

TIP-인덱싱 기법과 오디오 화일의 특징계수에 의한 내용기반 음악 검색[†]

(Content-based Music Retrieval by TIP-indexing
Techniques and Features of Audio files)

김 영 인*

(Young In Kim)

요약 방대한 양의 음악 데이터를 효율적으로 검색하기 위해서는 오디오 특징에 의해 인덱싱하고 검색하는 인덱싱 기법이 요구된다. 그러나 이와 관련한 인덱싱 기법에 대한 연구는 미진한 실정이다. 본 논문에서는 TIP-인덱싱 화일을 이용하여 오디오 특징 계수를 인덱싱하고 검색할 수 있는 내용기반 음악 정보 검색 기법을 제안한다. 또한 다양한 블록킹 인수를 이용한 성능 비교 실험을 위하여 TIP-인덱싱 화일을 만들고 실험한다. 실험 결과는 제안한 방법의 유효성을 보여준다.

핵심주제어 : 인덱싱 기법, 음악 정보.

Abstract To effectively manage a very large amount of music data, we need an indexing technique based on audio features. But the indexing techniques for audio features have not been studied completely. In this paper, we describe a content-based music information retrieval technique for audio features using the TIP-indexing file. In addition, we develop and experiment the TIP-indexing files using various blocking factors to present performance comparisons for effective indexing. Experimental results show the effectiveness of the proposed techniques.

Key Words : indexing technique, music information.

1. 서 론

최근 정보 기술의 발달로 인하여 대용량의 멀티미디어 데이터를 처리하려는 연구가 활발히 진행되고 있으며, 데이터베이스 분야에서도 멀티미디어 데이터와 같은 대량의 비정형 데이터를 관리, 검색, 저장하려는 필요성이 증대되고 있다. 멀티미디어 데이터의 한 종류인 음악 오디오 데이터를 다양한 정보 시스템에 효율적으로 저장하고 검색할 수 있는

음악 정보 검색 시스템에 대한 연구 개발의 중요성이 증대되어 활발히 연구되고 있는 실정이다[1]. 현재 음악 정보 검색에 대한 연구는 주로 기호 데이터와 오디오 데이터를 이용한 검색방법의 2가지 방법으로 이루어지고 있으며, 최근에는 이러한 2가지 검색방법 중, 보다 자연스러운 오디오 데이터를 이용한 연구가 활발히 이루어지고 있다[1, 2].

오디오 데이터를 이용한 검색은 웨이브와 mp3 등의 형태로 저장된 오디오 데이터에서 여러 가지 특징 계수를 추출하여 사용하며, 최근에 주로 사용하는 특징 계수는 MFCC(Mel Frequency Cepstral Coefficient)이다[2]. 이러한 특징 계수를 추출하여

[†] 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

* 부산대학교 바이오시스템공학부

사용하는 이유는 오디오 데이터는 원시 데이터(raw data)로서 대량이며 불필요한 데이터를 일부 포함하고 있으므로, 상대적으로 소량이면서 오디오 데이터의 특징을 나타내는 특징 계수가 보다 효율적이기 때문이다. 이러한 데이터는 시계열 패턴으로서 과학 응용의 공통적인 데이터 타입이며, 검색의 기본 대상이 된다. 본 논문에서는 음악 오디오 데이터의 범위를 음악 정보 검색의 중요한 특징 추출 데이터인 연속된 특징 계수를 시계열 패턴으로 한정하여 사용한다.

이러한 시계열 패턴을 데이터베이스에서 내용을 기반으로 효율적으로 저장하고 검색하기 위해서는 이에 적합한 인덱싱 기법이 필요하다. 이러한 형태의 데이터 처리를 위하여 다양한 다차원 공간 인덱싱 기법을 응용한 연구[3]의 필요성이 증가되고 있으며, 그 중 시계열 패턴을 위한 예비선택 기법을 포함한 인덱싱 기법인 TIP-인덱스 화일[4]을 이용한 연구가 필요하나 아직까지 이루어지지 않고 있는 실정이다.

본 논문에서는 오디오 데이터의 연속된 특징 계수를 검색하는데 적합한 TIP-인덱스 화일에 실제 오디오 데이터의 특징 계수를 저장하고 탐색에 활용하여 그 결과를 제시하고자 한다. 본 논문의 실험을 통하여 TIP-인덱스 화일이 음악 오디오 데이터의 검색에 적용 가능한가를 판단하고자 데이터 삽입시의 성능을 비교하기 위하여 디스크 접근 회수를 제시한다. 다음으로 데이터 탐색시의 성능을 비교하기 위하여 소요 시간을 구하여 보인다.

본 논문의 구성은 먼저 2장에서 관련된 연구에 대하여 살펴보고, 3장에서는 TIP-인덱스 화일에 대하여 알아본다. 4장에서는 실험 결과를 설명하며, 마지막으로 5장에서는 본 논문을 요약하고, 결론을 내린다.

2. 관련 연구

음악 오디오 데이터의 연속된 특징 계수를 데이터베이스에 저장하고 검색할 수 있는 인덱싱 기법으로는 주로 다차원 공간 화일 기법에 대한 연구가 이루어지고 있다[3]. 이러한 다차원 공간 화일 기법에는 음악의 특징 계수 각각을 프레임으로서 하나의 다중 키로 보며 다차원상의 점(point)으로

본다. 그러므로 다차원 공간 화일 기법에 연속된 특징 계수를 저장하여 검색할 수 있다. 이러한 다차원 공간 화일 기법은 많은 연구가 이루어지고 있으며, 그 중 주요한 몇 가지 기법을 설명하면 다음과 같다.

먼저 quad-트리[5]가 있다. quad-트리는 예를 들어 설명하면, 2차원 공간인 경우 북동, 북서, 남서, 남동의 사방을 기준으로 4-진 트리를 구성하여, 공간 물체를 저장하고 탐색한다. 둘째로 kd-트리[6]는 다차원 공간상의 점을 동질의 이진 탐색 트리에 각 차원을 운번 분할하는 방법으로, 분할 차원을 중심으로 이전에 저장된 점을 기준으로 작으면 왼쪽, 크면 오른쪽으로 저장하고 탐색한다. 셋째로 kd-트리[6]에 B-트리의 페이지징 방법과 높이-균등 트리로 만든 K-D-B-트리[7]가 있다. K-D-B-트리는 물체 식별자를 갖는 포인트 페이지와 분할된 부분 영역을 저장한 영역 페이지로 구성된다. 삽입 중에 포인트 페이지가 오버플로 되면, 해당 포인트 페이지에 저장된 점들을 같은 수로 나누어 영역 페이지에 새로운 포인트 페이지를 등록하고 새로운 페이지를 만들어 저장하고 탐색한다.

음악 화일의 특징 계수와 관련하여 이러한 다차원 공간 화일 기법의 사용이 어려운 이유는 첫째로, 시계열 패턴은 여러 개의 프레임으로 구성되며 각 프레임을 다차원 공간의 점으로 볼 때 하나의 점이 아닌 연관된 점의 순서 관계를 알아야 하는데, 이에 대한 기법이 있다는 점이다. 둘째로, 가장 가까운 점간의 예비 선택을 위하여 점간의 거리를 감안하여 저장해야 하는데 이에 대한 기법이 없다. 그러나 TIP-인덱스 화일은 연속된 특징 계수를 각 특징 계수의 순서를 감안하여 저장 및 탐색할 수 있는 것과 예비 선택이 가능한 장점을 가지고 있다.

3. TIP-인덱스 화일

TIP-인덱스 화일은 시계열 패턴에 적합하게 저장 단위인 프레임에 다중키(multi-key)의 형태로 특징 벡터를 저장한다. 저장한 프레임에는 순서 정보도 저장한다. 프레임을 저장할 때에는 거리를 감안한 분할 정책(split policy)을 사용하며, 프레임간의 분할 차원 값은 유클리드 거리를 기준으로 선정한다. 그리고 TIP-인덱스 화일에는 프레

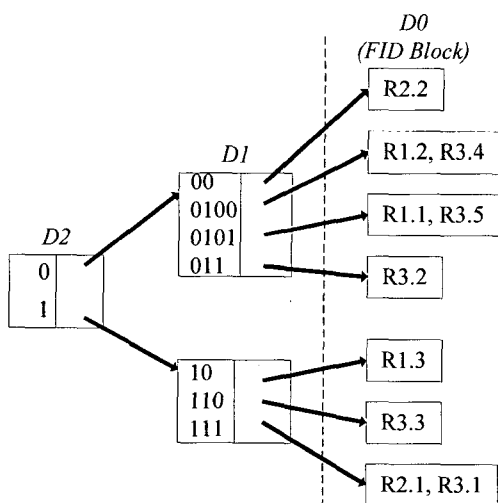
임 식별자만 저장하고, 실제 패턴은 별도의 데이터 화일에 저장한다. 프레임 탐색할 때에는 새로운 예비선택 기법으로서 동적 영역 테이블과 계층 디렉토리의 영역 벡터를 이용하여 미지 패턴의 프레임들을 순서대로 탐색하며, 이때 적중율의 향상을 위하여 ϵ 을 감안해서 찾아진 프레임 식별자로 예비선택 테이블을 작성하여 후보 패턴을 선정한다. 본 절에서는 TIP-인덱스 화일을 소개한다. TIP-인덱스 화일은 영역 벡터, 동적 영역 테이블, 계층 디렉토리로 구성된다. 각 구성 요소를 중심으로 설명하면 다음과 같다.

3.1 영역 벡터

각 차원의 영역 벡터가 중복됨으로써 발생하는 계층 디렉토리 구성시의 단점과 동적 영역 테이블의 구성 및 관리의 어려움을 해결하기 위하여, TIP-인덱스 화일에서는 Q-차원으로 구성된 데이터 공간내의 영역을 전체 분할 공간 영역에서 유일한 공간 영역을 나타내는 한 개의 해싱 값을 부여한다. 즉, 해싱 값은 영역 벡터로서, Q개로 구성된 영역 값에 대하여 영역을 2진수로 표현한 한 개의 원소를 갖는다. 그리고 영역 벡터는 계층 디렉토리에서 전위 2진수 매칭을 하여 탐색한다.

3.2 동적 영역 테이블

동적 영역 테이블은 영역 벡터와 영역 벡터의



<그림 1> 계층 디렉토리 구조

공간 영역을 나타내는 각 차원의 최소값과 최대값을 갖고 있다. 유일한 영역 벡터를 지원하기 위하여 테이블의 엔트리가 Q-차원이라고 할 때 <영역 벡터, 1차원 영역 값, 2차원 영역 값, ... , Q차원 영역 값>으로 구성되며, 동적 영역 테이블은 오직 한 개만 존재한다. 영역의 분열과 병합은 동적으로 변화한다.

$DRT = \{ (x,y) \mid x \text{는 하나의 공간 벡터, 그리고 } y \text{는 각 공간의 영역 값을 표현한다.} \}$

3.3 계층 디렉토리 블록

계층 디렉토리 블록은 유일한 영역 벡터를 이용하여 구성한다. 하위 레벨의 디렉토리가 블록킹 인수를 초과하면 공통의 전위 2진수를 기준으로 상위 레벨의 디렉토리를 구성한다.

$Dir = \{ (x,y) \mid x \text{는 하나의 공간 벡터, 그리고 } y \text{는 하위 디렉토리블록의 포인터이다} \}$.

$D1 = \{ (x,y) \mid x \text{는 하나의 공간 벡터, 그리고 } y \text{는 FID 블록의 포인터이다} \}$.

Dir은 계층 디렉토리로서 한개의 x는 공간 벡터로서 디렉토리 엔트리의 데이터 영역을 나타내는 영역 벡터이며, 다른 한개의 애트리뷰트인 y는 이 데이터 영역에 대한 보다 구체적으로 저장된 하위 단계 페이지의 주소이다. 영역 벡터는 각 키 애트리뷰트에 대응되는 N개의 해싱 값으로 구성된다. i번째 해싱값은 디렉토리 엔트리의 데이터 영역내에 속하는 모든 데이터 레코드의 i번째 키 애트리뷰트를 해싱하였을 때 나타나는 해싱값들의 공통 전위(prefix)가 된다. D1은 최하위 단계 디렉토리로서, 한개의 x는 공간 벡터로서 디렉토리 엔트리의 데이터 영역을 나타내는 영역 벡터이며, 다른 한 개의 애트리뷰트인 y는 이 데이터 영역에 속한 프레임 식별자 블록(FID Block: Frame Identifier Block) 저장된 하위 프레임 식별자 블록의 주소이다. 계층 디렉토리는 항상 균형 트리를 유지하며 디렉토리 블록의 블록킹 인수가 a이고, D1 디렉토리의 엔트리 갯수가 n이라고 할 때 $\lceil \log_a n \rceil + 1$ 의 레벨을 갖는다.

<그림 1>은 TIP-인덱스 화일 구조의 예이다.

<그림 1>과 같이 한 개의 영역 벡터만 사용하므로 기존보다 쉽게 상위 레벨의 디렉토리를 구성할 수 있다.

이러한 계층 디렉토리 블록의 분할은 최대거리 기반 분할 정책으로 분할된다. 최대거리기반 분할 정책은 분할되는 공간의 분할 차원을 저장된 참조 프레임의 각 차원 단위로 유클리드 거리로 최대, 최소값을 구하여, 그 중 차이가 가장 큰 차원을 선택한다.

4. 실험 및 결과

4.1 실험 방법

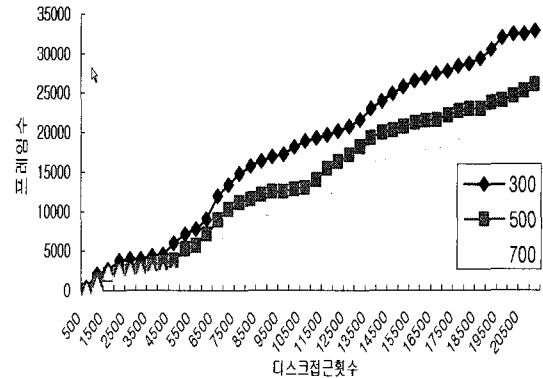
TIP-인덱스 화일을 사용하여 음악 정보 검색을 하기 위해서는 음악 오디오 화일에서 특징 계수를 추출하여 사용해야 한다. 실험에 사용한 음악 오디오 화일은 연주자들이 음악정보검색을 위하여 기부한 화일로 구성된 MIREX[2]에서 다운받은 6개 장르의 서로 다른 연주자 4그룹의 3곡으로 총 72 곡을 사용하였다. 특징계수는 13차 MFCC를 46ms 프레임 단위로 추출하여 사용하였으며, 실험에 사용한 오디오 클립은 각 오디오 화일에서 시작부분의 60초를 지난 13.9초를 음악 정보 분석 도구인 MARSYAS(Music Analysis, Retrieval and Synthesis for Audio Signals)[8]를 이용하여 추출하였다.[2] 실험에 사용한 오디오 화일은 22050Hz, 16bits 모노로 변환하여 사용하였다. 그리고 TIP-인덱스 화일의 성능 분석을 위하여 블록킹 인수는 300, 500, 700으로 하여 실험하였다. TIP-인덱스 화일의 영역 벡터는 320비트를 사용하였으며 실험에 사용한 컴퓨터는 IBM-PC 펜티엄IV(2.4GHz, 512MB)이며, 프로그래밍 언어는 C 언어를 사용하였다.

4.2 실험 결과

4.2.1 데이터 삽입시의 디스크접근회수

3가지 데이터 집합을 TIP-인덱스 화일에 블록킹 인수를 각각 300, 500, 700으로 삽입하면서 발생하는 디스크접근회수를 관찰하였다. <그림 2>는

3가지 데이터 집합에 대한 실험 결과를 나타낸 것이다.



<그림 2> 데이터 삽입에 따른 디스크접근회수

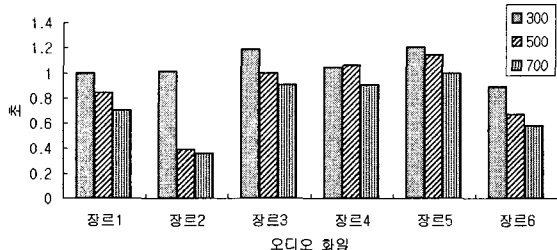
블록킹 인수가 작을수록 디스크접근회수가 증가하였다. 블록킹 인수가 300인 경우와 비교할 때 500과 700에서 각각 15%와 23%가 감소하였다. 블록킹 인수가 클수록 소요 시간이 짧은 이유는 데이터의 분할 횟수와 이에 따른 탐색 공간이 증가하였기 때문이다. 본 실험 결과로 TIP-인덱스 화일 구축에 있어서 적절한 블록킹 인수의 선택이 성능에 많은 영향을 미친다는 것을 보여 주고 있으며, 블록킹 인수가 커지면 공간 분할로 인한 소요 시간이 더욱 많아지고 성능이 저하됨을 알 수 있다.

4.2.2 데이터 탐색시의 소요시간 및 탐색 결과

TIP-인덱스 화일은 탐색되어진 프레임 식별자를 저장하고 비교하여 탐색된 후보 패턴을 선정한다. 그러므로 탐색의 성능 측정에 있어서 소요 시간은 중요하다. 시계열 패턴의 탐색 성능을 측정하기 위하여, 앞에서 생성한 TIP-인덱스 화일을 이용하여 서로 다른 장르의 임의의 6개 곡에 대하여 완전 일치로 검색되어지는 프레임 식별자 블록의 모든 데이터를 대상으로 성능을 측정하였다. 탐색시의 소요 시간을 비교한 결과는 <그림 3>과 같다.

완전 일치 탐색은 모두 성공하였으며, 실험 결과를 블록킹 인수가 300인 경우와 비교할 때 500과 700에서 각각 19%와 30%가 감소하였다. 블록킹

인수가 적을수록 탐색의 소요 시간이 많은 것으로 나타났다. 즉, 블록킹 인수가 적을수록 탐색에 읽은 디렉토리 블록이 많아 디스크 접근이 많이 발생하였기 때문이다.



<그림 3> 패턴 탐색에 따른 소요 시간

5. 결론

본 논문에서는 다차원 인덱스를 이용한 내용기반 음악정보검색 방법에 대한 관련된 기존 연구를 살펴보고, TIP-인덱스 화일을 사용한 방안과 실험 결과를 제시하였다. 실험을 위하여 6개 장르의 24 그룹의 아티스트 총 72곡을 사용하였으며, 특징계수로는 13차 MFCC를 추출하여 저장 및 검색에 사용하였다. 실험 결과에서는 다양한 블록킹 인수에 따른 저장 성능과 탐색 성능을 제시하였으며, TIP-인덱스 화일은 적절한 블록킹 인수에서 좋은 성능을 나타냄을 보였다. 이러한 실험 결과로 TIP-인덱스 화일에 대량의 음악정보인 특징 계수를 저장하여 사용할 수 있음을 보였다.

향후 실험에서는 보다 많은 오디오 화일에서 다양한 특징계수를 추출하여 사용하는 것이 필요하며, 또한 다른 다차원 인덱싱 기법을 시계열 패턴에 알맞게 확장 응용한 연구가 필요하다.

참고 문헌

[1] R. Typke, F. Wiering and R. Veltkamp, "A Survey of Music Information Retrieval Systems," In ISMIR Proceedings, pages 153-160, 2005.
 [2] The 1st Music Information Retrieval Evaluation

eXchange, <http://www.music-ir.org/mirex2005/>, 2006. (accessed 4/2006)

[3] J. Reiss, Jean-Julien Aucouturier and M. Sandler, "Efficient Multidimensional Searching Routines for Music Information Retrieval, In ISMIR Proceedings, 2001.
 [4] 김영인, 박영배, "시계열 패턴을 위한 개선된 인덱싱 기법," 한국정보과학회 논문지, Vol.22, No.12, 1995.
 [5] R. A. Finkel, J. L. Bentley, "Quad Trees : A data structure for retrieval on composite keys," Acta Informatica 4, 1974.
 [6] J. L. Bentley, "Multidimensional binary search trees used for associative searching," Comm. ACM, Vol. 18, No. 9, 1975.
 [7] J. T. Robinson, "The K-D-B-tree : A search structure for large multidimensional dynamic indexes," Proc. of Intl. Conf. on Management of Data, ACM SIGMOD, 1981.
 [8] George Tzanetakis, Marsyas software 0.2, <http://opihi.cs.uvic.ca/marsyas/>, 2004. (accessed 4/2006)



김 영 인 (Young In Kim)

- 1986년 한남대학교 전산학과 (학사)
- 1988년 명지대학교 전산학과 (석사)
- 1996년 명지대학교 컴퓨터공학과(박사)
- 1990년~1991년 (주)한국컴퓨터 기술연구소 연구원
- 1996년~2006년 밀양대학교 컴퓨터공학부 부교수
- 2006년~현재 부산대학교 바이오시스템공학부 부교수
- 관심 분야: 바이오지능정보시스템, 멀티미디어정보검색, 멀티미디어영상처리