http://dx.doi.org/10.3745/KIPSTB.2012.19B.1.019

서포트 벡터 머신과 퍼지 클러스터링 기법을 이용한 오디오 분할 및 분류

Ngoc Nguyen[†] · 강 명 수[††] · 김 철 홍[†††] · 김 종 면[††††]

요 약

최근 멀티미디어 정보가 급증함에 따라 콘텐츠 관리에 대한 요구도 함께 증가되고 있다. 이에 오디오 분할 및 분류는 멀티미디어 콘텐츠를 효과적으로 관리할 수 있는 대안이 될 수 있다. 따라서 본 논문에서는 동영상에서 취득한 오디오 신호를 분할하고, 분할된 오디오 신호를 음악, 음성, 배경 음악이 포함된 음성, 잡음이 포함된 음성, 묵음(silence)으로 분류하는 정확도가 높은 오디오 분할 및 분류 알고리즘을 제안한다. 제안하는 알고리즘은 오디오 분할을 위해 서포트 벡터 머신(support vector machine, SVM)을 이용하였다. 오디오 신호의 분류를 위해서는 분할된 오디오 신호의 특징을 추출하고 이를 퍼지 클러스터링 알고리즘(fuzzy c-means, FCM)의 입력으로 사용하여 각 계층으로 오디오 신호를 분류하였다. 제안하는 알고리즘의 평가는 분할과 분류에 대해 각각 그 성능을 평가하였으며, 분할 성능 평가는 정확도율(precesion rate)과 오차율(recall rate)을 이용하였으며, 분류 성능 평가는 정확성(classification accuracy)을 사용하였다. 또한 오디오 분할의 경우는 이진 분류기와 퍼지 클러스터링을 이용한 기존의 알고리즘과 그 성능을 비교하였다. 모의 실험 결과, 제안한 알고리즘의 분류 성능이 기존 알고리즘 보다 정확도율과 오차율 면에서 모두 우수하였다.

키워드 : 오디오 분할, 오디오 분류, 서포트 벡터 머신, 특징 추출, 퍼지 클러스터링

Audio Segmentation and Classification Using Support Vector Machine and Fuzzy C-Means Clustering Techniques

Ngoc Nguyen[†] · Myeongsu Kang[††] · Cheol-Hong Kim[†††] · Jong-Myon Kim[††††]

ABSTRACT

The rapid increase of information imposes new demands of content management. The purpose of automatic audio segmentation and classification is to meet the rising need for efficient content management. With this reason, this paper proposes a high-accuracy algorithm that segments audio signals and classifies them into different classes such as speech, music, silence, and environment sounds. The proposed algorithm utilizes support vector machine (SVM) to detect audio-cuts, which are boundaries between different kinds of sounds using the parameter sequence. We then extract feature vectors that are composed of statistical data and they are used as an input of fuzzy c-means (FCM) classifier to partition audio-segments into different classes. To evaluate segmentation and classification performance of the proposed SVM-FCM based algorithm, we consider precision and recall rates for segmentation and classification accuracy for classification. Furthermore, we compare the proposed algorithm with other methods including binary and FCM classifiers in terms of segmentation performance. Experimental results show that the proposed algorithm outperforms other methods in both precision and recall rates.

Keywords : Audio Segmentation, Audio Classification, Support Vector Machine, Feature Extraction, Fuzzy C-means Clustering

1. Introduction

In the age of digital information, audio data has become an essential part in many modern applications. A typical multimedia database often contains millions of audio scenes that necessitate an automatic segmentation and classification method for efficient productions and managements. In general, audio contents analysis can be

※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-0017941).
† 준 회 원 : 울산대학교 컴퓨터공학과 석사과정
†† 준 회 원 : 울산대학교 컴퓨터정보통신공학과 박사과정
††† 종신회원 : 전남대학교 전자컴퓨터공학부 교수
†††† 정 회 원 : 울산대학교 컴퓨터정보통신공학부 교수(교신저자)
논문접수 : 2011년 5월 23일
수 정 일 : 1차 2011년 8월 8일
심사완료 : 2011년 9월 22일

performed in two steps. In the first step, audio stream is divided into many segments. In the second step, audio-segments are classified into different audio-classes such as speech, music, silence, and environment sounds. Therefore, audio segmentation is an important preprocessing step in audio classification systems.

Our intensive study has been conducted on audio segmentation by employing different features and methods. Recently, a number of techniques for automatic analysis of audio information have been proposed [1]. Some conventional audio segmentation methods employed threshold for audio features such as zero-crossing rate, energy in order to detect audio changes which are called boundaries or audio-cuts [2-8]. However, the accuracy of audio-cut detection might be decreased when audio signals recorded in noisy environment are segmented.

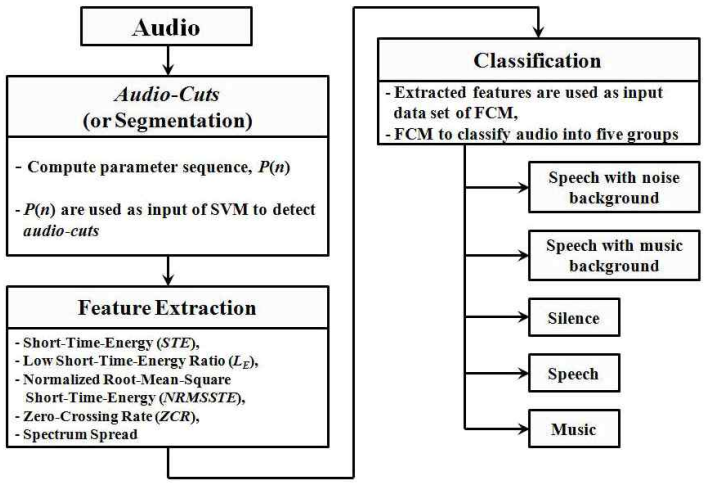
For audio classification, many researchers have conducted to enhance classification performance. In [5], audio materials are classified into speech, silence, laughter and non-speech sounds to segment discussion recordings in meetings using hidden Markov models. Recently, Li [6] presented a method using combinations of mel-frequency cepstral coefficients and other perceptual features including brightness, bandwidth, and subband energy. The author also presented the nearest feature line (NFL) classification, which contrasts with the nearest neighbor (NN) classification, and the NFL-based method produced consistently better results than the NN-based method. However, the accuracy of these algorithms in audio classification was not competitive (nearly 80%) because many audio signals in multimedia include audio effects and noise. To solve these drawbacks, some other methods utilized fuzzy c-means (FCM) as a classifier in order to eliminate audio effects such as fade-in and fade-out, attaining reasonable performances [8]. However, FCM has a standing difficulty that initializes membership values. In addition, FCM based on climbing searching methods can be trapped in local minimal values [9]. On the other hand, support vector machine (SVM) is a valid statistic learning method, and it has good capability to grasp the pattern property of low-level features from audio data [10].

Based on these advantages of the SVM classifier, this paper proposes a method to segment audio signals by employing SVM, and then classify audio segments using FCM. To evaluate segmentation and classification performance of the proposed algorithm, we utilize precision and recall rates for segmentation and classification accuracy for classification. Experimental results show that segmentation of audio signal using SVM outperforms other algorithms. This results in improving the accuracy of audio classification.

The rest of this paper is organized as follows. Section II introduces the proposed audio segmentation and classification algorithm. Section III evaluates segmentation and classification performance of the proposed algorithm. Section IV concludes this paper and suggests future researches.

2. Proposed Audio Segmentation and Classification Algorithm

This paper proposes a high-accuracy audio segmentation and classification algorithm, as shown in (Figure 1). The proposed algorithm is essentially composed of the following three main stages: *audio-cuts*, feature extraction, and classification.



(Figure 1) Block diagram for the proposed audio segmentation and classification algorithm

2.1 Target Audio Signals

In this study, target audio signals were captured from TV news program at 44.1 kHz, which were obtained from Ulsan broadcasting corporation.

2.2 Audio Segmentation

2.2.1 Detection of Audio-cuts

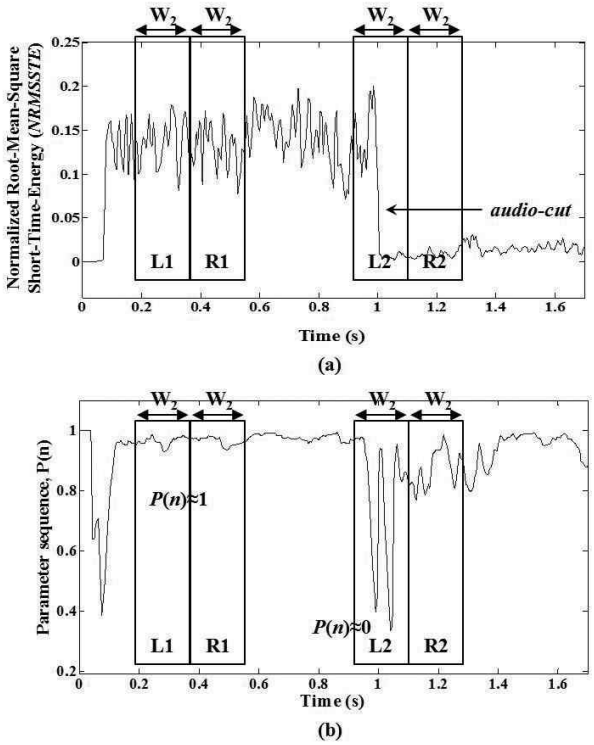
For accurate audio classification, it is firstly necessary to segment an audio signal into different audio signals at their boundaries, which are called *audio-cuts*, and it is also frame-based process with the window length of W_1. To do this, we compute the normalized root-mean-square short-time energy *(NRMSSTE)*of the audio signal as follows:

$$NRS(n) = \sqrt{\frac{1}{W_2}\sum_{m=0}^{W_2-1}\left[x(m)\right]^2}, \quad (1)$$

where $x(m)$ is the value of mth sample in a processing window with the length of W_2 in a frame in which the rectangular window is used. We then introduce the parameter sequence, $P(n)$, using the *NRMSSTE* sequence, $NRS(n)$. The parameter sequence, $P(n)$, is computed as follows:

$$P(n) = \frac{\sum_{m=0}^{W_2-1} NRS(n+m)\cdot NRS(n+m-W_1)}{\sqrt{\sum_{m=0}^{W_2-1}\left[NRS(n+m)\right]^2}\times\sqrt{\sum_{m=0}^{W_2-1}\left[NRS(n+m-W_1\right]^2}}. \quad (2)$$

The existence of the *audio-cut* can be detected by observing the parameter sequence $P(n)$. The sequence, $P(n)$, is computed by using the *NRMSSTE* values in two adjoining sliding windows that are depicted in (Figure 2). The case that an *audio-cut* exists in neither of the window is considered as follows: An example of this case is shown in the window L1 and R1. The *NRMSSTE* values in the both windows do not abruptly change, and consequently the numerator of $P(n)$ is close to its denominator; and then the value of $P(n)$ is close to 1. On contrast to this, the case that an *audio-cut* exists in either of the window is considered as follows: An example of this case is shown in the window L2, the *NRMSSTE* values in the window L2 abruptly change at the audio-cut, while the *NRMSSTE* values in the window R2 does not abruptly change. Thus, the numerator of $P(n)$ is much smaller than its denominator in the window L2, and therefore the sequence $P(n)$ is close to 0. This shows that the existence of the *audio-cut* can be detected by observing the sequence $P(n)$.



(Figure 2) Audio-cut versus non audio-cut. (a) RMS values in the processing windows, and (b) parameter sequence, P(n), in the processing windows

Finally, *audio-cuts* are detected by applying a support vector machine (SVM) to the parameter sequence $P(n)$ obtained from (3). To partition audio signal into two groups (*audio-cuts* and non *audio-cuts*), we define the following vector:

$$S(n-\Delta) = \left[P(n-\Delta),\ldots,P(n-\Delta+W_2-1)\right], \quad (3)$$

where Δ is a step size in the processing window, which is set to 10. The sequence $S(n-\Delta)$ is used as an input feature vector of SVM.

2.2.2 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning techniques introduced by V. Vapnik that analyze data and recognize patterns [11-12]. SVMs are applied to numerous fields of classification and

regression analysis, especially in pattern recognition [13]. The standard SVM is a non-probabilistic binary classifier. Given a set of training examples, each labeled as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. SVM is capable of learning in high-dimensional spaces, and can provide high performance with a limited training data set. Several techniques have been proposed to improve the classification performance and the time cost of SVMs [14-15]. The basic principle of the SVM classifier is the use of a hyper-plane to divide a given data set into two classes while maximizing the margin. However, in many real-world applications, there are a lot of data sets that are not linearly separable. There is likely no hyper-plane that can split the non-linear data sets into two classes. To deal with such cases, a soft margin SVM or a kernel function is used.

This paper employs SVM to partition an audio signal into two classes: *audio-cuts* and non *audio-cuts*. To apply SVM to audio segmentation, we utilize the Gaussian radial basis kernel function to map the input vector to a high-dimensional feature space. This is because SVM performs better with the Gaussian radial basis kernel than with other kernels [16]:

$$k\left(sv_i, sv_j\right) = \exp\left(-\frac{\left\|sv_i - sv_j\right\|^2}{2\delta^2}\right), \quad (4)$$

where $k(sv_i, sv_j)$ is the kernel function, sv_i, sv_j are the input feature vectors, and δ is a parameter set by the user and determines the width of the effective basis kernel function. If small δ values are used, overtraining occurs with the basis function wrapped tightly around the data points. In contrast, if large δ values are used, the basis function draws an oval around the points without defining the shape or pattern [16]. In this paper, the best performance occurred when the standard deviation, δ, was between 4 and 6. Thus, the value for δ was set to 5.5. In order to increase accuracy rate, accumulation of *audio-cuts* was used. In other words, only *audio-cut* point that occurred within one second continuously was considered a reliable *audio-cut*. Otherwise, it was declared as noise.

2.3 Feature Extraction

The next step for an automatic audio classification is to extract feature vectors that are composed of several features. Ideally, the feature vectors clearly separate all measured samples from different classes, but it is impossible. The purpose of the feature extraction is to obtain as much information as possible about the input audio. In this study, we utilize some feature such as short-time energy, low short-time energy ratio, root-mean-square, zero-crossing rate, and spectrum spread. The following sections explain how to extract features for audio classification.

2.3.1 Low Short-Time-Energy Ratio

Short-time energy (*STE*) is a simple feature that is widely used in various classification schemes, and it is defined to be the sum of a squared time domain sequence of data as follows:

$$STE(n) = \sum_{m=0}^{W_1-1} x^2(m), \quad (5)$$

where $w(n)$ is the rectangular with the length of W_1 and $x(m)$ is the value of the mth sample in the processing window. *STE* is generally suitable for discrimination between speech and music. This is because speech consists of words and mixed silence, and consequently it gives the higher *STE* value for speech than music.

Low short-time energy ratio (L_E) is defined as the ratio of number of frames whose *STE* value is below 0.5-fold average short-time energy in the processing window, as shown in (6).

$$L_E = \frac{1}{2N}\sum_{n=0}^{N-1}\left[\operatorname{sgn}\left(0.5\times\overline{STE} - STE\left(n\right)\right)+1\right], \quad (6)$$

where N is the total number of frames, n is the frame index, $STE(n)$ is the short-time energy at the nth frame, $\overline{STE}$ is the average *STE* in the processing window, and $sgn(\cdot)$ is 1 for positive arguments and 0 for negative arguments. L_E is also suitable for discrimination between speech and music signals. In general, the L_E value is high for speech. Although the L_E feature is effective for determining between speech and music, it is designed to recognize characteristics of single speaker speech so that it is possible to lose its effectiveness in circumstances that multiple speakers are considered.

2.3.2 Normalized RMS short-time energy(NRMSSTE)

The normalized root-mean-square short-time energy (*NRMSSTE*) value is a measurement of the energy in

the given signal, and it is defined to be the square root of the average of a squared signal, as shown in (1). To extract feature vector using the NRMSSTE value the processing window length is set to W_1. The mean and variation of the *NRMSSTE* values are able to classify between speech and music.

2.3.3 Zero-Crossing Rate

The zero-crossing-rate (*ZCR*) value is widely used in speech/music classification, and it is defined to be the number of zero-crossings within a processing window, as shown in (7).

$$ZCR(n)=\frac{1}{W_1-1}\sum_{m=0}^{W_1-1}\left|sgn\left[x(m)\right]-sgn\left[x(m-1)\right]\right|w(n-m), \quad (7)$$

where $x(m)$ is the value of the mth sample in the processing window, and $sgn(\cdot)$ is a sign function as mentioned in (6). In general, voiced and unvoiced speech sounds have low and high zero-crossing rates, respectively. This results in a high variation of *ZCR* whereas music typically has low variation of *ZCR*.

2.3.4 Spectrum Spread

Spectrum spread is a measure that signifies if the power spectrum is concentrated around the centroids or if it is spread out over the spectrum. Music consists of a broad mixture of frequencies whereas speech consists of a limited range of frequencies. Consequently, the spectrum spread is useful to determine between speech and music. Its mathematical definition is expressed by

$$SS(n)=\sqrt{\frac{\sum_{k=0}^{K-1}\left[\left(k-SC\right)^2\times\left|A\left(n,k\right)\right|^2\right]}{\sum_{k=0}^{K-1}\left|A\left(n,k\right)\right|^2}}, \quad (8)$$

where K is the order of the discrete Fourier transform (DFT), k is the frequency bin for the nth frame, *SC* is spectral centroid, and $A(n, k)$ is the DFT of the nth frame of a signal. Then, *SC* and $A(n, k)$ are calculated as follows:

$$SC(n)=\frac{\sum_{k=0}^{K-1}k\cdot\left|A(n,k)\right|^2}{\sum_{k=0}^{K-1}\left|A(n,k)\right|^2}. \quad (9)$$

$$A\left(n,k\right)=\left|\sum_{m=0}^{W_1-1}x\left(m\right)e^{-j\cdot\left(\frac{2\pi}{W_1}\right)\cdot k\cdot m}\right|. \quad (10)$$

2.4 Audio-Segments Classification

The proposed audio segmentation and classification algorithm includes audio-segment classification to partition an audio signal into the following five classes:

- **Silence**: An audio signal that only contains quasi-stationary background noise,
- **Speech**: An audio signal that contains the voices of human beings, such as the sound of conversation,
- **Music**: An audio signal that contains sounds made by musical instruments,
- **Speech with music background**: An audio signal that contains speech in an environment in which music exists in a background,
- **Speech with noise background**: An audio signal that contains speech in an environment in which noise exists in a background.

These five audio classes have the following features, which are utilized for audio-segment classification.

- The variance of *ZCR* of the audio signal (σ^2_{ZCR}),
- The mean of the *NRMSSTE* values of the audio signal (μ_{NRS}),
- The variance of the *NRMSSTE* values of the audio signal (σ^2_{NRS}),
- The L_E of the audio signal, and
- The mean of spectrum spread of the audio signal (μ_{SS}).

To accomplish audio-segment classification, this paper employs the well-known fuzzy c-means (FCM) that is well described in [17-18] in which the feature vectors are used as an input data set of FCM. The classification process using the FCM algorithm is described below.

$$X_f=\left[\sigma^2_{ZCR},\mu_{NRS},\sigma^2_{NRS},L_E,\mu_{SS}\right]. \quad (11)$$

- **Step 1**: Compute the number of group c and initialize centroids V(0) = $\{v_1^{(0)}, v_2^{(0)},\cdots, v_c^{(0)}\}$, where c is the number of classes.
- **Step 2**: Compute the membership values u_{ij} for each data element as follows:

$$u_{ij}=\left[\sum_{k=1}^{c}\left(\frac{d^2\left(x_j,v_i\right)}{d^2\left(x_j,v_k\right)}\right)^{\frac{1}{m-1}}\right]^{-1}. \quad (12)$$

• **Step 3**: Update the centroids v_i using (13),

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}}, \quad 1 \le i \le c. \tag{13}$$

where m is the degree of the fuzziness.

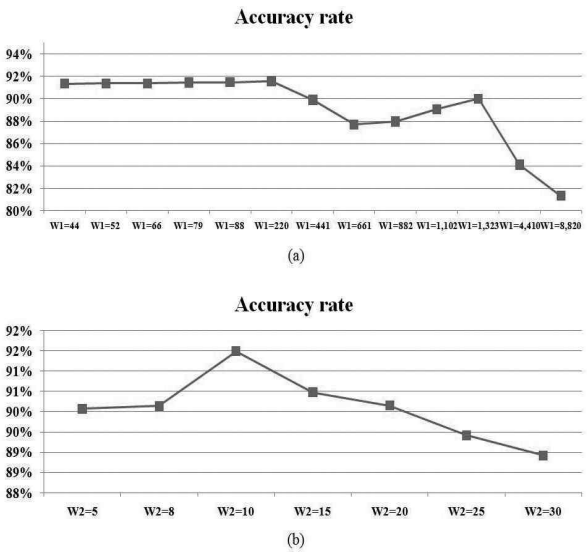
• **Step 4**: Evaluate the terminating condition $\max_{1\le i\le c}\left\{\left\|v_i^{(t)} - v_i^{(t-1)}\right\|\right\} < \varepsilon$ (where ||.|| is the Euclidean norm), where ε is termination threshold value. The iteration stops when it is satisfied, otherwise go to step 2.

• **Step 5**: Assign all audio segments to each cluster according to the corresponding maximum membership values.

3. Experimental Results

3.1 Window Lengths

The proposed segmentation and classification algorithm is a frame-based process, and therefore processing window lengths can affect segmentation and classification performance. However, there is no general consensus what lengths of processing windows give higher performance for them. To decide suitable window lengths, we conducted several experiments by changing window lengths W_1 in the range of 44 to 8,820 that is respective with the time from 0.001 to 0.2 seconds per frame and W_2 in the range of 5 to 30, and then we set the window lengths for W_1 and W_2 to 220 (0.005 seconds) and 10 (0.1 seconds), respectively. This is because they give the highest segmentation performance in terms of accuracy rate for segmentation described in (14). (Figure 3) shows accuracy rates according to window lengths W_1 and W_2.



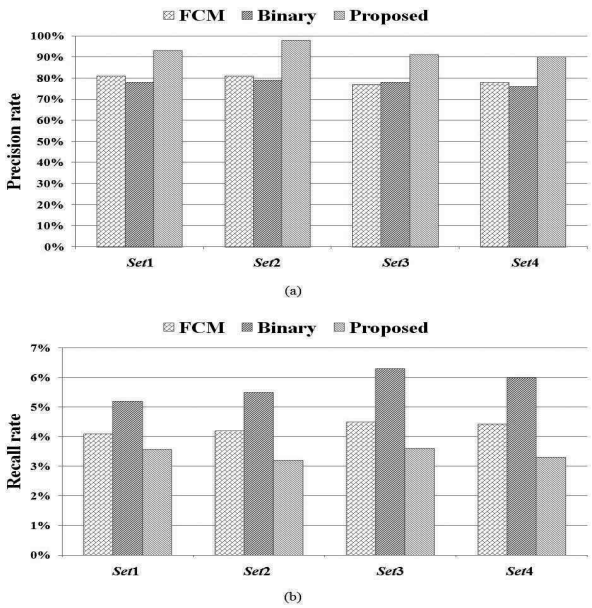
(Figure 3) Determining suitable window lengths with test data set. (a) window length W_1 to extract feature vectors, and (b) window length W_2 to compute parameter sequence $P(n)$

3.2 Audio Segmentation Results

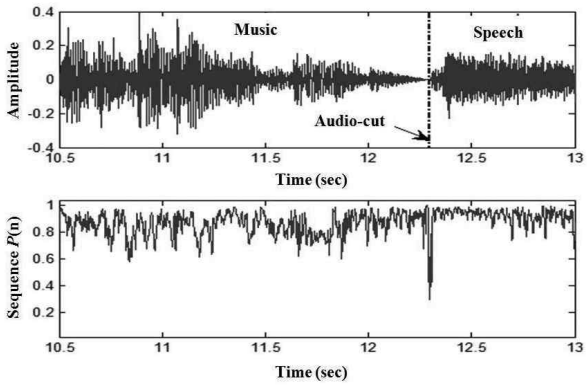
SVM is applied to detect audio-cuts using the sequence $P(n)$. To evaluate the segmentation performance, precision rate and error rate are considered as follows:

$$\text{Precision rate} = \frac{\text{Number of correctly detected audio-cuts}}{\text{Number of all audio-cuts}} \times 100\ (\%)$$
$$\text{Recall rate} = \frac{\text{Number of correctly detected audio-cuts}}{\text{Number of manually detected audio-cuts}} \times 100\ (\%). \tag{14}$$

Moreover, we compare the performance between the proposed method and other methods including binary [7] and FCM [8] classifiers in terms of precision rate and recall rate to show how much the proposed method improves the segmentation performance by applying the SVM technique. We have experimented for four sets of audio files that obtained from TV news program, and each set includes 14 to 16 15 minutes-long audio files that are sampled at 44.1 kHz. Set1 and Set2 are composed of speech signals of broadcasters in a studio, and Set1 and Set2 are different in contents. On the other hand, Set3 and Set4 consist of several parts of live shows or outside interviews. (Figure 4) presents the segmentation



(Figure 4) Comparisons of segmentation performance in terms of accuracy and error rates for different types of data sets. (a) accuracy rate, and (b) error rate



(Figure 5) Segmentation result using the proposed algorithm

performance of the proposed method and other methods. As shown in (Figure 4), the proposed method achieves the highest accuracy rates and the lowest error rates for segmentation (or *audio-cuts*). Especially, the accuracy rates of the proposed method yield around 95% for Set1 and Set2. This is because the separated regions of the target audio files in Set1 and Set2 are definite and short silences at transitions are also existed. Furthermore, the precision rates of the proposed algorithm are still higher than others for audio files in Set3 and Set4 which include different levels of noise signals. (Figure 5) shows that all *audio-cuts* are successfully detected using the proposed method. To detect 'audio-cut' position, this paper allows an error tolerance of ±100msec.

3.3 Audio Classification Results

To classify audio-segment classification, we employed the fuzzy c-means technique, and the degree of the fuzziness (m) and termination condition (ε) for FCM were set to 2 and standard deviation value of objective functions, respectively. For evaluation of classification, we utilized correctness that is widely accepted in recent researches. Classification accuracy of audio-segment classification is defined as follows:

$$\text{Classification accuracy} = \frac{\text{Number of correctly classified audio-segments}}{\text{Number of all classified audio-segments}} \times 100(\%). \quad (15)$$

For audio classification, we experimented with two target audio signals that consist of the following five classes: music (*Mus*), speech (*Spe*), speech with music (*Swm*), speech with noise (*Swn*), and silence (*Sli*). One is captured from TV news program, and another is captured from TV music program. <Table I and II> present classification results for each case.

〈Table I〉 Audio-segment classification results for TV news program

	Mus	*Spe*	*Swm*	*Swn*	*Sli*	Classification accuracy (%)
Mus	35	0	3	0	1	89.7
Spe	0	36	1	4	0	87.8
Swm	3	1	24	2	0	80.0
Swn	0	3	2	31	0	86.1
Sli	3	0	0	0	31	91.1

〈Table II〉 Audio-segment classification results for TV music program

	Mus	*Spe*	*Swm*	*Swn*	*Sli*	Classification accuracy (%)
Mus	44	0	2	0	1	93.6
Spe	0	5	1	0	0	83.3
Swm	1	0	15	2	0	83.3
Swn	0	0	0	0	0	100
Sli	0	0	0	0	5	100

As shown in <Table I and II>, misclassification results were mainly obtained between speech signals and speech with music signals. These were largely because the amplitudes of the music background were too small and unclear. In our simulation, drum and stringed instrument sounds were easily classified into the speech related classes (e.g., speech or speech with noise). Furthermore, misclassifications of speech signals occurred mostly when the segmented audio signals were too short. Therefore, it is necessary to find more effective feature extraction methods.

4. Conclusions and Future Works

To meet the rising need for efficient multimedia content management, this paper proposed a high-accuracy audio segmentation and classification algorithm for application in audio/video content analysis. To achieve higher segmentation and classification performances, we employed SVM in the segmentation process and FCM in the classification process, respectively. Experimental results show that the proposed audio analysis algorithm outperforms other methods in terms of precision and recall rates for segmentation, and classification accuracy for classification. Although the proposed algorithm achieves higher segmentation and classification performances than other methods, there are several works to be conducted in the future. Firstly, it is necessary to research on segmentation procedure to make it more robust to many kinds of situations and environments. Furthermore, we need to extract more efficient features to improve classification performance.

References

[1] J. Foote, "An Overview of Audio Information Retrieval," ACM Multimedia Systems, Vol.7, pp.2-10, 1999.

[2] Z. Liu and Y. Wang, "Audio Feature Extraction and Analysis for Scene Segmentation and Classification," J. VLSI Signal Processing, Vol.20, pp.61-79, 1998.

[3] L. Lu, H. J. Zhang, and H. Jiang, "Content Analysis for Audio Classification and Segmentation," IEEE Trans. Speech and Audio Processing, Vol.10, No.7, pp.504-516, 2002.

[4] C. Lin, S. H. Chen, and T. K. Truong, "Multiple Change-Point Audio Segmentation and Classification Using an MDL-based Gaussian Model," IEEE Trans. Speech and Audio Processing, Vol.14, No.2, pp.647-657, 2006.

[5] D. Kimber and L. Wilcox, "Acoustic Segmentation for Audio Browsers," in Proc. Interface Conf., pp.9-16, Sydney, 2006.

[6] S. Z. Li, "Content-based Audio Classification and Retrieval Using the Nearest Feature Line Method," IEEE Trans. Speech and Audio Processing, Vol.8, No.5, pp.619-625, 2000.

[7] T. Giannakopulos, A. Pikrakis, and S. Theodoridis, "A Novel Efficient Approach for Audio Segmentation," 19th Int'l Conf. Pattern Recognition, pp.1-4, 2008.

[8] N. Naoki, H. Miki, and K. Kideo, "Audio Signal Segmentation and Classification for Scene-cut Detection," IEEE Int'l Sym. Circuit and System, pp.4030-4033, 2005.

[9] Y. Shi and M. Mizumoto, "An Improvement of Neuro-fuzzy Learning Algorithm for Tuning Fuzzy Rules," Fuzzy Sets and Systems, Vol.1118, No.2, pp.33350, 2001.

[10] Y. Zhu, Z. Ming, and Q. Huang, "Automatic Audio Genre Classification Based on Support Vector Machine," 3th Int'l Conf. Natural Computation, Vol.1, pp.517-521, 2007.

[11] V. Vapnik, "Estimation of Dependences Based on Empirical Data," 1st Edition, Springer-Verlag, 1982.

[12] V. Vapnik, "Statistical Learning Theory," Springer, New York.

[13] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," J. Knowledge Discovery and Data Mining, Vol.2, pp.121-167, 1998.

[14] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Microsoft Research Technical Report MSR-TR-98-14, 1998.

[15] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge University Press, 2000.

[16] S. Gunn, "Support Vector Machines for Classification and Recognition," ISIS Tech. Rep., 1998.

[17] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal, "Fuzzy Models and Algorithms for Pattern Recognition and Image Processing," Kluwer Academic Publishers Norwell, MA, USA, 2005.

[18] S. Krinidis and V. Chatzis, "A Robust Fuzzy Local Information C-Means Clustering Algorithm," IEEE Trans. Image Processing, Vol.19, No.5, pp.1328-1337, 2010.

Ngoc Nguyen

e-mail : nguyenthungoc.dt2@gmail.com

2010년 하노이 과학기술대학교(학사)

2010년～현 재 울산대학교 컴퓨터공학과 석사과정

관심분야 : 고장진단, 음향신호처리, 임베디드시스템

강 명 수

e-mail : ilmareboy@ulsan.ac.kr

2008년 울산대학교 컴퓨터정보통신공학과 (학사)

2010년 울산대학교 컴퓨터정보통신공학과 (공학석사)

2010년～현 재 울산대학교 컴퓨터정보통신공학과 박사과정

관심분야 : 임베디드시스템, 음향신호처리, 멀티미디어응용, 워터마킹, 고장진단 등

김 철 홍

e-mail : cheolhong@gmail.com

1998년 서울대학교 컴퓨터공학과(학사)

2000년 서울대학교 컴퓨터공학부 (공학석사)

2006년 서울대학교 전기컴퓨터공학부 (공학박사)

2005년～2007년 삼성전자 반도체총괄 SYS.LSI사업부 책임연구원

2007년～현 재 전남대학교 전자컴퓨터공학부 교수

관심분야 : 임베디드시스템, 컴퓨터구조, SoC 설계, 저전력 설계 등

김 종 면

e-mail : jmkim07@ulsan.ac.kr

1995년 명지대학교 전기공학과(학사)

2000년 Electrical & Computer Engineering, University of Florida, USA(공학석사)

2005년 Electrical & Computer Engineering, Georgia Institute of Technology, USA(공학박사)

2005년～2007년 삼성종합기술원 전문연구원

2007년～현 재 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : 임베디드시스템, 시스템-온-칩, 컴퓨터구조, 병렬처리, 신호처리 등