韓國 컴퓨터情報學會 論文誌
第11卷 第5號, 2006. 11.

2006-11-5-1-8

오디오 특징계수를 이용한 시계열 패턴 인덱스 화일의 뮤지션 검색 기법

김 영 인*

Musician Search in Time-Series Pattern Index Files using Features of Audio

Young In Kim*

요 약

최근 멀티미디어 내용기반 검색 기술의 발달로 음악 정보 검색 기술 중 하나인 오디오 특징을 이용한 뮤지션 검색에 대한 관심이 증대되고 있다. 그러나 이와 관련한 음악 데이타베이스의 인덱싱 기법에 대한 연구는 부족한 실정이다.

본 논문에서는 시계열 패턴 인덱스 화일의 공간 분할 방법을 이용하여 오디오 특징 데이터를 사용한 뮤지션 검색 기법을 제시한다. 뮤지션 탐색을 위하여 오디오의 특징을 사용하며, 유사한 후보 뮤지션의 곡을 탐색하기 위한 인덱싱 기법으로 시계열 패턴 인덱스 화일을 사용한다. 실험 결과, 윤번 공간 분할 방법을 사용한 시계열 패턴 인덱스 화일이 뮤지션 검색에 있어서 효율적임을 보였다.

Abstract

The recent development of multimedia content-based retrieval technologies brings great attention of musician retrieval using features of a digital audio data among music information retrieval technologies. But the indexing techniques for music databases have not been studied completely.

In this paper, we present a musician retrieval technique for audio features using the space split methods in the time-series pattern index file. We use features of audio to retrieve the musician and a time-series pattern index file to search the candidate musicians. Experimental results show that the time-series pattern index file using the rotational split method is efficient for musician retrievals in the time-series pattern files.

▸ Keyword : 뮤지션(musician), 오디오 데이터(audio data), 시계열 패턴 인덱스 파일(time-series pattern file)

• 제1저자 : 김영인
• 접수일 : 2006.09.18, 심사일 : 2006.09.23, 심사완료일 : 2006.11.18
* 부산대학교 바이오시스템공학부 부교수

Ⅰ. 서 론

음악을 검색하는데 있어서 음악을 연주하거나 노래한 뮤지션을 기준으로 검색할 수 있다면 여러 가지 방면으로 다양하게 사용할 수 있다. 이를 위하여 음악이 저장된 오디오 화일에서 다양한 특징을 추출하여 검색하는 연구가 활발히 진행되고 있다[1,2,3,4].

기존의 음악 정보 검색에 대한 연구는 주로 기호 데이터와 오디오 데이터를 이용한 2가지 검색방법으로 이루어지고 있으며, 최근에는 이러한 검색방법 중 내용을 기반으로 한 검색으로 오디오 데이터를 이용한 연구가 주로 이루어지고 있다[3,5]. 오디오 데이터를 이용한 검색은 웨이브나 mp3 형태 등으로 저장된 오디오 데이터에서 특징 계수를 추출하여 사용하며, 최근에는 MFCC(Mel Frequency Cepstral Coefficient)가 좋은 성능을 보이는 것으로 알려져 주로 사용되고 있다[5]. 이와 같은 특징 계수를 이용한 연구는 대부분 패턴인식 기법을 이용하고 있으나, 대량의 오디오 데이터베이스를 사용한 검색에 있어서 중요한 예비 탐색 기법과 이를 사용하여 후보 곡의 대상을 줄이는 연구가 부족한 실정이다[3,4,6].

오디오 데이터베이스에 효율적으로 특징계수를 저장하고 검색하기 위해서는 적절한 인덱싱 기법이 필요하며, 이러한 형태의 인덱싱 기법은 주로 특징계수를 다차원 공간 상의 점으로 다룰 수 있는 다차원 공간 인덱싱 기법이어야 하며 연속된 특징계수의 순서를 감안한 저장과 탐색 기법을 가지고 있어야 한다. 이러한 인덱싱 기법에 대한 연구가 이루어졌으며, 그 결과로 시계열 패턴 인덱스 화일[6]이 제안되었다. 그러나 현재까지 시계열 패턴 인덱스 화일을 이용한 음악 정보 검색과 같은 실제 응용은 부족한 실정이다[7,8].

본 논문은 대량의 음악 데이터에서 추출한 오디오 특징계수에 데이터베이스의 인덱싱 기술을 이용하여 뮤지션을 검색할 수 있는 기법을 개발하는 것이다. 인덱싱 기술로는 오디오 데이터에서 추출한 특징계수를 시계열 패턴 인덱스 화일에 저장하는 기법을 의미하며, 이를 데이터베이스관리 시스템에 구현하여 유사 뮤지션의 곡을 탐색하는데 사용할 수 있다. 이러한 기법의 탐색 성능을 보이기 위하여, 다양한 블록킹 인수와 유사도 임계값을 2가지 공간분할방법과 함께 실험하여 결과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 시계열 패턴 인덱스 화일에 대하여 설명한다. 뮤지션 탐색의 성능은 3장에서 비교 분석한다. 마지막으로 4장에서 결론을 맺는다.

Ⅱ. 시계열 패턴 인덱스 화일

시계열 패턴 인덱스 화일은 영역 벡터, 동적 영역 테이블, 계층 디렉토리 그리고 공간분할방법으로 구성된다. 각 구성 요소를 중심으로 설명하면 다음과 같다[6,7,8].

1. 영역 벡터

영역 벡터는 Q-차원으로 구성된 데이타 공간내의 영역을 전체 분할 공간 영역에서 유일한 공간 영역을 나타내는 한 개의 해싱 값을 부여한다. 즉, 해싱 값은 영역 벡터로서, Q개로 구성한 영역 값에 대하여 영역을 2진수로 표현한 한 개의 원소를 갖는다. 그리고 영역 벡터는 계층 디렉토리에서 전위 2진수 매칭을 하여 탐색한다.

2. 동적 영역 테이블

동적 영역 테이블은 영역 벡터와 영역 벡터의 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. 동적 영역테이블의 예는 표1과 같다.

표 1. 동적 영역 테이블의 예
Table. 1 Example of Dynamic Range Table

영역 벡터	X축 영역값 (≤ , 〈)	Y축 영역값 (≤ , 〈)
0	0 , 20	0 , 10
1	0 , 20	10 , 20
00	0 , 20	0 , 5
01	0 , 20	5 , 10
10	0 , 20	10 , 15
11	0 , 20	15 , 20
010	0 , 16	5 , 10
011	16 , 20	5 , 10
110	0 , 7	15 , 20
111	7 , 20	15 , 20
0100	0 , 11	5 , 10
0101	11 , 16	5 , 10

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) | x는 하나의 공간 벡터, 그리고 y는 하위 디렉토리 블록의 포인터이다 }.

D1 = { (x,y) | x는 하나의 공간 벡터, 그리고 y는 FID 블록의 포인터이다 }.

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

4. 공간분할방법

시계열 패턴 인덱스 화일의 공간 분할 방법으로는 윤번분할방법과 최대거리기반분할방법이 있다.

윤번분할방법은 한 저장 공간에서 오버플로우가 발생하면 분할되는 공간의 분할 차원을 미리 지정된 순서에 따라서 윤번(rotation)으로 선택하여 저장된 특징 계수의 해당 차원 값의 중간 값을 구하여 공간 분할 기준으로 삼는다.

최대거리기반분할방법은 분할되는 공간의 분할 차원을 저장된 특징 계수의 각 차원 단위로 최대 값과 최소 값을 구하여, 그 중 차이 값이 가장 큰 차원을 선택한다.

최대거리기반분할방법과 윤번분할방법의 알고리듬은 다음과 같다. 여기서 FID_blk는 오버플로우가 발생하여 분할될 프레임 식별자 블록을 나타내며, split_dim은 분할할 차원을 나타낸다.

```
Algorithm: Select_dim_using_max_basis_split
/* Input : overflowed FID_blk, cur_feature_data
   Output : split_dim                          */
Select_dim_using_max (FID_blk, cur_feature_data)
{
   Get the feature data from FID_blk,
        let it be i;
   for each dimension {
        Get the feature data
             in the dimension, let it be j;
        Sort the feature data j[][dimension];
        Get the longest distance from data,
             let it be k[dimension];
        }
   Get the dimension of the longest
        distance from k[dimensions],
        let it be split_dim;
   return split_dim;
}
```

```
Algorithm Select_dim_using_rotation_basis_split
/* Input : overflowed FID_blk, cur_feature_data
   Output : split_dim                          */
Select_dim_using_rot (FID_blk, cur_feature_data)
{
   Get the dimension from split_dim, let it be i;
   j = ( i + 1 ) % dimension;
   store j to split_dim;
   return split_dim;
}
```

그림 1과 그림 2는 2차원 데이터를 2차원 공간에 ①부터 ⑩까지 순서대로 입력하여, 분할 방법에 따라 만들어진 저장 공간을 보인 예이다.

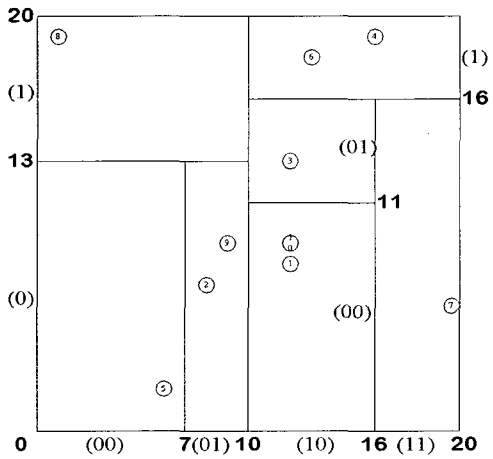

그림 1. 윤번분할방법의 예
Fig. 1 Example of rotational split policy

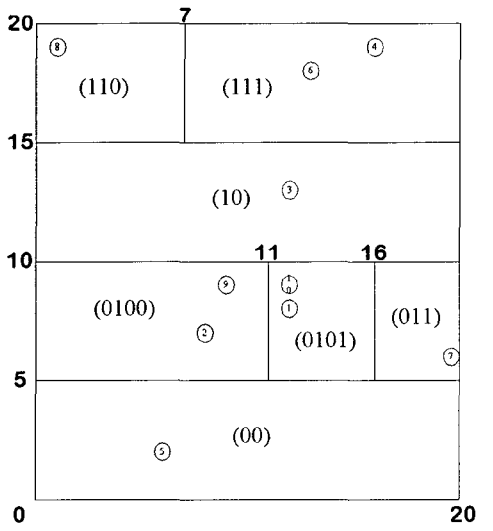

그림 2. 최대거리기반분할방법의 예
Fig. 2 Example of maximum distance based split policy

III. 실험 결과 및 고찰

1. 실험환경

사용된 프로그램은 마이크로소프트의 비주얼 C++ 6.0 이며 컴퓨터는 팬티엄 PC 2.6GHz CPU를 사용했다. 실험에 사용한 음악 곡은 MIREX[2]에서 구한 서로 다른 24가지 뮤지션의 각 3곡을 사용하였다. 각 오디오 화일은 22050Hz, 16비트 모노로 변환한 후, 각 오디오 화일에서 시작부분의 60초를 지난 13.9초 분량의 오디오 클립을 음악정보검색 도구인 MARSYAS(Music Analysis, Retrieval and Synthesis for Audio Signals)[9]를 사용하여 구하였다. 구해진 오디오 클립에서 13차 MFCC를 특징계수로 추출하여 사용하였다. 그리고 시계열 패턴 인덱스 화일의 다양한 성능 분석을 위하여 블록킹 인수는 400, 500, 600으로 하여 실험하였다. 시계열 패턴 인덱스 화일의 영역 벡터는 320비트를 사용하였으며 공간 범위는 0부터 100,000으로 하였다. 본 논문에서는 유사 탐색의 성능이 중요하다. 따라서 시계열 패턴 인덱스에서 유사도 임계값 ε의 변화에 따른 탐색 성능의 차이를 구하기 위하여 100, 200, 400, 800을 사용하였다. 또 저장하지 않은 미지의 뮤지션 곡에 대한 탐색 성능을 알아내기 위하여 각 뮤지션의 2곡만 저장하고, 1곡은 탐색에만 사용하였다.

2. 실험 결과

실험 결과, 표 2에서 표5는 ε 변화에 따른 뮤지션의 유사 곡 탐색 결과를 보인다. 표2에서 표5에 나타난 동일한 뮤지션의 곡이 포함된 후보곡 탐색률은 윤번분할방법이 최대거리기반분할방법보다 약 14.1%정도 높게 구해졌다. 그러나 탐색에서 찾아진 후보곡이 많으면 탐색 성능이 높아지며, 만일 찾아진 후보곡이 저장된 후보곡의 대부분이라면 탐색성능이 저하되는 원인이 된다. 따라서 탐색 결과로 찾아진 후보곡 중 해당 뮤지션의 곡이 포함된 비율이 중요하며, 이를 유효 후보곡 탐색률로 본다. 이에 대한 결과는 표6에서 표 9와 같다. 표6에서 표9의 결과에서는 즉, 실제 찾아진 곡 중 같은 뮤지션인 경우가 최대거리기반 분할이 약 2.6% 정도 높음을 알 수 있었다. 그러므로 후보곡 탐색률에 비해서 유효 후보곡 탐색률의 차이가 미미하므로, 전체적인 성능 측면으로 보면, 윤번분할방법을 사용한 시계열 패턴 인덱스 화일의 구축 방법이 보다 효과적임을 알 수 있

다. 따라서 대량의 오디오 데이터에 본 기법을 기존의 뮤지션 분류 검색 기법과 데이터마이닝 기술과 함께 사용하면 실용화에 도움이 될 것으로 판단된다.

표 5. 후보 곡 탐색 결과(ε=100)
Table. 2 Result for searching candidate music(ε=100)

		윤번분할 유사 탐색 성공률(%)	최대거리기반 유사 탐색 성공률(%)
블록킹 인수	400	37.5	31.25
	500	41.67	33.33
	600	50.0	33.33
평균		43.06	32.64

표 6. 후보 곡 탐색 결과(ε=200)
Table. 3 Result for searching candidate music(ε=200)

		윤번분할 유사 탐색 성공률(%)	최대거리기반 유사 탐색 성공률(%)
블록킹 인수	400	47.92	33.33
	500	43.75	37.5
	600	54.17	37.5
평균		48.61	36.11

표 7 후보 곡 탐색 결과(ε=400)
Table. 4 Result for searching candidate music(ε=400)

		윤번분할 유사 탐색 성공률(%)	최대거리기반 유사 탐색 성공률(%)
블록킹 인수	400	56.25	41.67
	500	58.33	43.75
	600	70.83	52.08
평균		61.80	45.83

표 8. 후보 곡 탐색 결과(ε=800)
Table. 5 Result for searching candidate music(ε=800)

		윤번분할 유사 탐색 성공률(%)	최대거리기반 유사 탐색 성공률(%)
블록킹 인수	400	75.0	58.33
	500	77.08	60.42
	600	85.42	66.67
평균		79.17	61.81

표 9. 유효한 후보 곡 탐색 결과(ε=100)
Table. 6 Result for searching the effective candidate music(ε=100)

		윤번분할 유효 탐색 성공률(%)	최대거리기반 유효 탐색 성공률(%)
블록킹 인수	400	15.83	18.45
	500	11.04	13.96
	600	12.41	14.36
평균		13.09	15.59

표 10. 유효한 후보 곡 탐색 결과(ε=200)
Table. 7 Result for searching the effective candidate music(ε=200)

		윤번분할 유효 탐색 성공률(%)	최대거리기반 유효 탐색 성공률(%)
블록킹 인수	400	16.76	17.47
	500	9.39	13.28
	600	12.0	14.24
평균		12.72	15.0

표 11. 유효한 후보 곡 탐색 결과(ε=400)
Table. 8 Result for searching the effective candidate music(ε=400)

		윤번분할 유효 탐색 성공률(%)	최대거리기반 유효 탐색 성공률(%)
블록킹 인수	400	12.83	13.02
	500	9.85	10.44
	600	10.27	18.44
평균		10.98	13.97

표 12. 유효한 후보 곡 탐색 결과(ε=800)
Table. 9 Result for searching the effective candidate music(ε=800)

		윤번분할 유효 탐색 성공률(%)	최대거리기반 유효 탐색 성공률(%)
블록킹 인수	400	10.44	14.38
	500	12.45	11.71
	600	8.24	12.65
평균		10.38	12.91

Ⅳ. 결 론

본 논문에서는 시계열 패턴 인덱스 화일을 이용한 오디오 데이터에서 뮤지션을 검색할 수 있는 방안을 제시하고자 노력하였다. 다양한 불록킹 인수와 유사도 임계값을 사용하여 시계열 패턴 인덱스 화일의 성능을 실험하였다. 실험 결과로 윤번 분할 방법을 사용하여 구축한 시계열 패턴 인덱스 화일이 최대거리기반 분할 방법을 사용한 경우보다 탐색의 성공률이 높으며, 이는 기존의 관련 기술과 함께 사용하면 검색의 성능 향상 및 실용화에 도움이 될 것으로 판단된다.

향후 연구 방향은 대량의 오디오 데이터를 이용하여 보다 더 효율적인 공간 분할 방법을 고안하여야 하며, 다른 형태의 검색 방법에 대하여도 연구할 필요가 있다.

참고문헌

[1] J. T. Foote, "Content-based retrieval of music and audio," Multimedia Storage and Archiving Systems II, Proc. of SPIE, pages 138-147, 1997.

[2] Annamaria Mesaros, Jaakko Astola, "The mel-Frequency cepstral coefficients in the context of singer identification," pages 610-613, ISMIR 2005.

[3] R. Typke, F. Wiering and R. Veltkamp, "A survey of music information retrieval systems," In ISMIR Proceedings, pages153-160, 2005.

[4] J. Reiss, Jean-Julien Aucouturier and M. Sandler, "Efficient multidimensional searching routines for music information retrieval, ISMIR 2001.

[5] The 1st Music Information Retrieval Evaluation eXchange, http://www.music-ir.org/mirex2005/, 2006. (accessed 4/2006)

[6] Kim Y. I., Park Y. & Chun J., "A Dynamic indexing structure for time-series patterns," The Twentieth Annual International Computer Software and Applications Conference(COMPSAC'96), 1996.

[7] 김영인, 김선종, "시계열 패턴의 효율적인 검색을 위한 공간 분할 정책의 성능 평가," 한국정보기술학회논문지, 4권, 4호, 71-76쪽, 2006.

[8] 김영인, "TIP-인덱싱 기법과 곡단위 특징을 이용한 내용기반 음악 검색", 한국산업정보학회논문지, 11권, 3호, 10-14쪽, 2006.

[9] George Tzanetakis, Marsyas software 0.2, http://opihi.cs.uvic.ca/marsyas/, 2004. (accessed 4/2006)

저 자 소 개

김 영 인

1996년 명지대학교 컴퓨터공학과 (박사).

1990년~1991년 (주)한국컴퓨터 기술연구소 연구원.

1996년~2006년 밀양대학교 컴퓨터공학부 부교수.

2006년~현재 부산대학교 바이오시스템공학부 부교수.

관심 분야 : 바이오지능정보시스템, 멀티미디오정보검색, 멀티미디어영상처리 등임