Ellis and Lee 2004], which uses the Bayesian information criterion (BIC) [Kass and Wasserman 1995] to evaluate the feature coherence across the candidate boundary, that is,

$$C(t) = BIC(t) = \log(\frac{L(s \mid M)}{L(s_l \mid M_l) l(s_r \mid M_r)}) - \frac{\lambda}{2} \| M \| \log(N_l + N_r) \tag{5.19}$$

where s represents the entire set of elementary audio segments within the sliding window, and s_l and s_r are the segments on the left and right from the candidate boundary. Furthermore, $L(s|M)$, $L(s_l|M_l)$ and $L(s_r|M_r)$ are the likelihoods of the data sets s, s_l and s_r under the corresponding models M, M_l and M_r, respectively, which are defined as Gaussian models in our implementation. Finally, ||M|| refers to the number of model parameters. As explained in [Ellis and Lee 2004], λ is a tuning constant that can be used to regulate the (over-) segmentation tendency of the method.

The two variants of the feature-based approach described above will be compared with four variants of our proposed approach. The first two variants use the same general formula for semantic affinity (5.1) and the content coherence (5.3), but work with a different or more limited amount of information about audio elements and their individual or joint behavior to compute the semantic affinity. The first variant (*Var1*) does not use the importance weights of audio elements nor the information on their

Table 5.1. A summary of six approaches (variants) in the experiments

ID	Approach (variant)
m1	audio element based, with rule-based weights, eq.(5.1)
m2	audio element based, with TFIDF-based weights, eq.(5.1)
m3	audio element based (*Var 2*), without weights, , eq.(5.21)
m4	audio element based (*Var 1*), without co-occurrence, eq.(5.20)
m5	feature-based, with BIC approach, eq. (5.19)
m6	feature-based, with COS similarity, eq. (5.18)

co-occurrence, but relies on their features to measure the similarity between two segments. In fact, this variant of our approach differs from the traditional feature-based approach only in that it searches for audio scene boundaries between audio element blocks (see Fig. 5.2) and not on a continuous time scale. Therefore, it can be also considered as another feature-based method. In this case the semantic affinity becomes,

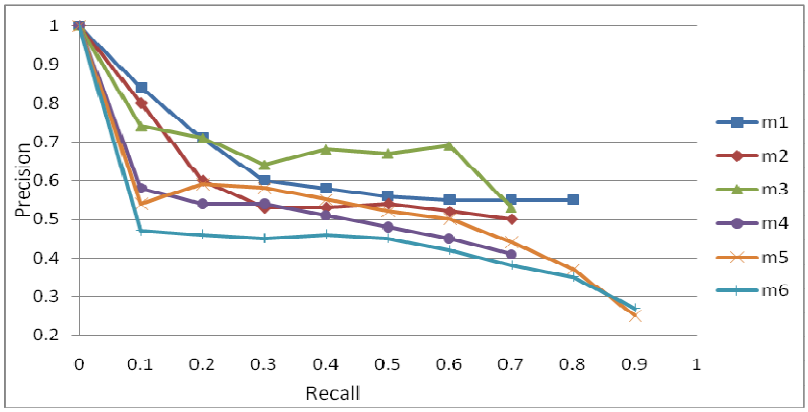
$$A'(s_i, s_j) = Sim(s_i, s_j)e^{-T(s_i,s_j)/T_m} \tag{5.20}$$

where the feature-based similarity $Sim(s_i, s_j)$ is also computed using (5.18). To explicitly evaluate the influence of the importance weights of audio elements on the parsing performance, we also define a second variant (*Var2*) of our method that relies on co-occurrence between audio elements but does not take into account the weighting of each audio element. In this case, the semantic affinity becomes

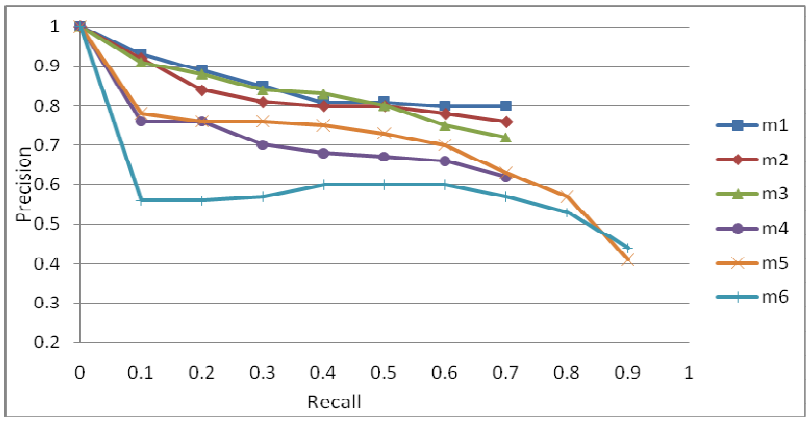
$$A''(s_i, s_j) = Co(e_i, e_j)e^{-T(s_i,s_j)/T_m} \tag{5.21}$$

Further two variants of our proposed approach both use the semantic affinity as defined in (5.1), but differ in the way the importance weights are computed (i.e. rule-based vs. TFIDF-based). The six segmentation methods described above are summarized in Table 5.1. We will further refer to *m1-m3* as the audio-element based methods, and to *m4-m6* as the feature-based methods.

In methods *m*1-*m*4, the value of the buffers L-Buf and R-Buf is set to 16 seconds, based on the results of the Experiment 2. In order to perform a fair comparison with methods *m*5 and *m*6, we set the length of the sliding window there to 32 seconds, which is equivalent to having the buffers L-Buf and R-Buf of 16 seconds surrounding the control point (candidate boundary) positioned in the middle of the window.



(a)



(b)

Fig. 5.5. Precision-Recall curve obtained for all six methods and for (a) true, and (b) both true and probable boundaries considered as ground-truth boundaries

In order to investigate the segmentation performance with respect to both true and probable boundaries, we define two experimental setups, a strict and a loose one. The strict setup considers only the true boundaries as ground-truth boundaries, and the probable ones – if detected – are considered as false alarms. In the loose setup, we treat both the true and probable boundaries as ground-truth boundaries. Moreover, in our approach, the confidence curve obtained by (5.4) is first linearly normalized to [0,1] based on the maximum and minimum value in the curve, before employing (5.5) for boundary detection. In this way, the curves from different audio tracks are normalized into the same value scale, so that a consistent *Th* can be used across different audio

tracks. Finally, a detected boundary is associated with an annotated boundary if they are mutually closest to each other, based on which the recall and precision of boundary detection are calculated.

Fig. 5.5(a-b) shows the precision and recall curves obtained for the six methods from Table 5.1, obtained by varying threshold *Th* in (5.5) (in the range [0,1]), and for two different setups described above. Based on these results, in addition to the expected structural increase in precision if both the true and probable boundaries are considered, the most significant result supporting the rationale behind the approach presented in this chapter is that the segmentation based on audio elements performs better than the one using traditional feature-based approaches. This difference in performance becomes visible if one compares the curves cluster *m1*, *m2* and *m3* with the cluster *m4*, *m5* and *m6*. Especially in the relevant parts of the precision-recall curves, that is, those with sufficiently high recall values (e.g. above 0.6), the dominance of the audio-element based methods becomes clearly visible.

Regarding the feature-based methods (*m4-m6)*, a generally lower performance in terms of precision can be traced back to the discussion in Section 5.1.1, in which we addressed the incapability of these methods to capture the entire content diversity of a high-level semantic concept. This results in audio scenes that are typically short, at the level of basic audio modalities (e.g. speech, music, noise) and of no higher (semantic) meaning. If we compare the performance of the methods *m4*, *m5* and *m6*, we observe that the more sophisticated method *m5* indeed performs structurally better than a simpler method *m6*. Further, compared to the methods *m5* and *m6* that check for the presence of an audio scene boundary between every two audio segments, the method *m4* focuses on the boundaries between audio elements blocks (Fig. 5.2) as the candidates for an audio scene boundary. This focus may negatively influence the segmentation performance in terms of recall, which is partly visible from the comparison with method *m5*. However, this focus may also help de-noise the set of audio scene boundaries, which is visible in particular when the simple method *m6* is taken as a reference.

To investigate the impact of various design choices on the segmentation performance, we focus in the analysis of the audio-element based methods *m1*, *m2* and *m3* on each of these choices separately. Regarding the audio element weighting mechanism (i.e. rule-based vs. TFIDF-based), a comparison of the performance of methods *m1* and *m2* shows that both weighting mechanisms lead to a rather similar performance, with a slight dominance of the rule-based weights in higher recall value ranges. The positive impact of audio element co-occurrence on the segmentation performance is clearly visible from the comparison of methods *m3* and *m4*. The same conclusion, although not as obvious as in the case of co-occurrence, can be drawn regarding the impact of the importance weights. It is namely interesting to observe that method *m3*, which does not

use weights, works better than methods *m1* and *m2* at some parts, especially at lower recall value ranges (Fig. 5.5(a)). However, for the practically relevant recall values (e.g. above 0.6), a better performance of *m1* and *m2* becomes visible. The different behavior of the precision-recall curve of the method *m3* in Fig. 5.5(a) and Fig. 5.5(b) relative to methods *m1* and *m2* can be explained by the treatment of probable boundaries in each of these two cases. We observed namely that more probable boundaries are detected by *m1* and *m2* than by *m3*, which can lead to the conclusion that the probable boundaries identified by our annotators are generally surrounded by relatively high-weighted (background) sounds. This conclusion is intuitive, as the changes in the audio content in terms of characteristic sounds are what move the annotator to recognize a scene break in the first place. As in the setup in Fig. 5.5(a) the probable boundaries are treated as false ones, the precision of *m1* and *m2* reduces compared to *m3*. This changes, however, in favor of the methods *m1* and *m2* if probable boundaries are also considered as the true ones (Fig. 5.5(b)). By further increasing the threshold *Th* to obtain higher recall values, the sophistication of the semantic affinity models becomes more and more important in order to avoid spurious valleys in the confidence curve (5.4). Therefore, as expected, the missing de-noising effect of the importance weights is likely to have negative influence on the relative precision of method *m3* in higher recall value ranges. This is indicated by the precision drop of the method *m3* both in Fig. 5.5(a) and Fig. 5.5(b).

We also observed that the audio element based approaches seldom achieves recall values higher than 0.7. This might be an indication of the changes of audio signal characteristics across those undiscovered boundaries being too small. An argument supporting a conclusion in this direction could be that the feature-based approach results in a rapid drop in the precision when the recall achieves more than 0.8. Specifically, for the methods *m1*, *m2*, *m3* and *m4*, we found that the upper limit for recall values for these methods is influenced by the length of the buffers L-buf and R-buf, over which the pair-wise semantic affinity values are averaged to compute the confidence value (5.4). We discuss this in the Experiment 2.

5.2.1.2 Experiment 2: Investigating the Effect of Buffer Length

In order to investigate the effect of the length of the buffers L-Buf and R-Buf on the segmentation performance, we designed an experiment, in which we use method *m1* as the reference, where we consider both true and probable boundaries as ground-truth boundaries, and in which we vary the buffer length over the values 4s, 8s, 12s, 16s, 24s and 32s. For each of the buffer values, we compute the precision-recall curve. The results of this experiment are shown in Fig. 5.6.

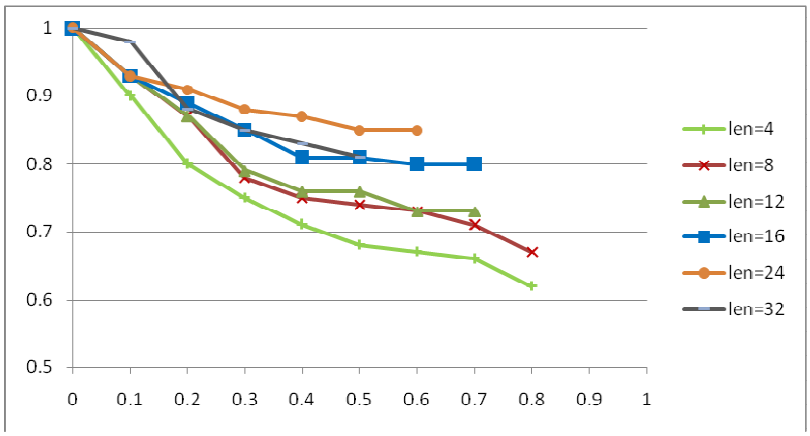


Fig. 5.6. Precision-Recall curves obtained for method *m1*, for different lengths of the buffers R-Buf and L-Buf, and considering both true and probable boundaries as ground-truth boundaries

Our first observation is that increasing the buffer length in general has a positive impact on the segmentation performance (from 4s to 24s), and this in particular for higher recall values. Our second observation is that, as already stated before, different maximum recall values appear for different precision-recall curves. Even more, the reduction in this upper limit directly follows the increase in the buffer length: the maximum reachable recall of around 0.8 is possible for the shortest buffers only, while the longest buffer length of 32s leads to the smallest reachable recall of around 0.5. Our search for the buffer length that is likely to lead to the best segmentation performance, but also provide the maximum possible recall value range led to the buffer of 16s that we adopted in the Experiment 1.

5.3.2 Audio Scene Clustering

As the original scheme for content-based audio analysis introduced in Chapter 2 implies that the co-clustering method will take automatically obtained audio scenes as input, we choose here to evaluate the actual clustering performance based on the best input we could obtain from the automated segmentation step. As indicated in Fig. 5.5, this is the case for the segmentation method using audio elements and rule-based importance weights. We again employed three persons to manually group the obtained scenes into a number of semantic categories. Based on this manual grouping, we established the ground truth for further evaluation.

To demonstrate the effectiveness of the co-clustering approach, we first compare it with a traditional one-way clustering algorithm. Here, the *X*-means algorithm [Pelleg and Moore 2000], in which the BIC is also used to estimate the number of clusters, is adopted for the comparison. Assuming that the maximum possible number of audio scene clusters is 10, we search in both clustering options for the proper number of scene clusters in the range of 1 to 10. In addition, for the co-clustering option, we search for the optimal number of audio element groups in the range of 1 to *n*, where *n* is the number of audio elements in the corresponding sound track.

Table 5.2. Detailed results of the comparison between the *X*-means clustering and our proposed co-clustering method applied to audio scenes of the sound track "Friends" (A_1)

Co-clustering	**No.**	**S_1**	**S_2**	**prec.**	**X-means**	**No.**	**S_1**	**S_2**	**prec.**
	1	34		1.00		1	7		1.00
	2		2	1.00		2	3		
	recall	1.00	1.00	**1.00**		3	3		
						4	12		
						5	9		
						6		2	1.00
						recall	1.00	1.00	**1.00**

Note: (S_1) scenes of dialog; **(S_2)** scenes of music

Table 5.3. Detailed results of the comparison between the *X*-means clustering and our proposed co-clustering method applied to audio scenes of the sound track "Tennis" (A_2)

Co-clustering	**No.**	**S_1**	**S_2**	**S_3**	**S_0**	**prec.**	**X-means**	**No.**	**S_1**	**S_2**	**S_3**	**S_0**	**prec.**
	1	1	17	4	1	0.74		1	2	12	10	1	0.48
	2	9				1.00		2	4				1.00
	3	1	2	14		0.85		3	1	1	8		0.70
	4		1	21				4		7	18		
	5	2	2	11				5	1		8		
	recall	0.69	0.77	0.92	0.00	**0.84**		6	5	2	6		
								recall	0.31	0.55	0.80	0.00	**0.65**

Note: (S_1) scenes of anchorperson talking during the break; **(S_2)** scenes of anchorperson talking and with audience applauding (with little ball-hits or in long play break); **(S_3)** scenes of anchorperson talking and players playing; **(S_0)** others

Table 5.4. Detailed results of the comparison between the *X*-means clustering and our proposed co-clustering method applied to audio scenes of the sound track "59th Annual Golden Globe Awards" (A_3)

Co-clustering	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	**S_0**	**prec.**
	1	19	4				0.83
	2	4	10				0.71
	3			9	2	3	0.64
	4	7	1			14	0.63
	5			1	2	5	
	6				38	4	0.90
	recall	0.63	0.67	0.90	0.90	0.73	0.77

X-means	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	**S_0**	**prec.**
	1	5				5	0.58
	2	12	8				
	3	5	3				
	4	4	4			2	0.40
	5			8	1	1	0.80
	6	4				8	0.60
	7			1	3	4	
	8				12	1	0.84
	9				16	3	
	10			1	10	2	
	recall	0.73	0.27	0.80	0.86	0.46	0.68

Note: (S_1) scenes of hosts or winners coming to or leaving the stage; **(S_2)** scenes of audience congratulating and applauding to the winners; **(S_3)** scenes of hosts announcing nominees or winner candidates; **(S_4)** scenes of winners or hosts' speech; **(S_0)** others

Table 5.5. Detailed results of the comparison between the *X*-means clustering and our proposed co-clustering method applied to audio scenes of the sound track "Band of Brother" (A_4)

Co-clustering	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	prec.
	1	8	0	0	0	1.00
	2	10	0	0	0	
	3	0	0	8	3	0.73
	4	0	11	1	0	0.94
	5	1	11	0	0	
	6	0	9	0	0	
	recall	0.95	1.00	0.89	0.00	0.92

X-means	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	**prec.**
	1	10	0	0	0	1.00
	2	3	0	0	0	
	3	4	0	0	0	
	4	0	0	7	3	0.70
	5	1	2	0	0	0.89
	6	0	9	1	0	
	7	1	9	0	0	
	8	0	10	1	0	
	9	0	1	0	0	
	recall	0.89	1.00	0.78	0.00	0.89

Note: (S_1) scenes of battle; **(S_2)** scenes of dialog in noisy background; **(S_3)** scenes of dialog in music background; **(S_4)** scenes of music

Table 5.6. Detailed results of the comparison between the *X*-means clustering and our proposed co-clustering method applied to audio scenes of the sound track "Sword Fish" (A_5)

Co-clustering	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	**S_0**	**prec.**
	1	3					1.00
	2		18	5	2	2	
	3		44				0.87
	4		35	1		4	
	5			6			1.00
	6			1	13	2	0.81
	7	1		1	4		0.67
	recall	0.75	1.00	0.43	0.89	0.00	0.87

X-means	**No.**	**S_1**	**S_2**	**S_3**	**S_4**	**S_0**	**prec.**
	1		7			2	
	2		18				
	3		11	2			
	4		18				0.83
	5		11	2	1	3	
	6		17	5		1	
	7		15		2	2	
	8	2		3			0.60
	9	1		1	5		0.80
	10	1		1	11		
	recall	0.00	1.00	0.21	0.84	0.00	0.82

Note: (S_1) movie end-scenes with music theme; **(S_2)** scenes of dialogs with music or other sounds in the background; **(S_3)** scenes of dialogs with strong music in the background; **(S_4)** scenes of actions, usually with strong music in the background; **(S_0)** others

Table 5.7. Performance comparison between the X-means and the Co-clustering on all audio tracks, with automatically obtained audio scenes and the audio elements found therein

No.	# Labeled Semantic Group	X-means		Co-clustering	
		# Group	Accuracy	# Group	Accuracy
A_1	2	6	1.00	2	1.00
A_2	4	6	0.65	5	0.84
A_3	5	10	0.68	6	0.77
A_4	4	9	0.89	6	0.92
A_5	5	10	0.82	7	0.87
Avr.	*4*	*8.2*	*0.80*	*5.2*	*0.88*

Table 5.2-5.6 shows the detailed comparison results of the two clustering algorithms on the five test sound tracks. Taking the sound track of the "59th Annual Golden Globe Awards" (A_3) and Table 5.4 as an example, we can observe that 115 scenes were detected, which are manually classified into 5 semantic categories: 1) scenes of hosts or winners coming to or leaving the stage (S_1), which are mainly composed of *applause*

and *music*, 2) scenes of audience congratulating and applauding to the winners (S_2), which are mainly composed of *music*, *applause* and *cheering*, 3) scenes of hosts announcing nominees or winner candidates (S_3), which are mainly composed of *applause* and *speech*, 4) scenes of winners' or hosts' speeches (S_4), which are mainly composed of *speech*, and 5) others which are hard to assign to any of the above four scenes (S_0). Our experiments resulted in 6 categories of audio scenes when using the information-theoretic co-clustering, and 10 scene categories if *X*-means is used. In Table 5.4, each row represents one obtained cluster and the distribution of the audio scenes contained therein across the ground truth categories. As indicated by the shaded fields, we assign an obtained cluster to a ground truth cluster if the corresponding ground truth scenes form the majority in this cluster. In case there are multiple obtained clusters that get associated to the same ground truth cluster, we manually group these clusters and then compute the precision and recall per cluster group. The obtained results show that the co-clustering algorithm is likely to perform better than a one-way clustering.

Table 5.7 summarizes the comparison results obtained from all the sound tracks in our test data set. The results in this table confirm the conclusion we draw based on the example in Table 5.4. The number of audio categories obtained by co-clustering is closer to the number of ground truth categories than in the case of one-way clustering. In other words, co-clustering can provide a more exact approximation of the natural cluster structure present in the data. For example, there are in average 4 semantic groups per sequence in the test data set. The co-clustering approach obtains 5.2 groups in average, while *X*-means obtains 8.2 groups. Furthermore, co-clustering leads to a higher precision and recall. In average, around 88% of the scenes are correctly clustered with the co-clustering algorithm, while the accuracy of the *X*-means is 80%.

In addition to the comparison of different clustering techniques, we also compared different ways of representing audio scenes (i.e. features vs. audio elements) when deploying our co-clustering algorithm. In order to implement feature-based co-clustering, we need to form a co-occurrence matrix $p(S_i, f_j)$ which is slightly different from the one defined in Section 5.2.2.3. Since acoustic features usually have varying value dynamics, they are first normalized prior to the construction of the co-occurrence matrix. That is, for *j*-th feature, its value in the *i*-th audio scene is re-scaled according to the following expression:

$$\hat{f}_j^i = (f_j^i - f_j^{\min}) / (f_j^{\max} - f_j^{\min}) \tag{5.22}$$

where $f_j^{\max} = \max(f_j^i, 1 \le i \le m)$ and $f_j^{\min} = \min(f_j^i, \quad 1 \le i \le m)$.

Table 5.8. Performance of co-clustering on all audio tracks, based on automatically obtained audio scenes and their corresponding features

No.	# Labeled Semantic Group	Co-clustering (automatic, audio scenes and features)	
		# Group	Accuracy
A_1	2	5	0.97
A_2	4	5	0.63
A_3	5	6	0.62
A_4	4	4	0.77
A_5	5	6	0.80
Avr.	*4*	*5.2*	*0.76*

With this procedure, the values of each feature in all scene samples are brought into the range of [0, 1], and can now be used to approximate the occurrence probability of the related feature in a given audio scene. As an example of the relation searched here, the larger the value of the "short-time energy," the higher is the probability of "high volume" in a given audio scene. In the final co-occurrence matrix, each element of $p(S_i, f_j)$ is further normalized to ensure the sum of the co-occurrence distribution is one, that is

$$p(S_i, f_j) = \hat{f}_j^i / \sum_{i=1}^{m} \sum_{j=1}^{n} \hat{f}_j^i \tag{5.23}$$

Table 5.8 summarizes the clustering results while using features to represent each audio scene. Compared to Table 5.7, using the features directly leads to a 12% decrease in average accuracy. Similarly to the results obtained for audio scene segmentation, this again confirms that using audio elements as mid-level representation improves the performance of high-level semantic inference.

Furthermore, we make a comparison between co-clustering using automatically and manually segmented scenes. Table 5.9 shows the obtained results. While the clustering based on manual segmentation performs – as expected - slightly better, the clustering based on automatically segmented audio scenes still results in acceptable performance figures. Implicitly, these results also provide an additional indication of a good performance of our automatic audio scene segmentation method.

Finally, as shown in Table 5.10, our (audio element based) co-clustering algorithm also suggests several audio element groups for each sound track. These groups realistically reveal the grouping (co-occurrence) tendency among the audio elements, as explained in Section 5.2.1. For example, in the "59th Annual Golden Globe Awards"

Table 5.9. Performance of co-clustering on all audio tracks, based on manually segmented audio scenes and the audio elements found therein

No.	# Labeled Semantic Group	Co-clustering (manual segmentation and audio elements)	
		# Group	Accuracy
A_1	2	2	1.00
A_2	4	5	0.91
A_3	5	5	0.83
A_4	4	5	0.92
A_5	5	4	0.89
Avr.	*4*	*4.2*	*0.91*

ceremony (A_3), we observed that the sounds of *applause with music* and *applause with dense-music* usually occur together in the scenes of "the hosts or winners coming to or leaving the stage", and they are also correctly grouped together using the co-clustering algorithm. This audio element grouping process can also help compensate for possible over-segmentation problem during audio element discovery, as mentioned in Chapter 4. If we again take the sound track of "59th Annual Golden Globe Awards" ceremony (A_3) as example, although the audio element detection process has spread the occurrences of the term "*speech*" over several audio elements indicated as *speech1*, *speech2*, and *speech3*, these elements were grouped together again using co-clustering.

5.3.3 Discussion

Extensive experimental evaluation reported in previous sections confirmed the superiority of the two-step audio semantic inference approach we adopted in this thesis. The way we deployed the joint behavior of audio elements in the inference process at the audio scene level led to a considerable improvement of the segmentation and clustering performance, compared to the approaches relying on the features directly. Next to the dedicated experiments designed to evaluate the performance of the audio scene segmentation algorithm, an additional indication of the algorithm quality was obtained through the evaluation of the co-clustering algorithm. The co-clustering performance namely decreased only for 3% in the case when automatically detected audio scenes served as input and compared to the case where manually segmented scenes were adopted. The proposed co-clustering algorithm proved to be superior to the classical one-way clustering approach, which emphasizes the importance of exploiting

Table 5.10. The audio element groups obtained using co-clustering

No.	#G	Audio Element Groups
A_1	3	{speech + noise}; {laughter, laughter + music}; {TV music, theme music, speech, applause + cheering}
A_2	3	{clean speech, noisy silence}; { speech + applause, music}; {applause, silence, ball-hit}
A_3	5	{speech1, speech2, speech3 }; {background noise}; {applause, speech + applause }; {music + applause 1, music + applause 2, (dense) music + applause }; {music + speech, music };
A_4	5	{speech, silence (some noise), background sounds, silence (with HF noise)}; {speech (gunshot background), speech, heavy noise, gunshot + speech 1, gunshot + speech 2}; {speech, noise + speech, speech, applause }; {music, music + speech }; {noise}
A_5	5	{speech, speech, speech + backgrounds, speech + backgrounds}; {speech + backgrounds, speech + backgrounds, backgrounds, backgrounds}; {fighting sounds 1, music, music}; {mixed backgrounds, music}; {fighting sounds 2, speech + backgrounds, speech in repressive env, fighting sounds}

local grouping tendencies of audio elements in the process of audio scene grouping. Finally, an additional value was created by the co-clustering algorithm regarding the problem of handling over-segmentation in the audio element detection process. The co-clustering algorithm namely not only grouped the audio scenes in meaningful clusters but also suggested groups of audio elements, which belong together in terms of their meaning but were separated due to the variations in audio signal properties.

The way the audio scene clustering was evaluated may be arguable, since we allowed multiple (obtained) clusters to get associated with one ground truth cluster for precision and recall computation (Table 5.2-5.6). Another possibility to perform this evaluation could be to associate only the most relevant obtained cluster (the one with the largest overlap) to each ground truth cluster. Actually, this evaluation strategy would even further emphasize the benefits of our method, since the number of scene groups obtained in the X-means based approach usually tends to be larger (thus more false alarms are introduced).

Chapter 6

Towards a Broader Perspective

In this chapter, we first revisit the original goal of this thesis and the approach we proposed to reach this goal. Then, we make an attempt to envision a possible expansion of the proposed approach towards an application scope broader than the one considered in this thesis.

6.1 Thesis Goal Revisited

As discussed in Chapter 1, the goal of the research reported in this thesis is to build a generic and flexible framework for content discovery from composite audio. Towards this goal, and in view of the discussion about the limitations of the related previous work on the subject in Chapter 2, we propose a novel approach to unsupervised semantic inference from composite audio that is based on the following main design choices:

Unsupervised Mining: In order to maximize the generic applicability of our envisioned content-based audio analysis solution, we choose for an unsupervised approach. The design of this approach was inspired by unsupervised text document analysis, recent works on video scene segmentation, and the idea of co-clustering.

Parts of this chapter are based on the following publication (also to be found in the list of references):

- Lu, L., and Hanjalic, A. "Unsupervised Anchor Space Generation for Similarity Measure of General Audio," *Proc. 33th Int'l Conf. on Acoustics, Speech, and Signal Processing*, 53-56, 2008

Two–step knowledge discovery: Introducing mid-level content descriptors and considering them in the content analysis process enables us to split the semantic inference process into two steps, which proved to lead to more robustness, compared to the case where semantics is inferred from the features directly.

Although the proposed approach obtained promising results on a representative test data set, it still leaves room for further investigation and improvement. While the possibilities for this improvement were discussed at various instances in the previous chapters of this thesis, we dedicate this chapter to reflect upon the main design aspects of our approach with the objective to identify possibilities for expanding its applicability scope.

Unsupervised mining has the advantage that it requires neither manual annotation of semantic categories nor offline collection of the training data. In this sense, it is likely to have a wide application scope and to be suitable particularly in those application scenarios where obtaining manual annotations and large training data sets is difficult. However, what about the situations in which reliable prior knowledge at various levels is available to help the semantic inference? Is the proposed approach flexible enough to accommodate and optimally exploit such knowledge? In other words, what are the possibilities to enhance the proposed unsupervised (generic) approach with the knowledge generated through supervised processes to improve semantic inference in specific domains?

In our approach, audio elements are extracted from a given audio document and used as mid-level semantic descriptors to infer higher-level semantic concepts in that document, like audio scenes, and to group them into semantically meaningful clusters. However, what about the case where audio content similarity needs to be computed at even a higher abstraction level, namely across many different audio documents, for instance for the purpose of management and retrieval of large audio document collections? How to obtain an effective audio document representation that would enable us to compare and group together large audio documents in the same way as we group audio scenes? Can we simply apply the same methodology as introduced in Chapter 4, or are there adjustments required?

In view of the questions posed above, we now present our views on the possibilities to expand the proposed approach in order to enable general audio search and management applications. We will search for such possibilities by focusing 1) on combining the unsupervised and supervised approaches, and 2) on expanding the concept of document-specific audio elements to an anchor space representing a large collection of long audio documents.

6.2 On Combining a Supervised and Unsupervised Approach

As discussed in the previous section, an unsupervised content discovery approach can be enhanced by prior knowledge available in a given application scenario. On the other hand, the results obtained using unsupervised content discovery can also benefit a supervised classification-based approach. In this section, we will discuss how to interchangeably use supervised and unsupervised content discovery components in different scenarios to maximize their mutual benefit.

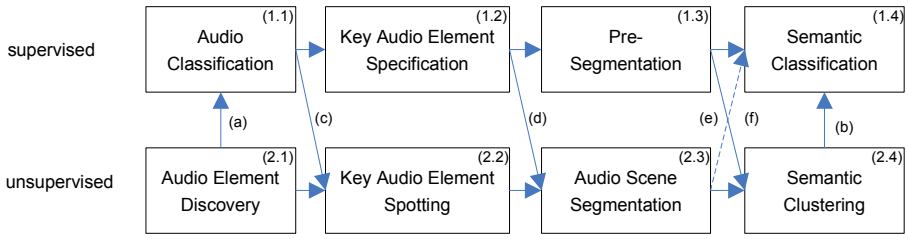


Fig. 6.1 Collaboration between supervised and unsupervised components in a combined audio content discovery approach, where some interesting directions for the transfer of the results from one component to another one are indicated by the arrows

Fig. 6.1 illustrates an example of a combined audio content discovery approach integrating the components of an unsupervised content discovery approach and its supervised counterpart. While supervised semantic inference can be realized in different ways, the term “supervised counterpart” refers here to a supervised inference process that best resembles our unsupervised approach with respect to the processing steps involved. The components of the supervised branch are realized using classification techniques or manual effort, including audio classification (e.g. classification of audio segments into audio elements), key audio element specification (e.g. learning the weights of audio elements offline by analyzing a training data set, or specifying the weights manually), pre-segmentation (e.g. training a boundary model offline, or manually dividing an audio document into audio scenes), and semantic classification (e.g. assigning audio scenes to trained semantic categories).

Each component in the scheme in Fig. 6.1 is numbered to facilitate later reference. Here, the notation (*j*. *k*) stands for the component *k* in the approach *j* (1 for supervised approach and 2 for unsupervised approach). The directed connections (also numbered) represent some interesting possibilities for a propagation of the results between two

components. For example, (1.1)→(2.2) means that the results of component (1.1) can be used for component (2.2). The marked connections between the components of the unsupervised and supervised approach can lead to a number of interesting combinations of the supervised and unsupervised approach. We will elaborate on some of these combinations in the next three sections.

6.2.1 Using Clustering to Enhance Classification

We first consider the case where (unsupervised) clustering results can be used to enhance supervised categorization. This case is represented by the arrows (a) and (b). After grouping audio segments into clusters, each cluster can be considered as a sample to be assigned to a class label in a supervised approach. Compared to the typical classification practice assigning labels to individual audio segments, this cluster-based approach is likely to improve the classification efficiency, but also the accuracy due to a "denoising" effect of the clustering process: e.g. majority vote can be applied to classify a set of segments to correct the "noisy" results obtained per segment. A potential problem here is that a segment might be assigned to a wrong cluster, which may harm the subsequent classification step. However, the impact of clustering errors onto the classification result can be possibly reduced, for instance, by using multiple-instance learning (MIL) techniques [Maron and Lozano-Pérez 1998].

6.2.2 Using Partial Supervised Knowledge to Enhance Clustering

In this case, the knowledge generated in a supervised fashion is incorporated in the unsupervised approach as indicated by the arrows (c), (d) and (e).

Regarding the connection (c), if trained statistical models for audio elements are available, we can follow the process (1.1)→(2.2)→(2.3)→(2.4). That is, audio elements can be detected in a supervised fashion, and then the obtained results can be used for unsupervised content discovery at higher abstraction levels, including audio element importance estimation, audio scene segmentation and clustering. The alternative processes involving the connections (d) and (e) follow the same general idea and differ from (c) in the amount of knowledge that the supervised process branch supplies into the unsupervised one. For instance, if we consider the process (1.1)→(1.2)→(2.3)→(2.4), not only the audio elements but also their importance weights are learned in a supervised fashion and then employed in remaining unsupervised knowledge discovery steps.

Combining the supervised and unsupervised components as explained in this section can enhance the semantic inference in well-defined or partially-defined domains, such as, for instance, *tennis* and *football*. In these domains, it is relatively easy to identify and train a priori the sets of characteristic audio elements, while it not easy to predefine higher level semantic entities like scenes. Therefore, supervised classification can successfully be applied for audio element detection, while higher-level semantic inference can best be approached through unsupervised mining. Promising results following such a combined approach were already reported in [Cai et al. 2005].

6.2.3 Enhancing Supervised Approach by Unsupervised Components

The arrow (f), in combination with arrows (c) and (d), is particularly useful in realizing the process paths (1.1)→ (2.2)→(2.3)→(1.4) or (1.1)→(1.2)→(2.3)→(1.4). In these two cases, audio elements and audio scenes are classified in a supervised fashion using trained statistical models, while the unsupervised module is employed to perform automatic scene segmentation instead of (typically) manual pre-segmentation. In this way, the connection (f) leads to a considerable reduction of the manual effort in audio database indexing processes. While collecting training data and learning statistical models is a "*one-time*" cost, manual annotation of scene boundaries is an "*all-time*" cost, as it is required for each audio document separately.

6.3 On Audio Document Clustering and Retrieval

While in the context of this thesis we addressed the problem of clustering so far mainly at the level of audio scenes and within a single audio document, we now consider the case where an audio document as a whole needs to be compared with another audio document for the purpose of audio document clustering or retrieval. Just like in the case of audio scene clustering, a fundamental step in obtaining meaningful clusters of audio documents is document representation. While for clustering short audio clips (e.g. for clustering audio segments into speech, music and noise), such representation can be obtained at the feature level, this is not likely to work in the case of longer audio documents due to the richness of signal mixtures and strong variations in signal properties over time. Clearly, a more sophisticated representation scheme needs to be found for clustering long audio documents, which reveals their high-level similarity and neglects irrelevant signal variations.

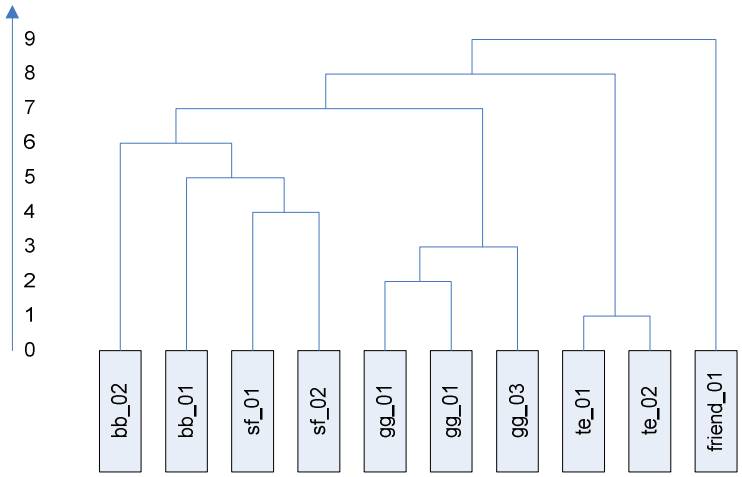


Fig. 6.2 Hierarchical agglomerative clustering of 10 audio documents

Similar to audio scene representation, an audio document can also be represented by a vector containing all the audio elements and their corresponding occurrence probability. Directly building on the audio element sets obtained per audio document using the methodology explained in Chapter 4, we did a preliminary test to investigate audio document clustering performance, in which we simply combined all the document-specific audio elements into an integral audio element set. For this test we used the data from Chapter 4 and 5. To have a sufficient number of audio documents, we manually split each test audio track into several parts, each with about 30-minutes in length, which resulted in 10 audio documents. Due to the fact that several documents stem from the same original audio data stream, each document had one or more documents semantically similar to it. Therefore, the semantic category of the original audio data stream can be taken as ground truth for labeling the audio documents. Also, a "good" clustering process should first cluster the audio documents stemming from the same original source and the same original category.

We applied a hierarchical agglomerative clustering algorithm to the obtained 10 audio documents, as shown in Fig. 6.2. Each audio document initially represented one cluster and at each iteration (indicated on the Y-axis) two most similar clusters were merged together. Also the abbreviations of the document names were used in the leaves of the graph. For example, "*bb_01*" is the first 30-min part from "Band of Brother", and "*sf_02*" is the second 30-min part from "Sword Fish".

The figure shows that most documents were clustered correctly. For example, *sf_01* and *sf_02*, *gg_01*, *gg_02*, and *gg_03*, *te_01* and *te_02* are grouped together within the first four iterations, since they belong to the same original sound track. One exception is *bb_01*, which is first clustered with *sf_01* and *sf_02*, and then with *bb_02*. This becomes understandable if one realizes that these segments are all from action/war movies, and the corresponding audio elements are very similar in these audio documents. These preliminary results indicate that the audio document representation obtained by simply integrating document-specific audio element sets and their weights is likely to lead to good audio document clustering results in the cases structured similarly as in our test.

However, with the increasing number of audio documents being considered, the approach tested above is likely to become less effective. This is because separately extracting audio elements from different audio documents and simply combining them together can lead to a large number of different audio elements in the integral set that all correspond to (conceptually) one and the same sound combination. Due to so many "synonyms", the integral set is likely to become unacceptably large and impractical.

As an alternative, we may also choose to simultaneously build a common set of N audio elements for the entire collection of audio documents. This set can also be referred as *anchor space*, with each anchor representing an eigen audio element. Then, each audio document could be represented by an N-dimensional vector, where each dimension indicates the occurrence probability per audio element in that document. An anchor space can be generated either in a supervised or unsupervised fashion. A supervised approach usually reaches high accuracy and allows control of the semantic level at which anchors are defined. As shown in [Berenzweig and Ellis 2003], the selection of anchor in the case of music classification and similarity computation can be done even at the level as high as artist names and music genres. Having available a set of pre-defined semantic classes and sufficient manually labeled training data (a development data set), a number of supervised learning techniques can be used to train the semantic class of an anchor. These techniques include SVM, HMM, GMM, and neural networks. However, as addressed in Chapter 1, the supervised approach is infeasible if processing an unknown composite audio document, or if audio content semantics is too complex (diverse) to easily select appropriate anchors.

With the objective of expanding the applicability of the anchor space concept onto a general audio content analysis case, we propose an unsupervised method for building an anchor space, which follows the analogy to the approach to audio element discovery from a single document, as explained in Chapter 4. As this approach is based on spectral clustering of audio segments, a practical issue to be resolved when expanding to a large audio document collection is the size of the affinity matrix, on which a SVD

is performed to extract the eigenvectors and map the original data into a low-dimensional space that can be easily clustered (see Chapter 4). If there are 300 audio clips in the development data set, and if each clip has 3 minutes (i.e. around 360 elementary audio segments), the size of the affine matrix will be around $(300*360)^2$ *4B > 40GB (each value in the matrix is a 4-byte float). Such a matrix is impractical to handle and slows down the SVD considerably. To resolve this, a simplification scheme is employed: Instead of using all feature vectors, we represent each audio document by the mean vector only (averaging the feature vectors therein), and then apply spectral clustering on the set of the mean vectors computed for the entire development data set. The obtained clusters are then adopted as audio elements (anchors). Regarding the number of clusters to be formed, spectral clustering proposes an estimation approach based on the eigen-gap. However, in our experimental setup, we manually set various cluster numbers to investigate its effect on the final similarity measure.

With the obtained set of anchors (C_1, C_2, …, C_n), the mapping of an audio document onto this anchor space can be represented by the vector

$$\left[p(C_1 \mid d), \quad p(C_2 \mid d), \quad ..., \quad p(C_n \mid d)\right] \tag{6.1}$$

Here, $p(C_i \mid d)$ represents the membership (posterior probability) of the audio document d with respect to the anchor C_i. The probability $p(C_i \mid d)$ can be further calculated as following, assuming that the prior $p(C_i)$ is uniformly distributed:

$$p(C_i \mid d) = p(d \mid C_i)\, p(C_i) / p(d) = \frac{p(d \mid C_i)}{\sum_i p(d \mid C_i)} \tag{6.2}$$

where the likelihood $p(d \mid C_i)$ is calculated as

$$p(d \mid C_i) = p(s_1, ..., s_N \mid C_i) = \prod_{k=1}^{N} p(s_k \mid C_i) \tag{6.3}$$

Here, s_k is the k-th audio segment in the audio document d, N is the segment number, and $p(s_k \mid C_i)$ is the segment likelihood given the anchor C_i.

Compared to the above "document-level normalization" (normalization of $p(d \mid C_i)$ according to (6.2)), we can also employ "segment-level normalization", that is, we can first map each audio segment onto an anchor, then normalize the likelihood vector $p(s_k \mid C_i)$ for each audio segment, and finally obtain the audio document representation by averaging the memberships of all the audio segments per anchor, that is,

$$p(C_i \mid d) \propto \frac{1}{N} \sum_{k=1}^{N} p(C_i \mid s_k) \tag{6.4}$$

where $p(C_i \mid s_k) = \frac{p(s_k \mid C_i)}{\sum_i p(s_k \mid C_i)}$ is the posterior probability of each audio segment with respect to anchor C_i. Independent of which normalization approach is employed, the representation vector (6.1) makes it possible to employ *KL Divergence* for computing the distance between two audio documents. The more similar the mappings of two documents, the more similar they are.

To test the proposed unsupervised anchor space generation idea, we formed a large dataset including 3000 audio documents that were extracted as sound tracks of the video clips from MSN Video. Each audio document lasts 2-5 minutes, and is associated with a category (also obtained from MSN Video). There are in total 15 categories, including *Autos*, *Business*, *Entertainment*, *Games*, *Live Music*, *Sports*, *Weathers*, and so on. To compare the proposed unsupervised approach with a supervised one, we also implemented a supervised approach in which each anchor is modeled as a GMM. To learn the model, we randomly chose 300 documents as a development data set to build mid-level content representation. The rest of the audio documents are used as a test set. For the sake of completeness, we also included the results obtained by computing the feature-based similarity of audio documents. Once the features are extracted from an audio document per audio frame, we either represent the document by averaging the feature vectors over all frames, or model the feature statistics in the document using a GMM [Lu and Hanjalic 2008b].

For the evaluation strategy, we apply a leave-one-out approach, that is, we select each audio document in the test set as a query, after which all other audio documents are ranked based on their similarity. The documents belonging to the same category are assumed similar in our experiments. *Mean average precision* (*mAP*), a common metric in information retrieval, is employed to quantify the retrieval performance. The *mAP* is actually the mean value of the average precisions (*AP*) computed for each query separately. To obtain the *AP* value for a particular query, the precision is first computed at each relevant document retrieved, and then these precisions are averaged over the entire test data set. Clearly, the more relevant documents occur higher in the ranked document list, the higher the *AP*. The *AP* value per query can be computed using the expression,

$$AP = \sum_{r=1}^{M} P(r) \times rel(r) \tag{6.5}$$

where r is the rank, M is the size of the test set, $rel(r)$ is a binary function indicating the relevance of the audio document at rank r with respect to the query, and $P(r)$ is the precision at top r returned documents.

Table 6.1 Comparison of the audio-element based and features-based audio document representation. In the feature based approach, audio document is either represented as a mean vector or as a Gaussian model. In audio-element based approach, various configurations are tried, comparing the supervised approach (sup) vs. unsupervised anchor space building (unsup); segment-level normalization (segl) vs. document-level normalization (docl); and various cluster numbers (as the first number in the first column indicates)

Audio Doc. Rep.	mAP	mAP25	mAP50	mAP100
Mean	41.4	71.9	66.8	61.2
Gaussian	43.3	72.6	67.6	62.1
[10, unsup, docl]	44.0	62.1	58.3	54.7
[10, unsup, segl]	45.0	65.6	61.7	57.8
[16, unsup, docl]	46.0	65.4	61.5	57.7
[16, unsup, segl]	48.5	70.6	66.7	62.7
[20, unsup, segl]	49.7	71.4	67.6	63.9
[24, unsup, segl]	50.7	72.5	68.8	65.1
[28, unsup, segl]	50.4	73.0	69.2	65.2
[15. sup, docl]	61.3	73.2	71.4	69.3
[15. sup, segl]	58.7	77.3	74.3	71.2

Next to the *mAP*, the *mAP@N* is also evaluated, which represents the mean average precision at the top *N* ranks (similar to (6.5), but with a fixed *N* replacing *M*). The latter metric may be practically useful since the users are usually ready to review only the first *N* retrieved documents and do not want to check the entire data set.

Table 6.1 shows the audio document retrieval performance comparing the audio-element based audio document representation and feature-based audio document representation. In audio element based approach, various numbers of audio elements are tried, including 10, 16, 20, 24, and 28. The best mAP (50.7%) is achieved by the audio element based approach with the cluster number 24 and with segment-level normalization. This corresponds to absolute improvement of 7% compared to the feature-based approach. However, the best result obtained using a supervised approach resulted in 10% accuracy improvement, compared with the unsupervised approach. This shows that there is still considerable room for improvement of the unsupervised approach.

References

[von Ahn and Dabbish 2004] von Ahn, L., and Dabbish, L. "Labeling Images with a Computer Game," in *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, 319-326, 2004

[Baeza-Yates and Ribeiro-Neto 1999] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, 1999

[Baillie and Jose 2003] Baillie, M. and Jose, J.M. "Audio-based Event Detection for Sports Video," in *Proc. Int'l Conf. Image and Video Retrieval, LNCS*, vol. 2728, 300-309, 2003.

[Beeferman et al. 1999] Beeferman, D., Berger, A., and Lafferty, J. "Statistical Models for Text Segmentation," Machine Learning, vol. 34, no. 1-3, 177-210, 1999

[Boreczky and Wilcox 1998] Boreczky, J. S. and Wilcox, L.D. "A Hidden Markov Model Frame Work for Video Segmentation Using Audio and Image Features," *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing'98*, 3741-3744, Seattle, May 1998.

[Bregman 1990] Bregman, A. S. *Auditory Scene Analysis*, MIT Press, 1990

[Brin and Page 1998] Brin, S. and Page, L. "The Anatomy of a Large-Scale Hypertextual Web Search Engine, " In *Proc. World Wide Web Conference*, 107–117, 1998.

[Brummer 1994] Brummer, J.N.L. "Speaker Recognition over HF Radio after Automatic Speaker Segmentation," *In Proc. IEEE South African Symposium on Communications and Signal Processing*, 171-176, 1994

[Cai et al. 2003a] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. "Highlight Sound Effects Detection in Audio Stream," In *Proc. the 4th IEEE International Conference on Multimedia and Expo*, Vol. 3, 37-40, 2003

[Cai et al. 2003b] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. "Using Structure Patterns of Temporal and Spectral Feature in Audio Similarity Measure," in *Proc. 11th ACM Multimedia*, 219-222, Berkeley, CA, 2003

[Cai et al. 2004] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. "Improve Audio Representation by Using Feature Structure Patterns," In *Proc. the 29th IEEE*

International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, 345-348, 2004.

[Cai et al. 2005] Cai, R., Lu, L. and Cai, L.-H. "Unsupervised Auditory Scene Categorization via Key Audio Effects and Information-Theoretic Co-Clustering," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing 05*, Vol. II, 1073-1076, 2005

[Cai et al. 2006] Cai, R., Lu, L., Hanjalic, A., Zhang, H.-J. and Cai, L.-H. "A Flexible Framework for Key Audio Effects Detection and Auditory Context Inference," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14, No. 3, 1026 – 1039, 2006

[Casey 1998] Casey, M. A. *Auditory Group Theory with Applications Statistical Basis Methods for Structured*. Ph.D dissertation, MIT Press, 1998

[Casey 2001] Casey, M.A. "MPEG-7 Sound-Recognition Tools," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 11, No. 6, 737-747, 2001

[Casey et al. 2008] Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. "Content-Based Music Information Retrieval: Current Directions and Future Challenges," *Proceedings of the IEEE*, vol. 96, no. 4, 668-696, 2008

[Chan et al. 1993] Chan, P.K., Schlag, M.D.F., and Zien, J.Y. "Spectral K-way Ratio-cut Partitioning and Clustering," in *Proc. the 30th international conference on Design Automation*, 749-754, 1993

[Chen and Gopalakrishnan 1998] Chen, S., and Gopalakrishnan, P. S. "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 127-132, 1998

[Cheng et al. 2003] Cheng, W.-H., Chu, W.-T., and Wu, J.-L. "Semantic Context Detection based on Hierarchical Audio Models," in *Proc. the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2003, 109-115

[Chu et al. 2005] Chu, W.-T., Cheng, W.-H., and Wu, J.-L. "Generative and Discriminative Modeling toward Semantic Context Detection in Audio Tracks," in *Proc. IEEE MMM'05*, 38 – 45, 2005

[Cohen and Lapidus 1996] Cohen, A. and Lapidus, V. "Unsupervised Speaker Segmentation in Telephone Conversations," in *Proc. the 19th Convention of Electrical and Electronics Engineers in Israel*, 102-105, 1996

[Dhillon 2001] Dhillon, I.S. "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," in *Proc. the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 269-274, San Francisco, CA, USA, 2001.

[Dhillon and Guan 2003] Dhillon, I. S., and Guan, Y. “Information Theoretic Clustering of Sparse Co-occurrence Data,” in *Proc. the 3rd IEEE International Conference on Data Mining*, 517-520, 2003

[Dhillon et al. 2003] Dhillon, I. S., Mallela, S., and Modha, D. S. “Information-theoretic Co-clustering,” in *Proc. the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 89-98, 2003

[Duan et al. 2003] Duan, L.Y., Xu, M., Chua, T.S., Tian, Q., and Xu, C.S. “A Mid-Level Representation Framework for Semantic Sports Video Analysis,” in *Proc. ACM Multimedia*, 33-44, 2003

[Duda et al. 2000] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification, Second Edition*. John Wiley & Sons, NJ, 2000

[Ellis and Lee 2004] Ellis, D., and Lee, K. “Minimal-impact Audio-based Personal Archives,” in *Proc. ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 39-47, 2004

[Ellis1996] Ellis, D. P. W. *Prediction-driven Computational Auditory Scene Analysis.* Ph.D dissertation, MIT Press, 1996

[El-Maleh et al. 2000] El-Maleh, K., Klein, M., Petrucci, G., and Kabal, P. “Speech/Music Discrimination for Multimedia Application,” in *Proc. Int’l Conf. on Acoustics, Speech, and Signal Processing*, Vol.4, 2445-2448, 2000

[El-Yaniv and Souroujon 2001] El-Yaniv, R., and Souroujon, O. “Iterative double clustering for unsupervised and Semi-supervised Learning,” *Proc. ECML'01*, 121-132, 2001

[Eronen et al. 2006] Eronen, A., Peltonen ,V., Tuomi, J., Klapuri, A.P., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. “Audio-based Context Recognition," *in IEEE Trans. Audio Speech Language Processing*, 14(1), 321-328, 200

[Feng et al. 2004] Feng, S.L., Manmatha, R., and Lavrenko, V. “Multiple Bernoulli Relevance Models for Image and Video Annotation,” in *Proc the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, 1002-1009, 2004

[Fletcher and Rossing 1998] Fletcher, N.H., and Rossing, T.D. *The Physics of Musical Instruments*, Springer-Verlag New York, Inc., 1998.

[Foote 1997] Foote, J. “Content-based Retrieval of Music and Audio,” in *Kuo, C. C. J., et al. eds., Multimedia Storage and Archiving Systems II, Proc. SPIE*, Vol. 3229, 138-147, 1997.

[Friedman 1998] Friedman, N. “The Bayesian Structure EM Algorithm,” *Proc. UAI'98*, 129-138, 1998

[Fujinaga 1998] Fujinaga, I. “Machine Recognition of Timbre using Steady-state Tone of Acoustic Instruments,” in *Proc. International Computer Music Conference*, 207–210, 1998

[Gish et al. 1991] Gish, H., Siu, M.H., and Rohlicek. R. “Segregation of Speakers for Speech Recognition and Speaker Identification,” in *Proc. Int’l Conf. on Acoustics, Speech, and Signal Processing’91*, 873-876, 1991

[Gygi 2001] Gygi, B. *Factors in the Identification of Environmental Sounds*, Ph.D. dissertation, Dept. Psycho., Indiana Univ., IN, USA, Jun. 2001.

[Hall and Llinas 1997] Hall, D.L., and Llinas, J. “An Introduction to Multisensor Data Fusion,” *Proceedings of the IEEE*, 85(1), 6-23, 1997

[Hanisch et al. 2002] Hanisch, D., Zien, A., Zimmer, R., Lengauer, T. "Co-clustering of Biological Networks and Gene Expression Data," *Bioinformatics*, vol. 18, no. 90001, S145-S154, 2002

[Hanjalic 2004] Hanjalic, A. *Content-based Analysis of Digital Audio*, Kluwer Academic Publishers, 2004

[Hanjalic and Xu 2005] Hanjalic, A., and Xu, L.-Q. “Affective Video Content Representation and Modeling,” *IEEE Trans. Multimedia*, Vol. 7, No. 1, 143-154, Feb. 2005.

[Hanjalic et al. 1999] Hanjalic, A., Lagendijk, R. L., and Biemond, J. “Automated High-level Movie Segmentation for Advanced Video-Retrieval Systems,’ *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 9, No. 4, 580-588, Jun. 1999

[Hanjalic et al. 2005] Hanjalic, A., Nesvadba, J., and Benois-Pineau, J. “Moving away from narrow-scope solutions in multimedia content analysis,” *The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT 2005)*, London 2005

[Hanjalic et al. 2008] Hanjalic, A., et al. eds. “Advances in Multimedia Information Retrieval,” in *Proceedings of the IEEE*, Vol.96, No.4, April 2008

[Hastie et al. 2001] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001

[Heckerman 1995] Heckerman, D. *A Tutorial on Learning with Bayesian Networks*, Technical Report, Microsoft Research, MSR-TR-95-06, 1995

[Herlocker et al. 2000] Herlocker, J.L., Konstan, J.A., and Riedl, J. “Explaining Collaborative Filtering Recommendations,” in *Proc. of ACM Conference on Computer Supported Cooperative Work*, 241 – 250, 2000

[Hua et al. 2005] Hua, X.S., Lu, L., and Zhang, H.-J. "*Robust Learning-based TV Commercial Detection*," in *Proc. IEEE International Conference on Multimedia and Expo*, 149-152, Amsterdam, The Netherlands, 2005

[Huang and Darwiche 1996] Huang, C. and Darwiche, A. "Inference in Belief Networks: a Procedural Guide," *International Journal of Approximate Reasoning*, Vol. 15, No. 3, 225-263, 1996

[Huang et al. 2001] Huang, X., Acero, A., and Hon, H.W. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR Upper Saddle River, NJ, 2001

[Hyvarinen and Oja 2000] Hyvarinen, A. and Oja, E. "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, vol. 13, no. 4-5, 411-430, 2000

[ISMIR] online: http://www.ismir.net/

[Jain and Dubes 1988] Jain, A. K., and Dubes, R. C. *Algorithms for Clustering Data*, Prentice Hall College Div., 1988

[Jeon et al. 2003] Jeon, J., Lavrenko, V., Manmatha, R. "Automatic Image Annotation and Retrieval using Cross-Media Relevance Models," in *Proc. ACM SIGIR Conference Research & Development on Information Retrieval*, 119-126, 2003

[Kass and Wasserman 1995] Kass, R.E., and Wasserman, L. "A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwarz Criterion," *Journal of the American Statistical Association*, Vol.90, No.431, 928-934, 1995

[Kender and Yeo 1998] Kender, J. R., and Yeo, B.-L. "Video Scene Segmentation via Continuous Video Coherence," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 367-373, 1998

[Kern et al. 2007] Kern N., Schmidt A., and Schiele B. "Recognizing Context for Annotating a Live Life Recording," in *J. Personal and Ubiquitous Computing*, 11(4), 251-263, 2007

[Kimber and Wilcox 1996] Kimber, D., and Wilcox, L. "Acoustic Segmentation for Audio Browsers," in *Proc. Interface Conference*, Sydney, Australia, July, 1996

[Li 2000] Li, S. Z. "Content-based Classification and Retrieval of Audio using the Nearest Feature Line Method," *IEEE Transactions on Speech and Audio Processing*, 8(5), 619-625, September 2000

[Li and Khokhar 2000] Li, G., and Khokhar, A.A. "Content-based Indexing and Retrieval of Audio Data using Wavelets," in *Proc. IEEE International Conference on Multimedia and Expo*, Vol.2, 885-888, 2000

[Liu et al. 1998] Liu, Z. Wang, Y. and Chen, T. “Audio Feature Extraction and Analysis for Scene Segmentation and Classification,” *Journal of VLSI Signal Processing Systems 20*, 61-49, June 1998

[Lu and Hanjalic 2006] Lu, L., and Hanjalic, A. “Towards Optimal Audio Keywords Detection for Audio Content Analysis and Discovery," *Proc. ACM Multimedia 06*, 825-834, 2006

[Lu and Hanjalic 2008a] Lu, L., and Hanjalic, A. “Audio Keywords Discovery for Text-Like Audio Content Analysis and Retrieval,” *IEEE Trans. on Multimedia*, vol 10, no. 1, 74-85, 2008

[Lu and Hanjalic 2008b] Lu, L., and Hanjalic, A. “Unsupervised Anchor Space Generation for Similarity Measure of General Audio,” *Proc. ICASSP‘08*, 53-56, 2008

[Lu and Zhang 2002] Lu, L., and Zhang, H.-J. “Speaker Change Detection and Tracking in Real-Time News Broadcasting Analysis,” in *Proc. the 10th ACM Multimedia*, 602- 610, Juan-les-Pins, France, 2002

[Lu and Zhang 2005] Lu, L., and Zhang, H.-J. “Unsupervised Speaker Segmentation and Tracking in Real-Time Audio Content Analysis,” *ACM/Springer Multimedia Systems Journal* 10 (4), 332-343, 2005

[Lu et al. 2001] Lu, L., Jiang, H., and Zhang. H.-J. “A Robust Audio Classification and Segmentation Method,” in *Proc. the 9th ACM Multimedia*, 203-211, 2001

[Lu et al. 2002] Lu, L., Zhang, H.-J., and Jiang, H. “Content Analysis for Audio Classification and Segmentation,” *IEEE Trans. Speech Audio Processing*, 10(7), 504-516, 2002

[Lu et al. 2003] Lu, L., Zhang, H.-J., and Li, S., “Content-based Audio Classification and Segmentation by Using Support Vector Machines,” *ACM Multimedia Systems Journal*, 8(6), 482-492, March, 2003

[Lu et al. 2005] Lu, L., Cai, R., and Hanjalic, A. “Towards a Unified Framework for Content-based Audio Analysis, ” in. *Proc. Int’l Conf. on Acoustics, Speech, and Signal Processing’05*, vol. II, 1069-1072, 2005

[Ma et al. 2002] Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M.-J. “A User Attention Model for Video Summarization,” in *Proc. the 10th ACM Multimedia*, 533-542, 2002

[Madeira and Oliveira 2004] Madeira S.C. and Oliveira A.L. “Biclustering Algorithms for Biological Data Analysis: a Survey,” *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 1(1), 24-45, 2004

[Maron and Lozano-Pérez 1998] Maron, O., and Lozano-Pérez, T. "A framework for Multiple Instance Learning," *Proc. Advances in Neural Information Processing Systems 10*, 570-576, July 1998

[Moncrieff et al. 2001] Moncrieff, S., Dorai, C., and Venkatesh, S. "Detecting Indexical Signs in Film Audio for Scene Interpretation," in *Proc. the 2nd IEEE International Conference on Multimedia and Expo*, 989-992, 2001,

[Moreno and Rifkin 2000] Moreno, P. J., and Rifkin, R. "Using the Fisher Kernel Method for Web Audio Classification". In *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol. IV, 2417-2420, 2000

[Mori and Nakagawa 2001] Mori, K., and Nakagawa, S. "Speaker Change Detection and Speaker Clustering Using VQ Distortion for Broadcast News Speech Recognition," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol. I, 413-416, 2001

[Naphade and Huang 2000] Naphade, M.R., and Huang, T.S. "A Probabilistic Framework for Semantic Indexing and Retrieval in Video," in *Proc. IEEE International Conference Multimedia and Expo*, Vol. 1, 475-478, 2000

[Naphade et al. 2001] Naphade, M.R., Wang, R. and Huang, T.S. "Supporting Audiovisual Query using Dynamic Programming," in *Proc. the 9th ACM Multimedia*, 411-420, 2001

[Ng et al. 2001] Ng, A. Y., Jordan, M. I., and Weiss, Y. "On Spectral Clustering: Analysis and an Algorithm," in *Proc. Advances in Neural Information Processing Systems (NIPS) 14*, 849-856, 2001

[Ngo et al. 2001] Ngo, C.-W., Ma, Y.-F., and Zhang, H.-J. "Video Summarization and Scene Detection by Graph Modeling," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 15, No. 2, 296-305, 2005

[Pelleg and Moore 2000] Pelleg, D., and Moore, A. W. "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in *Proc. the 17th International Conference on Machine Learning*, 727-734, 2000

[Peltonen et al. 2002] Peltonen, V., Tuomi, J., Klapuri, A.P., Huopaniemi, J., and Sorsa, T. "Computational Auditory Scene Recognition," in *Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol.2, 1941-1944, 2002

[Pfeiffer et al 1996] Pfeiffer, S., Fischer, S., and Effelsberg, W. "Automatic Audio Content Analysis," in *Proc. the 4th ACM Multimedia*, 21-30, 1996.

[Pye 2000] Pye, D. "Content-Based Methods for the Management of Digital Music," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing'00*, Vol. IV, 2437-2440, 2000

[Rabiner 1989] Rabiner, L.R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, 257-286, 1989

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

[Radhakrishnan et al. 2004] Radhakrishnan, R., Divakaran, A., and Xiong, Z. "A time Series Clustering based Framework for Multimedia Mining and Summarization using Audio Features," in *Proc. the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 157-164, 2004

[Reyes-Gomez and Ellis 2003] Reyes-Gomez, M.J., and. Ellis, D.P.W. "Selection, Parameter Estimation, and Discriminative Training of Hidden Markov Models for General Audio Modeling," in *Proc. IEEE International Conference on Multimedia and Expo*, Vol. 1, 73-76, Baltimore, MD, USA, Jul. 2003.

[Robertson and Jones 1997] Robertson, S.E., and Jones, K.S. *Simple, Proven Approaches to Text Retrieval*, Technical Report 356, University of Cambridge, 1997

[Rossignol et al.1998] Rossignol, S., Rodet, X., Soumagne, J., Collette, J.-L., and Depalle, P. "Features Extraction and Temporal Segmentation of Acoustic Signals," in *Proc. International Computer Music Conference*, 199–202, 1998

[Rowe and Jain 2005] Rowe, L.A., and Jain, R. "ACM SIGMM Retreat Report on Future Directions in Multimedia Research," *ACM Trans. on Multimedia Computing, Communications, and Applications*, Vol. 1, No.1, 3-13, 2005

[Rui et al. 2000] Rui, Y., Gupta, A., and Acero, A. "Automatically Extracting Highlights for TV Baseball Programs," in *Proc. the 8th ACM Multimedia*, 105-115, 2000

[Saunders 1996] Saunders, J. "Real-time Discrimination of Broadcast Speech/Music," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing'96*, Vol. II, 993-996, 1996

[Scheirer and Slaney 1997] Scheirer, E., and Slaney, M. "Construction and Evaluation of a Robust Multifeature Music/Speech Discriminator," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing'97*, Vol II, 1331-1334, 1997

[Scott and Longuet-Higgins 1990] Scott, G.L., and Longuet-Higgins, H. C. "Feature Grouping by Relocalisation of Eigenvectors of the Proximity Matrix," in *Proc. British Machine Vision Conference*, 103-108, 1990

[Shi and Malik 2000] Shi, J., and Malik, J. "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, 888-905, 2000

[Shlens 2005] Shlens, J. "A Tutorial on Principal Component Analysis", 2005, available: http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf

[Siu et al. 1992] Siu, M.H., Yu, G., and Gish, H. "An Unsupervised, Sequential Learning Algorithm for the Segmentation of Speech Waveform with Multiple Speakers," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing'92*, 189-192, 1992

[Smith et al. 1998] Smith, G., Murase, H., and Kashino, K. "Quick Audio Retrieval using Active Search," *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing'98*, Vol. 6, 3777-3780, 1998

[Srinivasan et al. 1999] Srinivasan, S., Petkovic, D., and Ponceleon, D. "Towards Robust Features for Classifying Audio in the CueVideo System," in *Proc. the 7th ACM Multimedia*, 393 – 400, 1999

[Stewart and Sun 1990] Stewart, G.W., Sun J. *Matrix Perturbation Theory*, Academic Press, 1990

[Sugiyama et al. 1993] Sugiyama, M., Murakami, J., and Watanabe, H. "Speech Segmentation and Clustering Based on Speaker Features," in *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol.2, 295-398, 1993

[Sundaram and Chang 2000] Sundaram, H., and Chang, S.-F. "Audio Scene Segmentation Using Multiple Feature, Models and Time Scales," in *Proc. IEEE Int'l Conf. on Acoustic, Speech and Signal Processing*, Vol.4, 2441-2444, 2000

[TRECVID] online: http://www-nlpir.nist.gov/projects/trecvid/

[Tzanetakis and Cook 2000] Tzanetakis, G., and Cook, P. "MARSYAS: a Framework for Audio Analysis," Organised Sound, 4: 169-175, Cambridge University Press, 2000

[Venugopal et al. 1999] Venugopal, S., Ramakrishnan, K.R., Srinivas, S.H. and Balakrishnan, N. "Audio Scene Analysis and Scene Change Detection in the MPEG Compressed Domain," in *Proc. MMSP99*, 191-196, 1999

[Wall et al. 2003] Wall, M.E., Rechtsteiner, A., Rocha, L.M. "Singular Value Decomposition and Principal Component Analysis," in *Berrar, D.P., et al. eds. A Practical Approach to Microarray Data Analysis*, 91-109, Kluwer: Norwell, MA, 2003, LANL LA-UR-02-4001

[Wang and Brown 2006] Wang, D.L. and Brown, G. J. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Wiley-Interscience, 2006

[Wang et al. 2003] Wang, D., Lu, L., and Zhang, H.-J. "Speech Segmentation without Speech Segmentation," in *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol. I, 468-471, 2003

[Weiss 1999] Weiss, Y. "Segmentation Using Eigenvectors: A Unifying View," in *Proc. International Conference on Computer Vision*, Vol.2, 975- 982, 1999

[Westner 1998] Westner, A. G. *Object-Based Audio Capture: Separating Acoustically-Mixed Sounds*. Master thesis, MIT press, 1998

[Wilcox et al. 1994] Wilcox, L., Chen, F., Kumber, D. and Balasubramanian, V. "Segmentation of Speech Using Speaker Identification," in *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, 161-164, 1994

[Wold et al. 1996] Wold, E., Blum, T., and Wheaton, J. "Content-based Classification, Search and Retrieval of Audio," *IEEE Multimedia*, 3(3), 27-36, 1996

[Wu and Leah 1993] Wu, Z. and Leahy, R. "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, 1101-1113, Nov. 1993

[Xie et al. 2003] Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. "Unsupervised Mining of Statistical Temporal Structures in Video," in Rosenfeld, A., eds. *Video Mining*, Kluwer Academic Publishers, 279-307, 2003

[Xiong et al. 2003] Xiong, Z., Radhakrishnan, R., Divakaran, A., and Huang, T.S. "Audio Events Detection based Highlights Extraction from Baseball, Golf and Soccer Games in a Unified Framework," in *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, Vol. 5, 632-635, 2003

[Xu et al. 2003] Xu, M., Maddage, N., Xu, C.-S., Kankanhalli, M., and Tian, Q. "Creating Audio Keywords for Event Detection in Soccer Video," in *Proc. the 4th IEEE International Conference on Multimedia and Expo*, Vol. 2, 281-284, 2003

[Yu and Shi 2003] Yu, S. X., and Shi, J. "Multiclass Spectral Clustering." In *Proc. of the 9th IEEE International Conference on Computer Vision*, Vol. 1, 313-319, 2003

[Zelnik-Manor and Perona 2004] Zelnik-Manor, L., and Perona, P. "Self-tuning Spectral Clustering," in *Proc. Advances in Neural Information Processing Systems (NIPS) 17*, 1601-1608, 2004

[Zhang and Kuo 1998] Zhang, T., and Kuo, C.-C. J. "Hierarchical System for Content-based Audio Classification and Retrieval," in *Proc. SPIE, Multimedia Storage and Archiving Systems III*, Vol. 3527, 398-409, 1998.

[Zhang and Kuo 1999] Zhang, T., and Kuo, C.-C. J. "Video Content Parsing based on Combined Audio and Visual Information," in *Multimedia Storage and Archiving Systems IV, Proc. of SPIE 3846*, Vol. IV, 78-89, 1999

Summary

In this thesis, we developed and assessed a novel robust and unsupervised framework for semantic inference from composite audio signals. We focused on the problem of detecting audio scenes and grouping them into meaningful clusters. Our approach addressed all major steps in a general process of composite audio analysis, from low-level signal processing (feature extraction), via mid-level content representation (audio element extraction and weighting), to high-level semantic inference (audio scene detection and clustering). We showed experimentally that our proposed content discovery scheme involving mid-level semantic descriptors as an intermediate inference result can lead to more robustness, compared to the classical content-based audio indexing approach, where the semantics is inferred from the features directly. To the best of our knowledge, this is the first proposal exploring the possibilities for a realization of an entirely unsupervised audio content discovery system aiming at high-level semantic inference results.

The first major algorithmic contribution of the thesis consists of an unsupervised approach to decompose an audio stream into (key) audio elements, based on a set of extracted audio signal features. Similar to speech recognition that transcribes a speech signal into text words, our proposed approach "transcribes" a composite audio signal into audio "words", where each word corresponds to a short temporal segment with coherent signal properties (e.g. music, speech, noise or any combination of these). We refer to these audio words as audio elements. To extract audio elements, we deployed an iterative spectral clustering method with context-dependent scaling factors. In this process, the elementary audio segments with similar features are grouped together into clusters. Then, all audio segments belonging to the same cluster are said to represent the same audio element. We now see an audio signal as a concatenation of audio segments corresponding to different audio elements, and develop an approach similar to those known from the text document segmentation field to divide the signal into meaningful longer segments. We refer to these segments as audio scenes. To develop such an approach, we computed the weights indicating the potential of each obtained audio element to help detect an audio scene boundary. To compute these weights, again

the concepts from text information retrieval have been adopted, such as the term frequency (TF) and inverse document frequency (IDF), based on which a number of their equivalents in the audio segmentation context have been introduced.

As the second major algorithmic contribution of the thesis, we presented a novel approach to audio scene segmentation and clustering. We first proposed a semantic affinity measure to determine whether two audio segments are likely to belong to the same audio scene. This measure considers the audio elements contained in the analyzed segments, their importance weights and their co-occurrence statistics. Then, the presence of an audio scene boundary at a given time stamp is investigated by jointly considering the values of the semantic affinity computed for a representative number of segment pairs surrounding the observed time stamp. Once the audio scenes are detected, a scheme based on the co-clustering concept was deployed to exploit the grouping tendency among audio elements when searching for optimal audio scene clusters. Here a method based on the Bayesian information criterion (BIC) was adopted to select the numbers of clusters in the co-clustering process.

Experimental evaluations on a large and representative audio data set have shown that the proposed approach can achieve encouraging results and outperform the existing related approaches. The obtained results show a relatively high purity of the obtained audio elements. The number of the obtained elements, the type of sounds they represent and the importance weights assigned to them were shown to largely correspond to the judgment of our test user panel. Moreover, for audio scene segmentation and clustering, we obtained a 70% recall of audio scene boundaries with a 80% precision, based on the ground-truth annotation obtained using a panel of human annotators. Our co-clustering based approach achieved better performance than a traditional one-directional clustering, regarding both the clustering accuracy and cluster number estimation.

We completed the thesis by making an attempt to envision a possible expansion of the proposed approach towards an application scope broader than the one considered in the thesis. We first considered the applications where domain knowledge is available. For such an application we investigated the possibilities to combine our unsupervised approach with a supervised one to benefit from the available domain knowledge and so improve the content discovery performance for that domain. Then, we also performed preliminary experiments to extrapolate the applicability of the proposed approach from a single document context to a collection of (long) audio documents. This involved a shift from the concept of document-specific audio elements to an anchor space representing a large collection of audio documents.

Samenvatting

Dit proefschrift beschrijft een robuust, automatisch systeem voor het extraheren van semantisch relevante informatie (ook wel “content discovery” genoemd) uit audiosignalen in multimediale databanken. Typische voorbeelden van dergelijke audiosignalen zijn de “soundtracks” van TV shows, documentaires en films. Kenmerkend voor deze signalen is dat ze uit een vermenging van muziek, ruis en spraak bestaan en dat de onderlinge verhoudingen tussen de verschillende audiomodaliteiten niet voorspelbaar zijn.

Het ontwikkelde systeem concentreert zich op allereerst het detecteren van betekenisvolle “audio scenes”, en vervolgens het groeperen van deze scenes in thematische clusters. Deze clusters zijn potentieel relevant tijdens het doorzoeken van de multimediale databank op basis van semantische zoekcriteria. De gekozen content discovery aanpak dekt alle belangrijke stappen in de audiosignaalanalyse, beginnend met kenmerkextractie tot het detecteren van audio scenes en hun onderliggende relaties, maar onderscheidt zich van de bestaande methoden door als basis de audiosignaalrepresentatie op het niveau van “audio elements” (ook wel “audio words” genoemd) te gebruiken. We laten experimenteel zien dat het voorgestelde systeem robuuster is dan de conventionele aanpak waarin audio scenes direct gedetecteerd worden op basis van signaaleigenschappen. Vergeleken met de bestaande methoden behoeft het detecteren en groeperen van audio scenes geen supervisie en wordt daarom “unsupervised” genoemd.

De eerste bijdrage van dit proefschrift is een methode voor het automatisch detecteren van audio elements. Vergelijkbaar met spraakherkenning, waar spraak naar een tekstdocument vertaald wordt, vertaalt onze methode een audiosignaal naar een opeenvolging van audio words. Hierbij correspondeert elk “woord” met een stuk signaal dat gekenmerkt wordt door bepaalde signaaleigenschappen en gerelateerd aan een specifieke vermenging van muziek, spraak en ruis. Om de audio elements te kunnen extraheren, werd een iteratieve clusteringmethode ontwikkeld, die gebruik maakt van contextafhankelijke schaalfactoren. Bij deze methode worden de

basissegmenten van het audiosignaal van elk één seconde lang die dezelfde signaaleigenschappen hebben bij elkaar gevoegd. Hierdoor worden “clusters” gevormd. Alle audio segmenten die zich binnen hetzelfde cluster bevinden worden gezien als vertegenwoordigers van één en hetzelfde audio element.

De tweede bijdrage van dit proefschrift is een algoritme dat hetzelfde probeert te bereiken als in de tekst documentanalyse, namelijk het audiosignaal in langere, betekenisvolle “paragrafen” te verdelen (audio scenes) en vervolgens deze scenes in thematische clusters te groeperen. Dit algoritme volgt het basisidee van tekstsegmentatie, namelijk het representeren van elke woord door een weegfactor dat de relevantie van het woord weergeeft voor het bepalen van paragraafgrenzen. Om deze weegfactoren te berekenen werden de bekende concepten van “term frequency” (TF) en de “inverse document frequency” (IDF) vanuit de tekstanalyse gebruikt om de geschikte alternatieven in het audio domain te ontwikkelen. De scene segmentatie zelf is gebaseerd op een zogenaamde “semantic affinity measure”. Dit is een maat waarmee de relatie tussen twee paragrafen op het semantisch niveau geschat kan worden. De maat wordt berekend op het niveau van audio elements in de paragrafen en met gebruik van hun weegfactoren en de statistiek van het gezamenlijk optreden van twee audio elements. De maat wordt toegepast op audio elementen ter linker en rechterzijde van het tijdspunt waarvoor de aanwezigheid van een audio scene begrenzing wordt geëvalueerd. De kans op de scene grens wordt groter naar mate meer audio element combinaties een hogere waarde voor de semantic affinity laten zien. Nadat de scenes gedetecteerd zijn, wordt co-clustering toegepast om de scene clusters te vormen. Het aantal clusters wordt automatisch geschat met behulp van het Bayesian Information criterion (BIC).

Uitgebreide experimentele evaluatie van de voorgestelde methoden en algoritmen op representatieve data collecties laten zien dat goede resultaten bereikt worden, die beter zijn dan die van de bestaande methoden. De verkregen resultaten worden vooral gekenmerkt door een relatief hoge nauwkeurigheid van de gedetecteerde audio elements. Het aantal en type van gedetecteerde elements zijn goed in overeenkomst met het oordeel van een gebruikerspanel. In vergelijking met de “ground truth” verkregen door dit panel, heeft onze methode voor audio scene segmentatie en clustering de waarde van “precision” en “recall” bereikt van respectievelijk 80% en 70%. De co-clustering aanpak die we gekozen hebben presteert consistent beter dan het klassieke één-richting clustering zowel voor de kwaliteit van de clusters als voor de schatting van het aantal clusters.

In het laatste hoofdsuk van dit proefschrift beschouwen we de mogelijkheden om de toepassingsmogelijkheden van de voorgestelde methoden en algorithmen te vergroten. Ten eerste hebben we de toepassingen geanalyseeerd waar voldoende domeinkennis

beschikbaar is. In zulke toepassingen hebben we de mogelijkheden onderzocht om onze unsupervised aanpak te combineren met een supervised stap en zo de prestaties van content discovery voor deze domeinen te kunnen verbeteren. Ten tweede passen we de voorgestelde methoden en algoritmen toe op een collectie van lange audio "documenten" en vergelijken de resultaten met onze originele aanpak die voor een enkelvoudig audio document ontworpen was. Om dit te kunnen doen werd het concept van document-specifieke audio elements gegeneraliseerd richting een "anchor space" dat representatief is voor een grote collectie van audio documenten.

Acknowledgments

To pursue a PhD degree was a big decision and a big step in my life. I feel very lucky to have been given the opportunity to become a part-time PhD student in 2004, when I was already working at Microsoft Research Asia (MSRA) in Beijing, China. While being a part-time PhD student may have induced some conflicts between my thesis research and work-related priorities, I was extremely fortunate to have had a number of persons around me who supported and encouraged me in this time period, and made it possible for me to complete this thesis.

First of all, I would like to thank my thesis advisor, Inald Lagendijk, for all the discussions, suggestions, and honest criticisms regarding my work. I still remember that, when he was on a tight business trip in Beijing, he spent his spare time in the evenings, discussing with me the current status of the thesis and my research methodology and plans.

Second, I would like to thank my thesis co-advisor, Alan Hanjalic. Alan worked with me for nearly 5 years. Through numerous invaluable discussions, suggestions, and paper/thesis revisions, he provided a great help to me and coordinated my progress towards the PhD degree. Without his help, I could have not been able to complete the thesis so smoothly.

Third, I would like to thank Hong-Jiang Zhang, my former manager at MSRA. It is him who created the opportunity for me within MSRA to pursue the PhD degree in parallel with my work, and who encouraged and supported me during the entire time period related to my thesis research. My thanks for their support also go to my later managers at MSRA, Hsiao-Wuen Hon, Frank Soong and Frank Seide.

Fourth, I would like to thank Rui Cai, who was a visiting student in our team and who worked with me for about 2 years. We discussed a lot and co-authored several papers together. He also helped run a portion of the experiments that are included in this thesis, for which I am very grateful.

Last but not least, I would like to thank all of the people who helped me during these years, especially my family, who is always on my side and is the one that makes all this effort worthwhile.

Curriculum Vitae

Lie Lu was born in Zhejing, China, 1975. After finishing high school, he started his study at Shanghai Jiao Tong University in 1993, where he received his BS and MS degree, both in Electrical Engineering, in 1997 and 2000 respectively. Since 2000, he has been with Microsoft Research Asia (MSRA), Beijing, China, where he is currently a Researcher with the Speech Group. In 2004, in parallel with his work at MSRA, he became a part-time PhD student at the Delft University of Technology, The Netherlands.

His current research interests include pattern recognition, content-based audio analysis and indexing, and content-based music analysis. He has published extensively in his field of expertise. His bibliography includes 13 papers in scientific journals, 6 book chapters and more than 40 papers in the leading conferences in the area of audio and speech processing and multimedia in general. Moreover, he owns nearly 20 patents or pending applications, and some of the technological solutions he developed have been successfully licensed or transferred to products, such as in Live/Bing Search and Windows MovieMaker.

He is a member of IEEE and ACM. He also served as a member of technical program committee for a dozen of conferences, such as IEEE International Conference on Multimedia and Expo 2004 and 2007, IJCAI Workshop on Multimodal Information Retrieval 2007, ACM Multimedia 2007 and 2008, Pacific-Rim Conference on Multimedia 2007, SIGIR Workshop on Mobile Information Retrieval 2008 and Asia Information Retrieval Symposium 2008.