Generative and Discriminative Modeling toward Semantic Context Detection in Audio Tracks

Wei-Ta Chu
Department of Computer Science and Information Engineering
National Taiwan University
wtchu@cmlab.csie.ntu.edu.tw

Wen-Huang Cheng
Graduate Institute of Networking and Multimedia
National Taiwan University
wisley@cmlab.csie.ntu.edu.tw

Ja-Ling Wu
Graduate Institute of Networking and Multimedia
National Taiwan University
wjl@cmlab.csie.ntu.edu.tw

Abstract

Semantic-level content analysis is a crucial issue to achieve efficient content retrieval and management. We propose a hierarchical approach that models the statistical characteristics of several audio events over a time series to accomplish semantic context detection. Two stages, including audio event and semantic context modeling/testing, are devised to bridge the semantic gap between physical audio features and semantic concepts. For action movies we focused in this work, hidden Markov models (HMMs) are used to model four representative audio events, i.e. gunshot, explosion, car-braking, and engine sounds. At the semantic context level, generative (ergodic hidden Markov model) and discriminative (support vector machine, SVM) approaches are investigated to fuse the characteristics and correlations among various audio events, which provide cues for detecting gunplay and car-chasing scenes. The experimental results demonstrate the effectiveness of the proposed approaches and draw a sketch for semantic indexing and retrieval. Moreover, the differences between two fusion schemes are discussed to be the reference for future research.

1. Introduction

Recently, large amount of multimedia content has been created, stored, and disseminated as rapid advance in media creation, storage, and compression technologies. However, massive multimedia data often depress users in content browsing and retrieving and diminish the benefits brought by digital media. Technologies for effective and efficient multimedia document indexing, therefore, are indispensable to ease the load of media access.

Some research issues have been investigated to facilitate efficient access and usage of massive multimedia data. To improve the effectiveness of browsing and retrieval, techniques for genre classification are widely studied. For audio tracks, some techniques [5, 6] are proposed to discriminate different types of audio and music genre classification [7]. For video content, genres of films [9] and TV programs [10] are automatically classified by exploring various features. In these content analysis techniques, various features from audio, video, and text [11] are exploited, and many multimodal approaches are proposed to efficiently cope with the access and retrieval issues of multimedia content.

Although the paradigms described above are efficient for browsing and content-based search, some problems exist in today's applications. The first is the apparent gap between low-level audiovisual features and high-level semantics. The similarity in low-level features often mismatch with user's perception. The second problem is that, from the viewpoint of end-users, it's not intuitive for people to retrieve video shots with color layouts or motion trajectories. A tool that provides semantic-level query is more practical than that one just supports unlabeled shots or rough genre classification.

Recently there are two research directions about analyzing multimedia documents from user's point of view. The first one is to find the attractive parts of movies or TV programs by exploiting the domain knowledge and production rules. According to *media aesthetics* [12], which are defined as the study and analysis of media elements such as lighting, motion, color, and sound both by themselves and their roles in synthesizing effective productions, these researches attempt to uncover the semantic and semiotic information by computational frameworks [13]. Some preliminary results have been reported on film tempo analysis [13] and scare scene detection by using audio dynamics in horror movies [15].

The second direction of user-centric multimedia analysis is to construct semantic indices for multimedia documents. Studies on semantic indexing can be classified as two categories according to the granularity they proceed:

1) isolated audio/video event detection and 2) semantics identification based on fusing the detection result of isolated events. In [16], several audio highlight events, such as applause, laughter, and cheer, are modeled by HMMs [18]. In the test stage, audio features of each audio segment are extracted to be the inputs of these three highlight event models, and the highlight events in an audio clip are detected via a decision algorithm.

The aforementioned approaches primarily detect audio events or video objects in audiovisual streams. However, detecting isolated audio/video events is still not close to user's notion. For example, we would not like to find when gunshots happen in an action movie. We would likely find the scene of gunplay, which may consist of gunshots, explosions, sounds of jeeps, and screams from soldiers for a while. This kind of scene conveys a solid semantic meaning and is at a reasonable granularity for semantic retrieval. Instead of just modeling isolated events, several approaches based on Bayesian network [2] and Gaussian mixture models [3] have been proposed to fuse the information of isolated events. Naphade and Huang [2] adopt a probabilistic framework, i.e. dynamic Bayesian network, to model semantic concepts, such as the scenes with 'outdoor' or 'beach' concepts. This framework models the correlations between different objects/events and provides inference functionalities. It fuses various multimedia objects and to infer objects that are not easy to be modeled directly from low-level features. However, they didn't model semantic contexts that involve several multimedia objects over a time series. In multimedia retrieval, semantic contexts that provide users with a complete and continuous semantic meaning often serve as the basic units of query. Therefore, a fusion scheme that models various audio events along the temporal axis should be devised to describe the evolution of semantic contexts in audiovisual streams.

In this paper, an integrated hierarchical framework is proposed to detect semantic contexts in action movies, which often consist of apparent audio events to show impressive scenes. Because there are often rapid shot changes and dazzling visual variations in this type of movies, we focus our investigation on analyzing audio tracks and accomplish semantic indexing via aural clues. By using the HMM-based approaches presented in [16], low-level events, such as gunshot and explosion sounds, are modeled first. For semantic context detection, which is viewed as a pattern recognition problem, generative (HMM) and discriminative (SVM) models are applied in fusing the information obtained from audio event detections. We perform some experiments to show the effectiveness of the proposed framework and compare these two fusion schemes. The results of semantic context detection can be applied to multimedia indexing and retrieval to facilitate efficient media access.

The remainder of this paper is organized as follows. Section 2 describes the definitions of audio event and semantic context and states the concept of hierarchical audio models. The audio features we used for event modeling are briefly introduced in Section 3. After extracting these features, HMMs are used to model audio events in Section 4. In Sections 5 and 6, issues on fusion schemes based on HMM and SVM are addressed, respectively. Performance evaluation and some discussions are shown in Section 7, and the concluding remarks are given in Section 8.

2. Hierarchical audio models

The semantic indexing process is performed in a hierarchical manner. Two stages of models, i.e. audio event and semantic context modelings, are constructed to hierarchically characterize audio clips.

2.1 Audio event and semantic context

Audio events are defined as short audio segments which represent the sound of an object or an event. They can be characterized as different patterns according to the statistics of several audio features. In this paper, we aim at indexing multimedia documents by detecting high-level semantic contexts. Therefore, the occurrences of *gunshot* and *explosion* events are used to characterize '*gunplay*' scenes. The occurrences of *engine* and *car-braking* events are used to characterize '*car-chasing*' scenes.

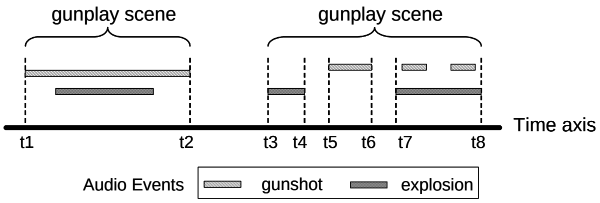
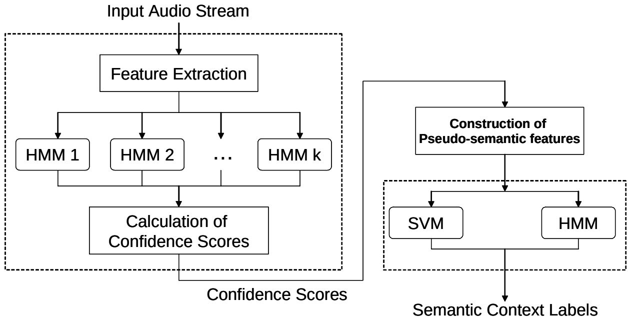


Figure 1. Examples of audio semantic contexts

A semantic context is a meaningful scene that may or may not contain all audio events at every time instant and no specific evolution pattern with these events. For example, in a gunplay scene, we cannot expect that explosions always occur after gunshots. Moreover, there may be some silence shots that contain no relevant audio events, but they are viewed as parts of the same gunplay scene in human's sense. Figure 1 illustrates the idea of semantic contexts of gunplay scenes. The audio clip from t1 to t2 is a typical gunplay scene, which contains mixed relevant audio events. In contrast to this case, no relevant event exists from t4 to t5 and from t6 to t7. However, the whole audio segment from t3 to t8 is viewed as the same scene in users' sense, as long as the duration of the ‘irrelevant clip’ doesn’t exceed users’ tolerance. Therefore, to model the characteristics of semantic contexts, we develop an approach which takes a series of events along the time axis into account rather than just the information at a time instant.



(a) Audio Event Modeling (b) Semantic Context Modeling

Figure 2. The proposed hierarchical framework contains (a) audio event modeling and (b) semantic context modeling.

2.2 Hierarchical framework

The proposed framework consists of two stages: audio event modeling and semantic context modeling. First, as shown in Figure 2(a), the input audio stream is divided into overlapped segments, and several audio features are extracted from each segment. The extracted features are then input to each HMM module. Through the Forward algorithm [18], the log-likelihood of an audio segment with respect to each audio event is computed. To determine how a segment is close to an audio event, a confidence metric based on the likelihood ratio test is defined. We say that the segments with high confidence scores from the gunshot audio event model, for example, imply high probability of the occurrence of gunshot sounds.

In the stage of semantic context modeling/detection, the confidence values from event detection constitute the cues for characterizing high-level semantic contexts. For an audio clip, the confidence values obtained from each audio event model are concatenated as feature vectors, which represent the characteristics of the audio clip with respect to several audio events. We call them *pseudo-semantic features* because they represent the interrelationship of several audio events, which are grounds for users to realize what the segment presents. With these features, two modeling approaches are used and investigated to perform semantic context modeling, as shown in Figure 2(b). As the usage in pattern recognition and data classification, SVM and HMM shed lights on clustering these pseudo-semantic features and facilitate detection processes.

3. Feature extraction

One of the important factors for elaborating pattern recognition is feature selection for adequately characterizing original data. For analyzing audio sequences, several audio features are extracted and utilized, including volume, band energy ratio, zero-crossing rate, frequency centroid, bandwidth, and mel-frequency cepstral coefficient (MFCC) [11]. They are shown beneficial for audio analysis and are widely applied [5,6,7,16].

When extracting audio features, all audio streams are down-sampled to 16 KHz, 16 bits and mono-channel format. A 1-sec sliding window moves through the input audio with 50% overlapping. The signal in each sliding window is further divided into overlapped frames which are 25-ms long with overlapping ratio 0.5 for feature extraction. Among them, volume and zero-crossing rate are calculated directly from audio signal amplitude. After Fourier transformation, frequency centroid and bandwidth are calculated to present the first- and second-order statistics of the spectrogram. The frequency spectrum is divided into four sub-bands with equal frequency intervals, then the band-energy ratio is computed to show the energy distributions in different bands. An 8-order MFCC is computed as the inverse Fourier transform of the logarithmic spectrum in a frame. Finally, a 16-dimensional (16-D) feature vector is constructed. Details of the audio feature extraction processes can be found in [11] and [3]. Furthermore, the temporal variations of the adopted features are also considered. That is, the differences of the features between adjacent frames are calculated and are combined with the original features. In our system, a 32-dimensional (32-D) feature vector is generated for each audio frame.

4. Audio events modeling

Detecting events in audio tracks is crucial to multimedia analysis. This section addresses some issues of audio event modeling, including the determination of model size, model training process, and the process for constructing pseudo-semantic features from detection results.

4.1 Model size estimation

We use HMMs to describe the characteristics of audio events. The 32-D feature vectors from a type of audio event are segmented into several sets, with each set denoting one kind of timbre, and modeled later by one state of an HMM. Determination of model size is crucial in applying HMMs. The state number should be large enough to characterize the variations of features, while it should also be compact when we consider computational cost of model training process. In this work, adaptive sample set construction technique [14] is adopted to estimate a reasonable model size of each audio event. Through this process, the state number is set as two for *car-braking*, four for *engine*, and six for *gunshot* and *explosion* sounds. These results make sense because we elaborately collect various kinds of training sounds for each audio event, and these numbers represent the degree of variations of different audio events. For example, the sounds of rifle and hand/machine gun are all collected as the *gunshot* training data. They vary significantly and should be represented by larger state numbers than that of simple sounds, such as the sharp but simple *car-braking* sounds.

4.2 Model training

For each audio event, 100 short audio clips each with length 3-10 seconds are selected as the training data. In the training stage, the features described in Section 3 are first extracted from each audio frame. Based on these feature vectors, a complete specification of HMM, which includes two model parameters (model size and number of mixtures in each state) and three sets of probabilities (initial probability, observation probability, and transition probability), are determined. The model size and initial probability could be decided by the clustering algorithm described in the previous subsection, and the number of mixtures in each state is empirically set as four. The Baum-Welch algorithm is then applied to estimate the transition probabilities between states and the observation probabilities in each state. Finally, four HMMs are constructed for the audio events we concern. Details of the HMM training process will be further described in Section 5, where HMMs are also used for semantic context modeling.

4.3 Pseudo-Semantic Features

After audio event modeling, for a given audio segment (the time unit for calculating log-likelihood values is 1 sec, the length of the sliding window defined in Section 3), the log-likelihood values with respect to each audio event are calculated by the Forward algorithm. However, unlike audio classification, we cannot simply classify an audio segment as a specific event even if it has the largest log-likelihood score. An audio segment may just present general environmental sound and doesn't present any predefined audio event. Therefore, to evaluate how likely an audio segment belongs or not belongs to a specific audio event, an approach based on the concept of *likelihood ratio test* [1] is applied.

For each type of audio event, two log-likelihood functions are constructed from the log-likelihood values. The first function $f_i(x\,|\,\theta_1)$ represents the distribution of the log-likelihood values obtained from a specific audio event model *i* with respect to the corresponding audio sounds. For example, from the *engine* model with the set of *engine* sounds as inputs, the resulting log-likelihood values are gathered to form the distribution. Figure 3(a) illustrates this construction process, and we call this distribution as the '*within distribution*' of the engine event. In contrast, the second function $f_i(x\,|\,\theta_0)$ represents the distribution of the log-likelihood values obtained from a specific audio event model with respect to other audio sounds. Like the previous example, the '*outside distribution*' of the *engine* event is constructed from the log-likelihood values gathered from the *engine* model with the sets of *gun*, *explosion*, and *car-braking* sounds as inputs. These two distributions show how log-likelihood values vary with respect to within and outside a specific audio event and help us for discriminating a specific audio event from others.

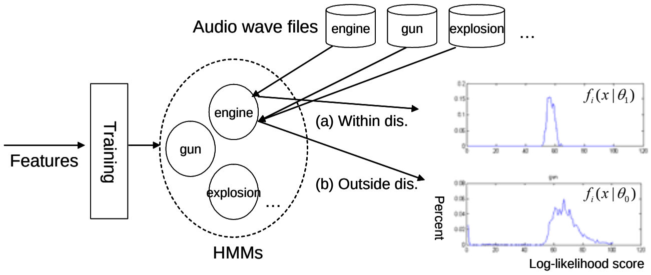


Figure 3. Calculation of (a) within distribution and (b) outside distribution for audio event *i*.

In the process of confidence evaluation, the audio segments with low average volume and zero-crossing rate are first marked as silence and the corresponding confidence values with respect to all audio events are set to be zero. For non-silence segments, the extracted feature vectors are input to the four HMMs, and the corresponding log-likelihood values are calculated. For a given audio segment, assume that the log-likelihood value from an event model is *x*, its confidence score with respect to audio event *i* is defined as:

$$sc_i(x) = \frac{f_i(x\,|\,\theta_1)}{f_i(x\,|\,\theta_0)}\,. \qquad (1)$$

By the definition in Section 2.1, a semantic context often lasts for at least a period of time, and not all the relevant audio events exist at every time instant. Therefore, the confidence scores of several consecutive audio segments are considered integrally to capture the temporal characteristics in a time series [7]. We define a *texture window* (c.f. Figure 4(b)) of 5-sec long, with 2.5-sec overlaps, to go through the confidence values of 1-sec audio segments (i.e. the *analysis windows*). The means of confidence values in each texture window are then calculated to be the input of semantic context modeling. For each texture window, the mean values of confidence scores are calculated

$$m_i = mean(sc_{i,1}, sc_{i,2}, ..., sc_{i,N}) \cdot \quad (2)$$

where $sc_{i,j}$ denotes the confidence score of the j-th analysis window with respect to event i, and N denotes the total number of analysis windows in a texture window. The pseudo-semantic feature vector p_t for the t-th texture window is defined as

$$p_t = [m_1, m_2, ..., m_k] \cdot \quad (3)$$

The total pseudo-semantic features P is

$$P = p_1;p_2;\ldots;p_{T.} \quad (4)$$

T is the total number of texture windows in the audio clip. By the settings described above, nine analysis windows, with 0.5 overlapping ratio, construct a texture window. The number of audio events k is four in this work.

We call the features formed by confidence scores as pseudo-semantic features because they represent the intermediate characteristics between low-level physical audio features and high-level semantic contexts. The audio segments with higher confidence scores in the audio events relevant to a semantic context are more likely to convey this semantics. For example, the audio segments with higher confidence scores in gunshot and explosion events somehow drop hints on the occurrence of gunplay scenes. In this work, we investigate generative and discriminative approaches to model the pseudo-semantic features and characterize two semantic contexts: the gunplay and car-chasing scenes. HMM is selected to be the instance of generative approach, and SVM is treated as the instance of discriminative approach.

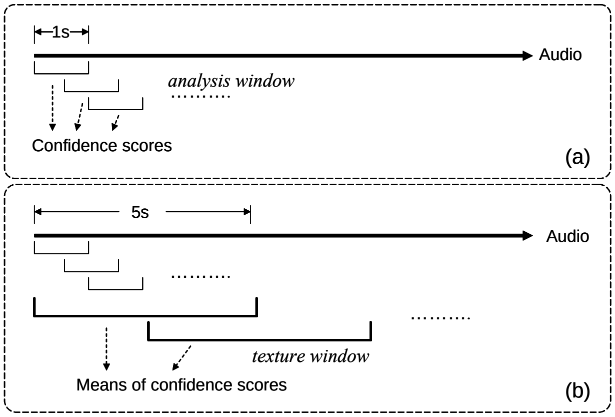


Figure 4. Pseudo-semantic features calculation for semantic contexts modeling: (a) analysis windows and (b) texture windows.

5. HMM for semantic context

For describing a sophisticated semantic context, a general model, e.g. Gaussian mixture model, that only covers the event data distributions is not affordable. It is preferable to explicitly model the time duration density by introducing the concept of state transition. The confidence scores of relevant events don't remain the same at every time instant. There would be some segments with low confidence scores because the sound effect is unapparent or is influenced by other environmental sounds. On the other hand, some segments may pose higher confidence because the audio events raise or explosively emerge. A model with more descriptive capability should take the temporal variations into consideration.

HMM is widely applied in speech recognition to model the spectral variation of acoustic features in time. It captures the time variation and state transition duration from training data and provides different likelihood values by giving different test data. In speech-related applications, the left-right HMMs, which only allow state index increasing (or staying the same) as time goes by, are considered to be suitable. But in the case of semantic context modeling, there is no specific consequence which formally represents the time evolution. Therefore, ergodic HMMs, or the so-called fully connected HMMs, are used in our work.

5.1 Model training

The confidence scores for some specific audio events, which are highly relevant to a specific semantic concept, are collected and modeled for conducting the high-level semantic context detection. To perform model training, six gunplay and car-chasing scenes, each with length 3-5 minutes, are manually selected from several Hollywood action movies as the training data.

For each semantic context, the parameters of an HMM can be estimated by using the Expectation-Maximization (EM) strategy or the so-called Baum-Welch algorithm. The state number N and number of distinct observation symbols M are set as four in this work. After the training process, parameters of two ergodic HMMs, which respectively represent the gunplay and the car-chasing scenes, are estimated. These models elaborately characterize the densities of time-variant features and present the structures of sophisticated semantic contexts.

5.2 Semantic context detection

The semantic context detection process is conducted following the same idea as that of the audio event detection. For every 5-sec audio segment (a texture window), the log-likelihood calculated by the Forward algorithm represents how the semantic context models match the given pseudo-semantic features. The binary indicator $\alpha_{s,t}$ is defined to show the appearance of semantic context s at the t-th texture window, s = 1 or 2 for gunplay or car-chasing scenes. That is,

If $\sigma_s > \varepsilon$, $\alpha_{s,t} = 1$. Otherwise, $\alpha_{s,t} = 1$, (5) where σ_s is the log-likelihood value under semantic context model s, and ε is a pre-defined threshold for filtering out those texture windows with too small values. The threshold can be adjusted by the user to tradeoff the precision and recall of semantic context detection.

6. SVM for semantic context

SVM has been shown to be a powerful discriminative technique [8]. It focuses on structural risk minimization by maximizing the decision margin. The goal of SVM is to produce a model which predicts target value of data instances in the testing set. In our work, by giving the pseudo-semantic features, we exploit SVM classifiers to distinguish the textures of 'gunplay', 'car-chasing', and 'others' scenes. Although the features obtained from the same semantic context may disperse variably in the feature space (which is caused by the various patterns of the same semantic context), the SVM classifier which maps features into a higher dimensional space and finds a linear hyperplane with the maximal margin can effectively distinguish one semantic context from others.

Note that SVMs were originally designed for binary classification. In our work, we should classify a segment into gunplay, car-chasing, or other scenes, thus the SVM classifiers should be extended for multiclass classification both in training and testing processes.

6.1 Model Training

Recently, a few researches are conducted on reducing a multiclass SVM into several binary SVM classifiers [17]. According to the performance analysis of multiclass SVM classifiers [4], we adopt the 'one-against-one' strategy to model these three scenes. Three SVM models are constructed, i.e. 'gunplay vs. car-chasing', 'gunplay vs. others', and 'car-chasing vs. others'. For training each model, given a training set of instance-label pairs (x_i, y_i), where $x_i \in R^n$ and $y_i \in \{1, -1\}$, a SVM finds the solution of the following optimization problem:

$$\min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \qquad (6)$$

subject to $y_i\left(\omega^T\phi(x_i)+b\right) \geq 1-\xi_i$, $\xi_i \geq 0$.

The training data x_i are mapped to a higher dimensional space by the function ϕ and C is the penalty parameter of the error term. In model training, the kernel function $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ we used is the radical basis function (RBF), which is suggested in many SVM-based researches. That is, our kernel function is

$$K(x,y) = \exp(-\gamma \left\| x_i - x_j \right\|^2), \ \gamma > 0. \qquad (7)$$

It is crucial to find the right parameters C and γ in RBF. We apply five-fold cross validation with a grid search of varying (C,γ) on the training set to find the best parameters achieving the highest classification accuracy.

For training one SVM model, the pseudo-semantic features obtained from four audio events are labeled in the unit of a texture window. Then all labeled texture windows are collected together to produce the training data. Three binary SVM classifiers will be combined later to identify what semantic context a texture window belongs to.

6.2 Semantic Context Detection

In semantic context detection, the Decision Directed Acyclic Graph SVM algorithm (DAGSVM) [17] is applied to combine the results of one-vs-one SVMs. Figure 5 illustrates one example of the detection procedure. Initially, the test vector from a texture window is viewed as the candidates for all three semantics. In the first step of detection, the test vector is input to the root SVM classifier, i.e. 'car-chasing vs. others' classifier. After this evaluation, the process branches to left if more vectors are predicted as 'others' segments, and the car-chasing semantics is removed from the candidate list. The 'gunplay vs. others' classifier is then used to re-evaluate the testing vector. After these two steps, the vector which represents the characteristic of a texture window is labeled as 'gunplay' or 'others'.

The DAGSVM separates the individual classes with large margins. It is safe to discard the losing class at each one-vs-one decision because, for the hard margin case, all of the examples of the losing class are far away from the decision surface. Hence, the choice of the class order in detection procedure is arbitrary.

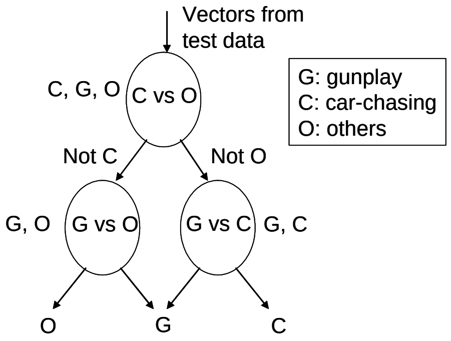


Figure 5. The testing procedure of DAGSVM

7. Performance evaluation

The training data for audio event and semantic context modeling are manually selected from Hollywood movies. Note that the criteria of selecting training data for audio events and semantic contexts are different. For semantic context modeling, we collected the *gunplay* and *car-chasing* scenes based on the experienced users' subjective judgments, no matter how many relevant audio events exist in the scene. On the contrary, the training data for audio event modeling are many short audio segments that are exactly the audio events. For each audio event, 100 short audio clips are manually extracted from movies as training data. The total audio data for detecting each semantic context are about one hour long. They are manually labeled and are divided into two parts each with 30 minutes for training and testing.

7.1 Evaluation of audio event detection

In audio event detection, a 'correct detection' is declared if the audio segment within an analysis window is evaluated as an audio event and its corresponding confidence score is larger than a predefined threshold. The overall detection performance is listed in Table 1. The average recall is over 70% and the average precision is about 85%. Although the detection accuracy is often sequence-dependent and is affected by confused audio effects, the reported performance shows the applicability of our approach in conducting effective semantic context modeling. In addition, different audio events have different evaluation results. Because the car-braking sounds are often very short in time (less than one second, which is the length of one basic analysis unit defined in our work) and are mixed with other environment sounds, the detection accuracy is particularly worse than others. This situation is different for gunshot sounds because there is often a continuity of gunshots (the sounds of a machine gun or successive handgun/rifle shoots) in a gunplay scene. Nevertheless, because the car-braking sound is a representative audio cue of car-chasing scenes, we still take the detection results of car-braking sounds into account in car-chasing context modeling.

Table 1. Overall performance of audio event detection

Audio Event	Recall	Precision
Gun	0.938	0.95
Explosion	0.786	0.917
Brake	0.327	0.571
Engine	0.890	0.951
Average	0.735	0.847

7.2 Evaluation of semantic context detection

In the semantic context detection, the models based on HMM and SVM are evaluated respectively. As the basic analysis unit is one texture window, the metrics of recall, precision, and false alarm rate are calculated to show the detection performance.

We tested six 5-min movie segments (selected from 'We Were Soldiers', 'Windtalker', 'The Recruit', and 'Band of Brother') for gunplay and seven 5-min movie segments (selected from 'Terminator 3', 'Ballistic: Ecks vs. Sever', 'The Rock', and '2 Fast 2 Furious') for car-chasing. The detection performance is somewhat sequence-dependent because different movies posses different essential characteristics. As shown in Table 2, the HMM-based approach generally achieves over 95% recall and near 80% precision in detecting both semantic contexts, while the SVM-based approach achieves about 75% recall and over 80% precision. These results show a promising achievement of the proposed fusion schemes.

The accuracy of semantic context detection would degrade when bad audio event detection is involved. For example, in Table 2, the detection accuracy and false alarm rate in test clips 4, 12, and 13 are relatively worse than that of the others, in both two fusion schemes. This degradation is caused by mixed audio sounds or confused acoustic characteristics between different sounds. One example is the simultaneous occurrence of gunshots and explosions, while the bass environmental sound may be misclassified as an engine event because of their acoustic similarity. Furthermore, the performance variation in car-chasing detection is generally larger than that of the gunplay detection. This trend also verifies the influences of different event detection accuracies described in Section 7.1, where the car-braking event detection doesn't work as well as others.

7.3 Discussion

In comparison with the HMM-based and SVM-based approaches, we observed that the HMM-based scheme produces better recall rate, while the SVM-based scheme provides better precision rate. This result confirms the essential difference between generative and discriminative models. Because the discriminative approach directly models the decision boundary, it has better capability to exactly separate two sets of data that have different distributions. In addition, the model training process finds the hyperplane which maximizes the decision margin, and thus the discriminative approach has better precision in identifying a test vector. However, for the same reason, the discriminative approach is sensitive to noise data, which may be caused by the intermission of events or event detection errors. Like the duration from t4 to t5 in Figure 1, if the intermission duration is larger than the length of a texture window, or the mean confidences don't show the apparent tendency towards some semantic contexts, the corresponding texture window may be misclassified. In the contrast, the generative approach computes the posteriori probability of a texture window with respect to two semantic contexts, and the result is flexible rather than just saying “yes” or “no”. Hence, the HMM-based approach works better in recall rate.

Table 2. Performance of semantic context detection by (a) HMM and (b) SVM.

Semantic Context		Rcl. (a)	Pr. (a)	FA.	Rcl. (b)	Pr. (b)	FA.
Gunplay	Cp1	1	0.706	0.294	0.792	0.755	0.245
	Cp2	1	0.797	0.203	0.847	0.877	0.123
	Cp3	1	0.873	0.127	0.823	0.859	0.141
	Cp4	0.917	0.611	0.389	0.667	0.696	0.304
	Cp5	1	0.932	0.068	0.794	0.857	0.143
	Cp6	1	0.850	0.150	0.648	0.833	0.167
	Avg.	0.986	0.795	0.205	0.762	0.813	0.187
Car-chasing	Cp7	0.952	0.919	0.081	0.795	1	0
	Cp8	0.968	0.821	0.179	0.600	0.934	0.066
	Cp9	1	0.902	0.098	0.667	0.949	0.051
	Cp10	1	0.815	0.185	0.545	0.686	0.314
	Cp11	1	0.755	0.245	0.950	0.884	0.116
	Cp12	1	0.5	0.5	0.771	0.607	0.393
	Cp13	1	0.605	0.395	0.806	0.773	0.227
	Avg.	0.989	0.760	0.240	0.734	0.833	0.167

8. Conclusion

We presented a hierarchical approach that bridges the gaps between low-level features and high-level semantics and facilitates semantic indexing in action movies. The proposed framework hierarchically conducts modeling and detection at two levels: audio event and semantic context. After careful selection of audio features, HMMs are applied to model the characteristics of audio events. At the semantic context level, generative (HMM) and discriminative (SVM) approaches are used to fuse pseudo-semantic features obtained from the results of event detection and to model semantic contexts. Experimental results demonstrate a remarkable performance of the fusion schemes and signify that the proposed framework draws a sketch for constructing an efficient semantic indexing system.

The proposed framework can be extended to other types of videos. It may be necessary to consider different combinations of events or include visual information according to the production rules of targeted videos. Another possible improvement may include the elaborate feature selection from a candidate pool by developing an automatic feature induction mechanism.

9. Acknowledge

This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC92-2622-E-002-002, NSC92-2213-E-002-023 and 89E-FA06-2-4-8.

10. References

[1] R.O. Duda, P.E. Hart, D.G. Stork. *Pattern Classification.* John Wiley & Sons, 2001.

[2] M.R. Naphade, and T.S. Huang. Extracting semantics from audiovisual content: the final frontier in multimedia retrieval. *IEEE Trans. on Neural Network, vol. 13, no. 4,* 2002, 793-810.

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

[4] C.W. Hsu and C.J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks, vol. 13, no. 2,* 2002, 415-425.

[5] L. Lu, H.J. Zhang, and H. Jiang. Content analysis for audio classification and segmentation. *IEEE Trans. on Speech and Audio Processing, vol. 10, no. 7,* 2002, 504-516.

[6] T. Zhang and C.C.J. Kuo. Hierarchical system for content-based audio classification and retrieval. *Proceedings of SPIE, Multimedia Storage and Archiving Systems III, vol. 3527,* 1998, 398-409.

[7] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing, vol. 10, no. 5,* 2002, 293-302.

[8] V.N. Vapnik *Statistical Learning Theory.* Wiley, New York, 1998.

[9] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. *Proceedings of ACM MM,* 1995, 295-304.

[10] Z. Liu, J. Huang, and Y. Wang. Classification of TV programs based on audio information using hidden Markov model. *Proceedings of IEEE Workshop on Multimedia Signal Processing,* 1998, 27-32.

[11] Y. Wang, Z. Liu, and J.C. Huang. Multimedia Content Analysis. *IEEE Signal Processing Magazine,* Nov., 2000, 12-36.

[12] H. Zettl. Sight Sound Motion: *Applied Media Aesthetics.* Belmont, CA: Wadsworth Publishing, 1999.

[13] C. Dorai and S. Venkatesh. *Media Computing: Computational Media Aesthetics.* Kluwer Academic Publishers, 2002.

[14] S.T. Bow. *Pattern Recognition and Image Preprocessing.* Marcel Dekker, 2002.

[15] S. Moncrieff, S. Venkatesh, and C. Dorai. Horror Film Genre Typing and Scene Labeling via Audio Analysis. *Proceedings of ICME, vol. 2,* 2003, 193-196.

[16] R. Cai, L. Lu, H.J. Zhang, and L.H. Cai. Highlight sound effects detection in audio stream. *Proceedings of ICME, vol. 3,* 2003, 37-40.

[17] J.C. Platt, N. Cristianini, and J. Shawe-Taylor, Large margin DAGs for multiclass classification. *Advances in Neural Information Processing Systems,* Cambridge, MA: MIT Press, vol. 12, 2000, 547-553.

[18] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, vol. 77, no. 2,* 1989, 257-286.