다중 특징 벡터를 이용한 고속 오디오 검색

반지혜*, 김기만*, 박규식**
*한국해양대학교 전파공학과, **단국대학교 정보·컴퓨터학부 컴퓨터과학 전공

Quick Audio Retrieval Using Multiple Featrue Vector

Ban Ji-hye*, Kim Ki-man*, Park Kyu-sik**
*Korea Maritime University, **Dankook University
yaly815@hanmail.net, kimkim@mail.hhu.ac.kr, kspark@dankook.ac.kr

요약

최근 MPEG-7 등에서 컨텐츠 내용 기반 검색에 대한 연구가 이루어지고 있다. 내용 기반 검색은 기존의 키워드 기반 검색이 아닌 컨텐츠 내의 특징 벡터를 추출하여 이와 일치하는 것을 찾는 작업으로써 차세대 디지털 방송 등에 적용될 예정이다.

본 논문은 긴 오디오 stream에서 찾고자 하는 오디오의 위치를 빨리 찾을 수 있는 고속 검색 방법을 제시한다. 기존의 방법에서는 zero-crossing rate만을 이용하여 검색을 했었으나 본 논문에서는 오디오 신호의 특성을 표현할 수 있는 여러 가지 특징 벡터들을 이용한 고속 검색 방법을 고찰 한다. 본 논문의 가장 중요한 부분은 active search 알고리즘과 히스토그램, 그리고 적절하게 조합된 다중 특징 벡터들을 이용한 오디오 검색의 정확도와 속도를 향상시키는데 있다.

1. 서론

최근에 인터넷의 확산으로 다양한 멀티미디어 데이터는 정신없는 속도로 우리 삶을 지배하고 있다. 이런 상황 속에서 디지털 방송의 등장과 급속도로 증가하는 멀티미디어 데이터를 효과적으로 검색하기 위해서 MPEG-7 등에서 컨텐츠 내용 기반 검색에 대한 연구가 활발히 진행되고 있다.

MPEG-7에서 데이터 그 자체가 아닌 데이터의 내용에 대한 특징 표현 방법으로 검색하는 표준을 제시하고 있다. 예를 들어 비디오 부분에서 어떤 장면을 찾고 싶을 때 그 장면의 색과 texture등의 특성 정보를 이용해 검색을 하고, 오디오도 마찬가지로 노래의 특정 멜로디를 구분하여 검색할 수 있는 것이다. 이 기술은 컨텐츠 자체와 연결되어 사용자가 관심있는 멀티미디어 자료를 빠르고 효율적으로 찾을 수 있게 한다.

본 논문에서는 방송 오디오 데이터의 stream으로부터 찾고자 하는 사운드의 위치를 빨리 찾을 수 있는 고속 검색 방법을 제시한다. 기존에는 계산량 문제로 zero-crossing rate만을 이용하여 검색을 한 경우가 많으나, 본 논문에서는 오디오 신호의 여러 가지 특징 벡터들을 이용하여 검색을 한다. 제안된 방법의 효율성을 입증하기 위해 임의의 오디오 방송을 레코딩해서 특정 음악의 위치를 검색하는 작업을 수행하였다.

2. 내용 기반 검색 과정

오디오 데이터에 대한 내용기반 검색 방법에는 크게 브라우징과 인덱스를 통한 검색방법이 있다. 인덱스를 통한 검색 방법은 다시 오디오의 음향이나 음악 등을 분석하여 특징벡터로 인덱스를 만든 후 사용자가 멜로디나 음향 효과로 질의를 하여 원하는 곡을 찾는 방법과 오디오 내의 음성을 인식하여 키워드 기반의 인덱스를 만든 후 사용자가 질의를 음성이나 텍스트로 해주는 방법으로 구분할 수 있다.

본 논문에서는 전자의 방법으로 내용 기반 검색을 했다. 내용 기반 검색 과정은 먼저 멜로디로 질의를 하여 전체의 오디오로부터 추출한 특징 벡터와 입력한 멜로디의 특징 벡터간의 유사도를 측정하여 원하는 곡을 찾게 된다. 이 과정이 <그림1>에 나타나있다.

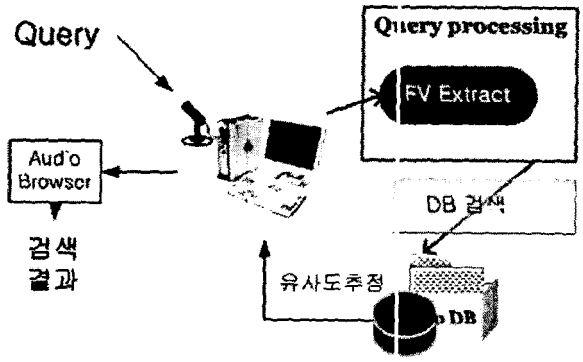

그림 1. 내용 기반 검색 과정

3. 다중 특징 벡터 구성

특징 벡터들의 추출은 오디오 신호의 부분의 특성을 수적으로 나타내기 위한 것이다. 이러한 특징 벡터들 중 STFT(Short Time Fourier Transform)이 기반을 둔 특징 벡터가 매우 일반적이고 계산을 빨리 할 수 있다는 이점을 지니고 있다. 그래서 본 논문에서는 STFT에 기반을 둔 특징 벡터들을 이용하였다[1].

3.1 Spectral Centroid

spectral centroid는 STFT의 magnitude 스펙트럼의 중심을 뜻한다.

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \quad (1)$$

여기서 $M_t[n]$은 프레임 t와 주파수 bin n에서 FT의 magnitude이다. centroid는 centroid는 스펙트럼의 형태의 측정방법중의 하나고 더 높은 centroid일 수록 더 높은 주파수에서 선명한 음질을 나타낸다[1].

3.2 Spectral Roll off

spectral roll off는 magnitude 분포의 85%가 집중해 있는 주파수 R_t이하를 말한다.

$$\sum_{n=1}^{R} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \quad (2)$$

roll off는 스펙트럼 형태의 또 다른 측정 방법이고 신호의 에너지고 너 낮은 신호에 얼마나 많이 집중되어 있는 가를 보여준다[1].

3.3 Spectral Flux

spectral flux는 연속된 스펙트럼의 분포의 normalize된 magnitude들의 차를 제곱한 것을 의미한다.

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \quad (3)$$

여기서 $N_t[n]$, $N_{t-1}[n]$은 각각 현재 프레임 t와 이전의 프레임 $t-1$에서 Fourier transform의 normalize된 magnitude이다[1].

4. 검색 알고리즘

<그림2>는 본 논문에서 사용한 검색 알고리즘이다[2]. 검색 방법은 먼저 reference 신호와 test 신호로부터 특징 벡터들을 추출한 후 각각의 특징벡터들을 히스토그램 모델링을 이용하여 템플릿으로 만든다. 그 다음에 테스트 오디오 stream에 reference 템플릿을 슬라이딩하면서 서로간의 유사도를 구하는 것이다. 유사도가 어떤 한계점을 넘어버리면 reference 사운드가 감지되어지고 그것의 위치를 찾을 수 있다.

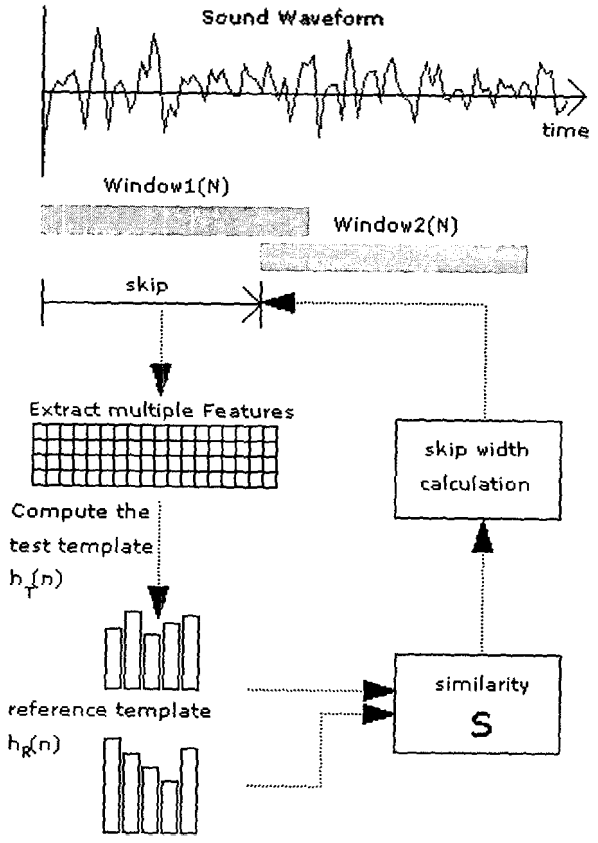

그림 2. 검색 알고리즘

4.1 히스토그램 모델링

특징벡터 $f(k)$는 다음과 같이 나타낼 수 있다.

$$f(k) = (f_1(k), f_2(k), f_3(k), \cdots, f_N(k)) \quad (4)$$

여기서 N은 한 프레임의 특징 벡터 수를 나타낸 것이고 k는 샘플 시간을 나타낸다.

ZCR은 기존의 논문에서 계산적으로 간단하기 때문에 많이 사용되어온 특징 중의 하나이다. 이것은 음악으로부터 음성의 실시간 인식에 적용되어진다. 샘플 n에 대한 i차 zero-crossing Z_i는 다음과 같이 정의되어진다.

$$Z_i = \sum_{n=1}^{N} \frac{|\, sgn(s_i(n)) - sgn(s_i(n-1))\,|}{2} \quad (5)$$

$$sgn(s_i(n)) = \begin{cases} 1 & s_i(n) > 0 \\ -1 & s_i(n) < 0 \end{cases}$$

본 논문에서는 기존의 방법과 달리 ZCR 외에도 다중 특징 벡터 구성을 위해 앞에서 제시한 STFT에 기반을 둔 특징 벡터들을 사용하였다.

각각의 프레임으로부터 특징 벡터를 추출한 후에 특징 벡터들의 분포를 알기 위해 히스토그램이 사용되어진다. test와 reference 템플릿 h_R과 h_T의 유사도는 히스토그램을 이용하여 다음과 같이 나타낼 수 있다[2].

$$S(h_R, h_T) = \sum_{i=1}^{B} min(h_R^i, h_T^i) \quad (6)$$

4.2 유사도 측정

각각의 시간마다 reference 템플릿을 슬라이딩해가면서 유사도 측정을 하는 방법이 가장 간단한 방법이기는 하지만 매시간 유사도를 측정하려면 계산량이 많아지게 된다.

test 템플릿과 reference 템플릿은 현재 시간의 유사도가 다음시간의 유사도에 영향을 끼치게 되므로 upper bound를 계산함으로써 계산량을 감소시킬 수 있으며 더욱더 빠른 검색을 할 수 있다.

reference 템플릿을 h_R이라 하고 프레임 n_1, n_2에서의 템플릿을 각각 $h_T(n_1)$과 $h_T(n_2)$라고 한다. 그리고 프레임 n_1과 n_2 사이의 유사도를 각각 $S(h_R, h_T(n_1))$과 $S(h_R, h_T(n_2))$가고 한다면 $S(h_R, h_T(n_2))$의 upper bound는 다음과 같다[2,3,5].

$${}_{ub}S(h_R, h_T(n_2)) = S(h_R, h_R(n_1)) + \frac{(n_2 - n_1)}{N} \quad (7)$$

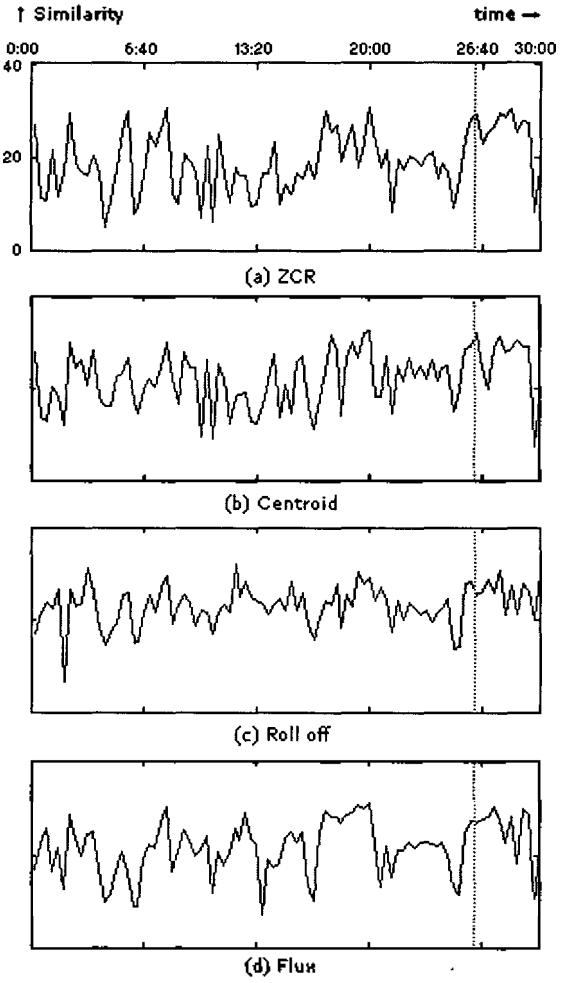

그림 3. 특징벡터의 유사도 비교

$$(n_2 - n_1) = N' {}_{ub}S(h_R, h_T(n_2) - S(h_R, h_R(n_1))' \quad (8)$$

그러므로 skip 넓이 w는 다음과 같이 정의 되어진다.

$$w_i = \begin{cases} floor(N(S_{thresh} - S_i)) + 1 & if S_i < S_{thresh} \\ 1 & otherwise \end{cases} \quad (9)$$

5. 실험 결과 및 고찰

본 논문에서 제안한 검색 방법의 성능을 알아보기 위해 30분짜리 라디오 방송을 11.025 kHz로 녹음하였고 방송에서 25분 13초부터 28분 35초에 흘러나온 박기영의 '시작' 이란 노래를 직접 O.S.T를 찾아 그 노래의 20초만을 reference로 두어 실험을 하였다.

5.1 검색 정확도

먼저 우리가 사용할 여러 가지 특징 벡터들을 히스토그램으로 나타내어 각각의 유사도를 비교해보았다. 그 결과는 <그림3>에 나타나 있고 어두운 부분이 찾고자 하

는 노래의 위치이고 점선은 reference 사운드가 위치해 있는 지점이다. 그러므로 그 지점에서는 다른 구간 보다 더 높은 값이 나와야 한다. 그러나 ZCR과 Centroid는 어느 정도의 결과가 나오는 반면 Roll off와 Flux는 좋은 결과가 나오지 않았다. 그래서 ZCR과 Centroid로 구성된 특징 벡터를 가지고 다시 유사도를 측정하였다. 그 결과는 <그림4>에 나타나 있다.

<그림3>에서 ZCR과 Centroid는 찾고자 하는 위치 외의 다른 구간에서도 더 높은 값이 나오는 반면, <그림4>의 ZCR과 Centroid로 구성된 특징 벡터는 찾고자 하는 위치에 유사도가 전 구간 내에서 가장 높은 값이 나왔으므로 검색의 정확도 면에서 성능이 향상되었다고 할 수 있다.

5.2 검색 속도

본 논문에서는 오디오 검색을 하는데 걸리는 시간과 계산량을 줄이기 위해 upper bound proof를 적용시켰다.<그림 5>는 ZCR과 Centroid로 구성된 특징 벡터를 가지고 upper bound proof를 적용시켜 유사도를 측정한 그림이다. 가로축은 프레임 개수이고 시간 영역으로 볼 때는 비선형 그래프이다. 점선은 찾고자 하는 사운드의 위치를 말한다. overlap을 시키지 않고 각각의 프레임의 유사도를 구할 때는 거의 80번째 프레임이 되어서야 사운드의 위치를 감지할 수 있지만 upper bound proof를 적용했을 때는 단 6번째 프레임에서 사운드의 위치를 감지할 수 있다. 그리고 사운드의 위치를 감지하는데 소요되는 시간은 각각의 프레임의 유사도를 구할 때는 17분이 걸렸지만 upper bound proof를 이용한 방법에서는 단 1분 35초가 걸렸으므로 속도도 향상된 것을 알 수 있다.

5. 결론

본 논문은 다중 특징 벡터를 이용하여 긴 오디오 stream에서 찾고자 하는 오디오의 위치를 빨리 찾을 수 있는 고속 검색 방법을 제안하였다. 그리고 ZCR과 STFT에 기반을 둔 Centroid, Roll off, Flux를 이용하였을 때 각각의 성능 비교 실험을 수행하였다. 비교 실험 결과 ZCR과 Centroid가 다른 특징들에 비해 좋은 성능을 보였으므로 ZCR과 Centroid로 특징 벡터를 구성하여 유사도를 비교한 결과 우수한 성능이 나왔고 upper bound proof를 적용하여 검색 속도를 증가시켰다.

본 논문에서의 실험을 토대로 장르마다 적합한 특징벡터 구성에 대한 연구를 지속하여 좀더 정확하고 빠른 오디오 검색 방법을 추구해 나가야 할 것이다.

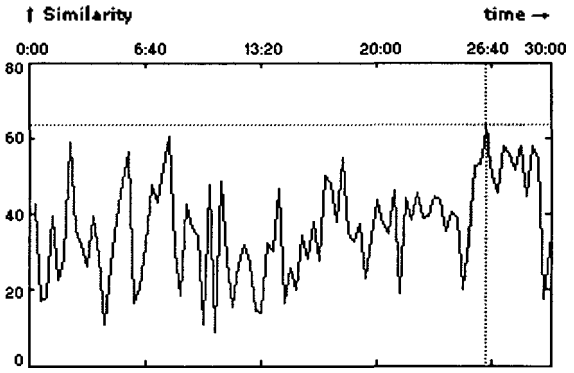

그림 4. ZCR과 Centroid로 구성된 특징 벡터를 사용한 유사도

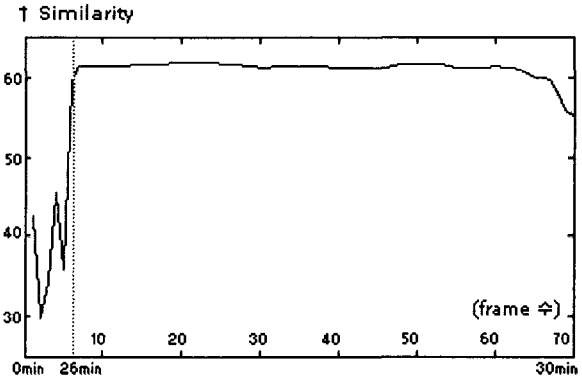

그림 5. upper bound proof를 적용한 유사도

참고문헌

[1] G. Tzanetakis.: "Manipulation, Analysis and Retrieval Systems for Audio", June, 2002

[2] G. Smith, H. Murase and K. Kashino: "Quick Audio Retrieval Using Active Search", *Proc.of ICASSP-98*, Vol. 6 (1998)

[3] V.V.Vinod and H. Murase: "Focused Color Intersection with Efficient Searching for Object Extraction", *Pattern Regonition*, vol. 30, No.10 (1997).

[4] J Saunders. Real-time discrimination of broadcast speech/music. In *ICASSP 96*, Vol. 2, pp. 993-996, 1996

[5] K. Kashino, G. Smith and H. Murase :"Time-Sereis Active Search for Quick Retrieval of Audio and Video", *Proc. of ICASSP99*, Vol. 6, pp. 2993-2996, March.1999