

DWT를 이용한 오디오 데이터 인덱싱 및 검색

조 용 춘, 이 배 호
전남대학교 컴퓨터공학과

Audio Data Indexing and Retrieval Using DWT

Yong-Choon Cho, Bae-Ho Lee
Dept. of Computer Engineering, Chonnam National University
E-mail : bhlee@chonnam.ac.kr

요 약

본 논문은 오디오 데이터의 인덱싱과 검색을 위해 DWT를 이용한 방법을 제안하였다. 오디오 데이터는 그 자신이 가지고 있는 다양한 특성 때문에 좋은 검색 효율을 위한 인덱스를 구성하기가 쉽지 않다. 신호 및 영상처리에서 각광받고 있는 DWT를 이용한 인덱스는 웨이블릿 변환이 가지고 있는 여러 특징들로 인해 데이터를 블록으로 나누지 않은 상태에서의 인덱싱과 검색을 가능케 한다. 즉 웨이블릿의 마지막 단계의 고주파 부분과 저주파 부분에서 고주파 부분은 String Matching 기법으로 블록을 결정하고, 저주파 부분은 결정된 블록에 대해서 세부적인 비교를 한다. 실험은 적절한 비교 계수를 결정하기 위한 실험과, 질의 길이의 변화에 따른 검색율의 변화를 보여준다. 마지막 결론에서는 본 논문에서 제안한 방법을 이용한 발전방향과 응용에 대해서 서술한다.

1. 서 론

멀티미디어 데이터의 사용의 폭발적인 증가로 멀티미디어 데이터에 대한 내용기반 인덱싱(Indexing) 방법의 개발과 관심이 증대되고 있다. 텍스트 등에서의 내용기반 인덱싱은 많은 부분 발전되었지만, 오디오 데이터에 대한 발전과 관심은 아직 부족하다.

오디오 데이터는 멀티미디어 데이터의 하나의 요소이지만, 다른 멀티미디어 데이터의 구성요소와는 달리 비독립적인 요소의 성향이 강하다. 즉 멀티미디어 데이

터에서 오디오는 부가적이면서도 필수적인 요소이다.

오디오 데이터도 역시 다른 멀티미디어 데이터가 가지고 있는 특성을 갖고 있다. 즉 많은 저장 공간, 전송에 있어서 실시간 필요성, 큰 대역폭 등의 특성은 멀티미디어 데이터가 가지고 있는 특성과 그리 다를 바가 없다. 내용기반 멀티미디어 데이터 베이스 시스템에서 기본은 간결하고 검색이 쉽게 기술된 표현으로 각각의 멀티미디어 객체를 인덱스 하는 것이다. 잘 조직된 인덱스는 빠르고 효과적인 자료의 열람과, 원하는 데이터에 대한 직접적인 접근을 가능케 한다. 이러한 것을 가능케 해주는 인덱스는 각각의 멀티미디어 객체 모델을 표현하고 명확하게 해주는 정보에 기반을 둔 의미 기술자(Descriptor)를 만들어 냄으로써 구성될 수 있다.

본 논문은 DWT를 통해 효과적인 내용기반 오디오 데이터 검색을 위한 인덱싱 및 검색방법과 제안된 인덱스를 이용한 검색 효율에 대해서 기술하였다.

II. 오디오 데이터 인덱싱

인덱스는 데이터를 효과적으로 검색하는 과정에서 사용되는 데이터의 검색 기법이다. 기존의 멀티미디어 데이터나 오디오 데이터에서 사용하는 키워드(Keyword) 기반 인덱싱은 제한적이다. 이것은 멀티미디어 데이터는 많은 표현이 존재 할 수 있고, 그러한 데이터를 해석해 내기도 그리 쉬운 일도 아니기 때문에 기존의 키워드 기반 인덱싱은 오디오 데이터에 적용하기는 많은 제한을 받을 수밖에 없다. 따라서 멀티미디어 데이터의 인덱싱은 내용기반 질의를 구성해야 한다. 또한 이러한

인덱스는 검색뿐 아니라 데이터의 삭제, 추가 등의 기능에도 적합해야 한다. 본 논문은 효율적인 검색을 위한 Wavelet 변환을 통한 인덱싱에 초점을 맞추고 있다. 오디오 데이터 인덱싱을 위해 사용되는 방법들은 아래와 같이 분류한다. [1]

(1) 통계 기반 인덱싱

이 방법은 평균, 분산, 영점 교차 등의 통계적 성질을 이용하는 것이다.

(2) 음향 정보를 이용한 인덱싱

이 방법은 오디오 데이터의 크기, 피치(Pitch)나 소리의 세기, 화성 등과 같은 음향정보를 이용하는 방법이다. 결과적으로 통계적인 해석은 이 특징 벡터를 만들기 위해 이러한 속성들을 이용하고 인덱싱이나 검색에 사용한다.

(3) 변환(Transform) 기반 인덱싱

이 방법은 DCT와 같은 변환은 데이터를 변환시키고, 적절한 계수를 골라내서, 그 계수를 인덱스로 사용한다. 여기서 적절한 변환방법의 적용과 최적의 계수를 찾아내서 인덱싱의 적용이 중요하다.

[3]에 의하면 변환 기반 인덱싱은 멀티미디어 데이터의 인덱싱에 좋은 결과를 기대하게 해 주고, 잡음에 영향을 최소화한다. 하지만 DCT와 같은 변환은 윈도우(Window) 크기에 주파수 정보와 시간영역 정보의 정확도가 좌우된다. DWT는 이러한 문제점을 해결해 준다.

III. 웨이블릿 기반 인덱싱

1. 웨이블릿 변환

Wavelet 변환은 임의의 함수 $f(x)$ 를 시간-주파수 공간에서 동시에 지역성(locality)을 갖는 Wavelet 기저함수의 선형결합으로 표현하는 것이다. $\Psi_{a,b}(x)$ 는 $\psi(x)$ 로 정의되는 원형 웨이블릿을 변이 시키고 신축 팽창, 그리고 이동시킴으로써 얻어진 함수들의 집합을 의미한다.

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{2}} \psi\left(\frac{x-b}{a}\right)$$

a : 스케일 변수(scale variable)

b : 이동 변수(translation variable)

웨이블릿 변환의 가장 일반적인 모델 중 한가지는 푸리에 변환과 lowpass filter, highpass filter를 사용하는 것이다. 웨이블릿 변환 중에서 DWT는 기저 함수들의 중복성이 없도록 직교 기저를 구성하기 위해 이산화한

것이며, 신호를 근사계수(Approximation Coefficient)와 세부계수(Detail Coefficient)로 나눔으로써 신호에 대해서 다른 분해능을 갖고 다른 주파수 영역에서 해석한다.

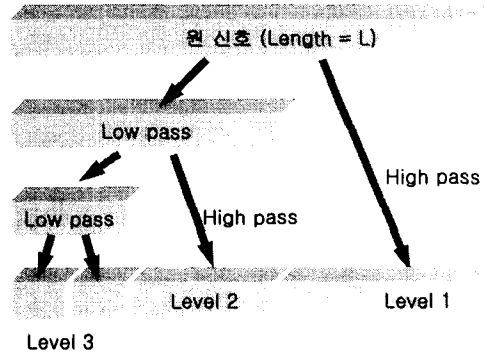


그림 1. 3단계 DWT 블록 다이어그램

그림 1.은 3단계 DWT의 블록도를 보여준다. 전체 신호의 길이를 L이라고 하면 n단계의 DWT시행 이후에도 신호의 길이는 변화가 없다. DWT의 단계가 내려 갈수록 근사계수의 크기는 작아지지만 그만큼 세부계수의 크기가 커져서 전체 신호의 크기는 원 신호의 크기와 변화가 없다.

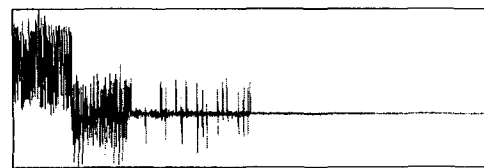
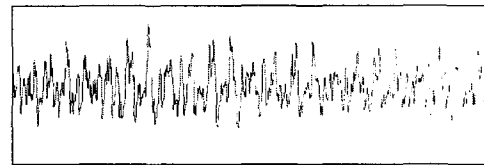


그림 2. 3단계 Wavelet 변환

마지막 단계의 저주파 영역은 근사계수이고, 각 단계의 고주파 영역은 세부계수이다. 모든 단계의 근사계수와 세부계수의 조합이 DWT 계수이다. 주목할 것은 저주파 영역은 연속해서 두 개의 영역으로 분리되고, 고주파 영역은 그대로 유지된다는 것이다. 그러므로 근사계

수는 높은 주파수 분해능을 갖게 되고, 시간 분해능은 낮아지게 된다. 그림 2는 원 신호에 대한 3단계 Wavelet 변환의 그래프를 보여준다.

2. String Matching 알고리즘

위에서 언급한 DWT를 이용한 인덱스의 작성은 한 개의 데이터 전체의 특징을 대표하는 계수를 갖게 된다. 오디오 데이터의 어떤 부분을 찾기 위해서는 표본 데이터의 각 부분마다의 DWT 계수를 가지고 있어야 한다. 즉, 하나의 데이터에 대한 하나의 계수를 가지고서는 DWT를 통한 인덱싱은 적어도 10초 정도 길이에 대해서 40 블록 이상에서 60%이상의 결과를 기대할 수 있다[1]. 따라서 표본 데이터를 블록화 하지 않고 오디오 데이터에 대한 검색을 하기 위한 인덱스를 작성하기 위해서는 DWT 계수에서 질의 데이터에 대한 블록을 설정할 수 있는 알고리즘이 필요하다. 본 논문에서는 작성된 DWT 계수에서 질의 데이터가 위치하는 블록을 결정하기 위해서 음악데이터 검색에 자주 사용되는 String Matching 알고리즘을 이용한다.

String Matching 알고리즘은 내용기반 음악 데이터 검색에 많이 쓰이는 알고리즘으로, 음악데이터의 멜로디를 세가지 문자의 연속으로 표현한다. 즉 음악 데이터의 Melody는 S(same pitch), U(pitch increase), D(pitch decrease)의 세가지 문자의 연속으로 변환시킨다.[4] 이러한 변환에 의한 검색은 개략적이게 되고, 데이터가 가지고 있는 중요한 특징들을 무시하게 된다. 따라서 String Matching 알고리즘을 이용하게 위해서는 세 글자의 연속으로 표현된 정보에 대해서 경계값을 설정하여야 한다. 가령 70%의 경계값을 설정한다면 표본 데이터의 길이에 대한 70%이상의 매칭 결과에서 비교 블록을 결정한다. 본 논문에서는 이러한 String Matching 알고리즘을 이용하기 위해 DWT 계수의 마지막 고주파 성분을 S, U, D의 세 문자의 연속으로 표현한다.

3. 인덱싱

본 논문에서는 DWT의 가장 마지막 단계의 계수를 이용하여 인덱싱한다. 그림 3은 본 논문에서 사용한 인덱싱 방법에 대한 블록 다이어그램이다.

표본 데이터의 길이를 N , n 단계 웨이블릿 변환 후 데이터의 길이를 L 이라고 하면 데이터 길이는 $L = N/2^n$ 이 되고, 근사계수와 세부계수의 크기는 $L/2$ 가 된다. 그 중에서 세부계수는 S, U, D 세 글자의 연속으로 변환하고, 이 문자열과 근사계수를 연결하여 인덱싱한다.

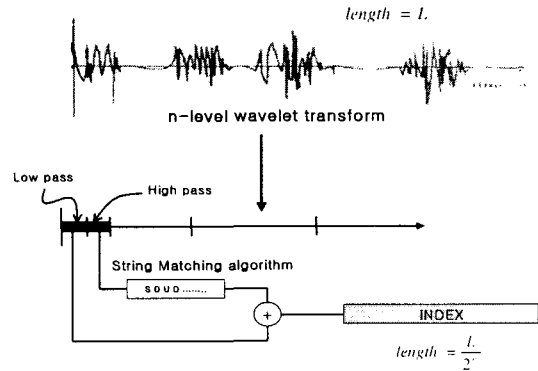


그림 3. 웨이블릿 변환을 이용한 인덱스 작성

4. 인덱스를 이용한 검색

위의 인덱스를 이용한 검색은 질의 데이터를 먼저 위의 인덱싱 방법과 동일하게 처리하는 전처리 과정을 거쳐서 데이터베이스의 인덱스와 비교한다. 질의 데이터의 세부계수와 데이터베이스의 세부계수를 이용하여 질의 데이터에 대해서 해당되는 후보 블록을 결정하고, 결정된 후보 블록에 대해서 근사계수를 비교하여 질의 데이터와 표본 데이터의 차이를 계산하고, 최소값을 갖는 데이터를 결과로 한다. 이러한 검색방법과 인덱싱 방법의 적용은 질의 데이터가 데이터베이스에 있는 어떤 데이터의 일부의 조각일지라도 찾아낼 수 있게 하기 위함이다. 즉, 질의 데이터가 어떤 오디오 데이터 파일의 중간정도에서 잘라낸 데이터일 경우 질의 데이터가 잘려져 나온 원래의 오디오 데이터와 그 위치를 정확히 찾아내기 위함이다.

IV. 실험결과

실험용 프로그램은 Microsoft Visual C++를 이용하여 프로그래밍 하였다. 실험은 100여개의 Wave File을 인덱싱하여 인덱스를 이용한 질의에 대한 검색에 초점을 맞추었다. 각 데이터들은 8bit mono, 초당 22KHz 샘플링 비트(Sampling Bit) 사운드를 사용하였다. 표본 데이터는 음악, 자연의 소리, 전자음 등을 혼합하였고 각 데이터의 길이는 225KB에서 3M까지 다양하게 구성하였다.

본 논문에서 제안한 알고리즘에서 사용한 인덱스를 이용한 검색은 세부계수를 이용한 블록을 결정하고, 결정된 블록에 대해서만 근사계수의 차이를 검사한다. 세부계수의 결정은 경계값을 이용한다.

먼저 세부계수를 이용하여 블록을 결정하기 위해서는 경계값을 결정하여야 한다. 경계값 결정을 위한 실험은 경계값을 인덱스로 구성한 세부계수와 질의 데이

터의 세부계수를 상호 매칭시켜 매칭된 개수를 c 라 하고, 질의 데이터의 세부계수의 개수를 q_c , 매칭 계수를 τ 라고 한다면, $c > q_c \times \tau$ 일 때, 블록으로 결정한다. 표 1.은 τ 값의 변화에 따른 검색율을 보여준다.

표 1. τ 값에 따른 검색율

질의 길이 \ τ	0.5	0.6	0.7	0.8
10초	70.4	81.5	55.6	18.5
6초	66.7	69.8	44.4	18.5

표 1.에서 τ 값이 0.5 일때와 0.6일때의 검색율은 그리 차이가 나지 않는다. 하지만 검색 시간에 있어서 많은 차이가 있다. τ 값을 0.5로 했을 경우 너무 많은 유사블록이 존재하게 되므로 많은 검색 시간을 소모하게 된다. 즉 본 논문에서 제안한 방법을 통한 검색은 τ 가 0.6일 때 좋은 효율을 낸다는 것을 볼 수 있다.

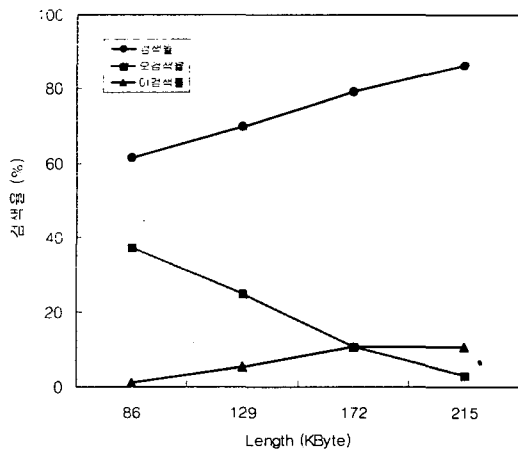


그림 4. 질의길이에 따른 검색율

그림 4.는 τ 를 0.6으로 설정하고 100여개의 표본 데이터에 대해 질의 데이터의 길이에 따른 검색율과 오검색율, 그리고 미검색율을 보여준다. 여기서 오검색율은 검색하였으나 질의한 데이터의 위치가 틀렸거나 잘못된 검색결과를 도출한 것이고, 미검색율은 검색하지 못해서 없는 데이터로 결과를 낸 것이다.

표 1.과 그림 4.의 결과를 볼 때 경계값 계수 $\tau=0.6$ 이고, 질의 데이터의 길이가 길수록 정확한 검색결과를

얻을 수 있다. 80% 이상의 검색율은 질의의 길이가 172K 이상일 때 기대할 수 있다. 이것은 [1].에서 제안한 알고리즘을 이용한 방법인 10초의 데이터를 40블록으로 나눈 후의 검색 결과보다 나은 결과임을 확인할 수 있다.

V. 결론

오디오 데이터를 블록으로 나누지 않고 인덱싱 하는 방법은 기존의 데이터를 여러 개로 나눈 후 인덱싱하는 방법과 상이하며, 임의의 데이터에서 잘려나온 질의를 가지고 원래의 데이터를 찾을 수 있다는 장점을 가지고 있다. 또한 String Matching 알고리즘을 이용하여 후보 블록을 결정하는 것은 기존의 사람이 블록을 결정하는 수 작업의 한계를 극복할 수 있게 하였고, 검색율에 있어서도 [1]에서의 검색율에 비해 상당히 향상된 결과를 보였다. 이러한 검색율의 향상에도 불구하고 아직은 질의의 길이와 모든 오디오 데이터 부류에 안정된 검색율을 기대할 수 있는 경계값 계수를 찾는 것은 아직 많은 실험과 연구를 필요로 한다.

이 방법은 오디오 데이터가 멀티미디어 데이터의 일부분인 점을 감안한다면, 멀티미디어 데이터의 워터마킹(Water marking)이나 비디오 데이터의 검색의 일부분으로 적용하여 기존의 방법과 더해져서 좋은 결과를 도출해 낼 수 있을 것이다.

참고 문헌

- [1] S. R. Subramanya and Abdou Youssef, "Wavelet-based Indexing of Audio Data in Audio/Multimedia Databases", IEEE, 1998
- [2] Chin-Chin Liu, Jia-Lien Hsu and Arbee L. P. Chen, "An Approximate String Matching Algorithm for Content-Based Music Data Retrieval", IEEE, 1999
- [3] S. R. Subramanya, "Experiments in Indexing Audio Data", Tech. Report, GWU-IIST, January 1998.
- [4] A Ghias, *et al.*, "Query by humming". Proc. ACM Multimedia Conf., 1995
- [5] T. C. Chiueh, "Content-Based Image Indexing", Proc. of the 20th VLDB Conf., pp. 582-593, 1994