

Cosine Measure를 이용한 음악 동기간 유사도 계산

임상혁*, 구경이**, 김유성*

*인하대학교 정보통신공학부

**인하대학교 전자계산공학과

e-mail:g2021026@inhavision.inha.ac.kr

Similarity Computation between Music Motifs Using Cosine Measure

Sang-Hyuk Lim*, Kyong-I Ku**, Yoo-Sung Kim*

*School of Information and Communication Engineering, Inha University

**Dept of Computer Science Engineering, Inha University

요 약

음악에서 동기는 독립성을 지니는 최소 단위이며, 저작권 검사의 단위로 이용된다. 따라서, 한 음악에서 약간의 변화를 가지고 반복되는 주제선율을 추출하거나, 다른 음악간의 유사도를 측정하는데 유사도 계산은 필요하다. 본 논문에서는 비교되는 동기의 선율정보를 음길이와 음높이가 함께 고려되는 시계열 데이터로 변환하고, cosine measure를 이용하여 동기간의 유사도를 계산한다. 시계열 데이터에서 유사도 계산으로 사용되는 유클리드 거리함수 대신 cosine measure를 이용한 경우, 공간상의 거리 합대신 변화 방향이 반영됨으로써 비교되는 동기간의 유사도를 정확하게 계산한다. 본 논문에서 제안된 동기간의 유사도 계산은 내용 기반 음악 검색에서 색인으로 사용되는 주제선율을 추출하거나, 다른 음악의 동기간의 유사성을 비교하는데 이용될 수 있다.

1. 서론

멀티미디어 정보 응용분야의 처리 기술 발달에 따라 대용량 멀티미디어 데이터의 한 분야인 음악 데이터에 대한 효율적인 관리 및 검색이 필요하다 [1]. 음악에서 동기는 독립성을 지니는 최소 단위이며, 저작권 검사의 단위로 이용된다[2]. 따라서, 한 음악에서 약간의 변화를 가지고 반복되는 주제선율을 추출하거나, 다른 음악간의 유사도를 측정하기 위해 동기간 유사도 계산은 필요하다.

음악은 음의 높고 낮음이 시간에 따라 변화하는 선율 패턴을 가지고 있고, 이러한 선율 패턴은 시계열 데이터(time-series data)의 패턴과 유사하므로 음악 데이터를 시계열 데이터로 변환하여 처리하면 음의 높이와 길이를 함께 고려하는 음악의 특성 표현이 가능하다. 본 논문에서는 국제적 표준인 미디(MIDI : Musical Instrument Digital Interface) 파일[3]로부터 음악의 특징 요소 및 유사도 계산을 위해 필요로 하는 정보를 추출한다. 추출된 정보를 통해 음악을 구성하는 요소 중 독립성을 지닐 수 있

는 최소 단위이면서 한 곡이 통일성을 갖게 하는 동기 단위로 선율 정보를 시계열 데이터로 변환한다. 시계열 데이터의 유사도 계산은 유클리드 거리함수 [4]를 이용하지만, 계산된 값은 각 원소간의 거리합이기 때문에 공간상의 거리 표현에는 적합하나, 음악의 유사도값을 나타내기에는 적합하지 않다.

본 논문에서는 유클리드 거리함수대신 cosine measure[5]를 이용하여 변화 방향이 반영되도록 동기간의 유사도를 계산한다. cosine measure를 이용한 유사도 계산은 한 음악에서 약간의 변화를 가지고 반복되는 해당 음악의 주제선율을 추출하거나, 다른 음악간의 유사성을 비교하는데 이용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구를 소개한다. 3장에서는 본 논문에서 제안된 음악 정보의 시계열 데이터 변환 과정과 변환된 데이터를 이용한 유사도 계산식을 보인다. 4장에서는 실험을 통하여 cosine measure를 이용한 동기간 유사도 계산의 정확성을 분석하고, 5장에서 본 논문의 결론 및 향후 연구 방향을 기술한다.

2. 관련연구

시계열 데이터는 주식 데이터의 변화, 기온의 변화 등과 같이 일정 시간 간격에 따라 변화하는 값들의 연속으로 표현된 시퀀스로서 $X = (x_1, x_2, x_3, \dots, x_n)$ 으로 표현될 수 있다[4].

음악을 시계열로 분석하려는 연구로서 음악적 언어들로 표현한 악보를 시계열 자료로 해석하는 방법을 [6]에서 제안되었다. 그러나, 일정한 패턴을 가지는 시계열로 변환하는 최적화된 방법을 제시하지 않았다. 또한, 악보를 통한 시계열 모델링 방법의 이용은 음악의 유사도 계산이 불가능하다.

대량의 시계열 데이터를 관리하는 시계열 데이터베이스에서는 데이터베이스내의 시계열 데이터의 집합 $S = \{X_1, X_2, X_3, \dots, X_N\}$ 로부터 주어진 질의 $Q = (q_1, q_2, q_3, \dots, q_k)$ (단, $k \leq n$)과 유사한 패턴을 포함하는 시계열 데이터 X_i 를 검색한다. 즉, 질의 Q 와 데이터베이스내의 각 시계열 데이터의 서브패턴간의 거리 $dist(Q, SubPattern(X_i))$ 를 계산하여 주어진 거리 안에 위치한 서브패턴을 포함하는 모든 시계열 데이터를 검색한다. 이때 두 시계열 데이터간의 유사도는 (식 1)과 같은 다차원공간내의 유클리드 거리함수[4]를 사용한다.

$$dist(Q, SubPattern(X_i)) = \sqrt{\sum_{j=1}^k (Q_j - X_i(m+j))^2}, m=1..n-k \text{ (식 1)}$$

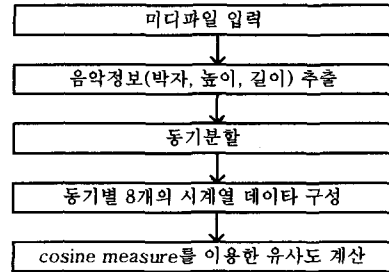
그러나, 유클리드 거리만을 이용한 유사검색을 통해서 사용하는 사용자가 원하는 시퀀스들을 검색하지 못하기 때문에 유사 모델을 적절하게 정의할 수 있도록 정규화, 이동평균, 타임워핑 변환을 수행해야 한다. 또한, 계산된 값이 각 원소간의 거리함이기 때문에 공간상의 거리 표현에는 적합하나, 음악의 유사도값을 나타내기에는 적합하지 않다.

[7]은 음악을 일정한 선율에 따른 음들의 배합으로 정의하고, 선율을 구성하는 음높이와 음길이 정보를 복합적으로 이용하여 두 음악 정보간의 유사도 계산 알고리즘을 제안하였다. 하지만, 계산된 유사도값은 음길이와 음높이에 대한 상대적인 차의 거리함으로서 음의 변화 방향을 반영하지 않는다.

3. Cosine measure를 이용한 동기간 유사도 계산

본 절에서는 음악을 구성하는 요소 중 독립성을 지닐 수 있는 최소 단위인 동기 단위로 유사도를 계산한다. 한 음악에서 cosine measure를 이용한 동

기간 유사도 계산의 수행 과정은 <그림 1>과 같다. 한 음악에서 동기간 유사도 계산은 한 음악에서 약간의 변화를 가지고 반복되는 해당 음악의 주제선율을 추출하는데 이용된다.



<그림 1> 동기간 유사도 계산 수행 과정

동기간 유사도 계산의 입력으로서 국제적 표준인 미디(MIDI : Musical Instrument Digital Interface) 파일[3]을 대상으로 고려한다. 입력된 미디 파일로부터 음악의 특징 요소 및 유사도 계산을 위해 필요로 하는 정보를 추출해야 하는데, 필요로 하는 정보는 박자(time-meter), 키(key), 음높이(pitch)와 음길이(length) 집합 등이다. 추출된 정보를 이용해 입력된 미디 파일을 동기별로 분할한다.

음악은 시간의 흐름에 따른 음의 높이 변화, 즉 일반적인 시계열 자료의 형태인 시간의 경과에 따른 출력의 형태를 가진다. 이러한 사실을 바탕으로 음악의 정보들 중, 음의 길이와 높낮이만을 사용하여 시계열 자료로 재구성한다. 음길이는 32분 음표와 쉼표를 기준으로 재구성한다. 음표와 쉼표의 시간 영역의 정보로 재구성된 값은 <표 1>을 이용한다. 악보의 음악적 정보들을 시계열 자료의 시간으로 변환하였다면, 음높이를 시계열 자료에서 시간에 대응되는 출력으로 변환해야 한다. 이러한 값은 <표 2>의 MIDI의 음높이 값을 이용한다.

<표 1> 시계열 데이터로서의 음길이 값

기호	명칭	32분음표와의 길이
	32분음표	1
	16분음표	2
	8분음표	4
	4분음표	8
	2분음표	16
	은 음표	32

<표 2> 미디에서의 음높이 값

Note	Octave										
	0	1	2	3	4	5	6	7	8	9	10
C	00	0C	18	24	30	3C	48	54	60	6C	78
C# / Db	01	0D	19	25	31	3D	49	55	61	6D	79
D	02	0E	1A	26	32	3E	4A	56	62	6E	7A
D# / Eb	03	0F	1B	27	33	3F	4B	57	63	6F	7B
E	04	10	1C	28	34	40	4C	58	64	70	7C
F	05	11	1D	29	35	41	4D	59	65	71	7D
F# / Gb	06	12	1E	2A	36	42	4E	5A	66	72	7E
G	07	13	1F	2B	37	43	4F	5B	67	73	7F
G# / Ab	08	14	20	2C	38	44	50	5C	68	74	
A	09	15	21	2D	39	45	51	5D	69	75	
A# / Bb	0A	16	22	2E	3A	46	52	5E	6A	76	
B	0B	17	23	2F	3B	47	53	5F	6B	77	

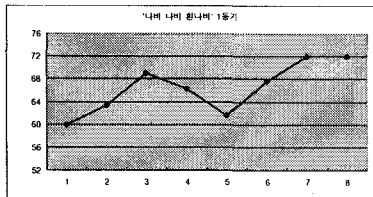
MIDI 파일에서 추출된 음악 정보(박자, 높이, 길이)는 <표 1>과 <표 2>를 이용하여 음높이를 시계열 영역에서의 시간에 대한 출력으로 재구성한다. 즉, 'C5'인 '1'은 음높이 60(hex 3C)을 갖으면서 길이값 8로 표현된다. 단, 헛표의 경우, 음높이값은 0으로 한다.

<표 1>를 기준으로 하여 n/m 박자($\rightarrow n/4$)를 갖는 한 동기내의 음들은 음길이에 따라 표현될 경우, 중간 과정으로서 $16n'$ 개의 시퀀스로 표현되며, 다시 길이 2 n' 개씩 정규화된다. 한 동기내의 모든 음들은 따라서 8개의 시계열 데이터로 표현되어진다. 한 동기내의 음들을 시계열 데이터로 모델링하는 경우, 한 음악내에서 변박이 발생되거나, 박자가 서로 다른 음악내의 동기가 서로 비교되어도 동일한 개수를 갖는 시계열 데이터로 변경된다.

<그림 2(a)>의 동요 "나비 나비 흰나비"의 첫 번째 동기를 시계열 데이터로 나타낸 것은 <그림 2(b)>와 같다.



<그림 2(a)> 동요 "나비 나비 흰나비"의 1동기 악보



<그림 2(b)> 1 동기의 시계열 데이터의 표현

<그림 2> 동요 1동기 시계열 데이터의 표현 예

정보 검색 모델중 벡터 모델의 유사도 계산에 이용되는 cosine measure는 문헌을 질의에 유사한 순서대로 정렬시킬 수 있으며, 가중치 할당이 가능하다[5]. 2개의 항목 i, j 를 m 차원 공간에서 두 벡터로 표시한 후, 두 항목의 유사도를 두 벡터 사이의 코사인 각도를 이용하여 측정하는 cosine measure는 (식 2)와 같다.

$$\cos(i, j) = \frac{i \cdot j}{|i| \times |j|} \quad (\text{식 2})$$

cosine measure에 따른 유사도값은 0과 1사이의 값이 된다. 따라서 cosine measure를 이용한 음악의 유사도 계산은 유사도값에 따라 순위를 매기기 때문에 질의와 유사한 음악을 검색할 수 있으며, 일정한 유사도값 이상의 음악을 검색하거나, 순위에 대한 가중치를 줄 수도 있다.

비교되는 두 동기간의 유사도 계산시 동기내에 발생하는 음의 개수가 같은 경우, 동기내의 음들을 시계열 데이터로 모델링하는 대신 휴리스틱하게 미디의 음높이와 음길이의 값을 사용하여 cosine measure에 적용할 수도 있다.

본 논문에서 제안된 시계열 모델링 과정과 cosine measure를 이용하여 계산된 동기간 유사도값을 이용하면, 해당 음악을 구성하는 각 동기들을 유사성을 갖는 선율들로 클러스터하여 해당 음악의 주제선율을 추출할 수 있다. 추출된 주제선율만을 색인으로 저장하고, 검색시 색인된 주제선율을 이용하여 음악 정보를 적은 공간을 사용하여 빠르고 정확하게 내용 기반 검색을 수행할 수 있다.

4. 동기간 유사도 계산에 대한 실험

<그림 3>과 같이 4개의 동기로 구분되는 동요 "남생아 놀아라"의 동기간 유사도 계산은 <그림 4>와 같다. i 번째 행과 j 번째 열의 엔트리 (i, j)는 i 번째 동기와 j 번째 동기의 유사도값을 의미하며, 0~1사이의 값을 0~100사이의 값으로 표현하였다.



<그림 3> 동요 '남생아 놀아라'의 악보

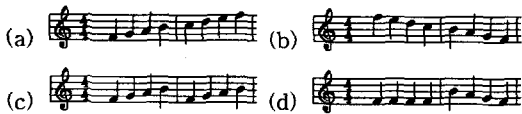
	1	2	3	4
1	100	75	100	87
2	75	100	75	94
3	100	75	100	87
4	87	94	87	100

Press any key to continue

<그림 4> "남생아 놀아라" 동기간 유사도 계산 결과

<그림 4>의 악보에 따르면, 1, 3번째 동기가 정확하게 서로 같음을 인지할 수 있으며, 따라서 엔트리 (1,3), (3,1)은 유사도값으로서 100을 갖는다. 2, 4번째 동기의 경우, 유사도값으로 96을 갖는다.

cosine measure를 이용한 동기간 유사도 계산의 정확성을 비교하기 위해 [7]의 유사도 계산 알고리즘, 유클리드 거리함수를 본 논문에서 제안하는 cosine measure를 이용한 유사도 계산과 비교실험하였다. 비교 실험을 위해 사용된 동기 데이터는 <그림 5>이며, 실험 결과는 <표 3>과 같다.



<그림 5> 실험을 위한 동기 데이터

<표 3> 실험 결과

	유클리드 거리함수	[7]의 연구	cosine measure
(a)-(b)	7.9	81	44
(a)-(c)	4.8	90	84
(a)-(d)	6.2	86	68
(b)-(c)	6.4	85	51
(b)-(d)	7.0	86	57
(c)-(d)	4.1	80	49

실험 결과에 따르면 (a)-(b), (a)-(c)동기 사이의 유사도 계산 값에서 [7]보다 cosine measure가 음악의 변화 방향을 정확하게 반영함을 알 수 있다. 또한, (a)-(c)동기와 (c)-(d) 동기의 경우 유클리드 거리함수는 방향 변화 반영하지 못하지만, cosine measure의 경우 방향 변화를 반영함을 알 수 있다. 따라서, 본 논문에서 제안하는 cosine measure를 이용한 동기간 유사도 계산은 음악의 변화 방향에 반응하여 정확하게 유사도를 계산함을 알 수 있다. 계산된 유사도값은 내용 기반 음악 검색 시스템에서 곡 전체를 색인하는 대신에 약간의 변화를 가지고 반복되는 선율인 주제선율을 추출하거나, 다른 음악 간의 유사도 계산에 사용되어질 수 있다.

5. 결론 및 향후 연구 방향

음악에서 동기는 독립성을 지니는 최소 단위이며, 유사성을 비교하는 저작권의 단위로 이용된다. 음의 높고 낮음이 시계열 데이터의 패턴과 유사하므로 음악 데이터를 시계열 데이터로 변환하여 처리하면 음의 높이와 길이를 함께 고려하는 음악의 특성 표현이 가능하다. 본 논문에서는 음악의 특성을 유지하도록 선율 정보를 정규화 과정을 거쳐 시계열 데이터로 변환하고, cosine measure를 이용하여 선율정보간의 유사도를 계산한다. cosine measure를 이용한 유사도 계산값은 시계열 데이터에 이용되는 유클리드 거리함수나, 기존의 유사도 계산에 비해 음악의 변화 방향에 반응하여 좀 더 정확한 유사도 값을 지님을 실험을 통해 알 수 있었다.

동기간 유사도값을 이용하여 유사성을 갖는 선율들로 클러스터링하면 해당 음악의 주제선율을 추출하거나, 저작권을 위해 비교되는 동기간의 유사도를 계산할 수 있다. 향후 본 논문에서 제안된 동기간 유사도 계산 알고리즘을 이용한 내용 기반 음악 검색의 수행 과정을 보이도록 한다.

참고문헌

[1] P.Y.Rolland, "Music information retrieval : A brief overview of current and forthcoming research," Workshop Engineering and Music, 2002.
 [2] 박세광, 음악대사전, 세광음악출판사, 1996.
 [3] 김종화, MIDI 총론, 세운, 1994.
 [4] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Database," Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms, Chicago, Illinois, pp. 69-84, Oct. 1993.
 [5] 김명철 외 5인, 최신정보검색론, 홍릉과학출판사, 2001년.
 [6] 손세호, 권순학, "퍼지 이론을 이용한 악보의 모델링," 한국 퍼지 및 지능 시스템 학회 논문지, Vol. 11, No. 3, pp.264 - 269, 2001.
 [7] Jong-Sik Mo, Chang-Ho Han, and Yoo-Sung Kim, "A Similarity Computation Algorithm for MIDI Musical Information," The Proceedings of 1999 IEEE Knowledge and Data Engineering Exchange Workshop, Chicago, USA, 1999.