

허밍 질의를 위한 그룹 허밍 표현법⁺

남현우*, 황성호*, 박능수*, 권순일**

*건국대학교 컴퓨터공학과

**한국과학기술연구원

e-mail:namhw@konkuk.ac.kr

A Group Humming Expression for Query By Humming

Hyunwoo Nam*, Seong-Ho Hwang*, Neungsoo Park*, Soonil Kwon**

*Dept of Computer Engineering, Kon-Kuk University

**Korea Institute of Science and Technology

요 약

최근 멀티미디어를 검색하기 위해 메타데이터 기반의 검색 방법에서 콘텐츠 자체를 검색 하려는 연구들이 활발히 진행되고 있다. 특히 음악 검색의 경우 허밍 입력으로 검색을 하려는 QBH(Query By Humming)가 많은 관심을 끌고 있다. 하지만 허밍 데이터는 개인마다 음높이나 박자 정보들이 모두 다르고 숨소리 등의 내재된 오류 정보들이 많아 정확한 검색 결과를 얻기가 쉽지 않다. 허밍 검색의 정확도 향상을 위해서는 음 데이터 추출이나 허밍의 오류 보정, 유사도 측정과 관련된 연구들이 선행되어야 한다. 본 논문에서는 효과적인 멜로디 표현방법에 대해 다양한 실험을 통해 최적의 모델을 제시하려 한다. 방법으로 UDR을 다양한 범위로 나누고 가중치를 달리하는 방법으로 실험을 한 결과 허밍을 그룹으로 분류하는 방법이 정확도를 향상 시키는 것을 확인 하였다.

1. 서론

허밍을 통한 음악 검색은 기존 메타데이터 기반의 검색 시스템에서 해결할 수 없었던 부분들을 해결해주게 되어 많은 관심을 끌고 있다. 1990년 중반부터 연구들이 시작되어 학계와 산업체에서 지속적으로 연구되고 있는 분야이기도 하다.[1][2] 현재까지 완벽한 검색이 가능하지는 않지만 좀 더 실용적인 단계까지 연구가 진행 된다면 실생활에서도 많이 활용될 것이다.

일반적으로 메타데이터 태그를 이용한 음악검색에서는 가수, 작곡가, 가사 등 정확한 정보를 알지 못하면 노래를 찾는 것이 불가능 하였다. 허밍 검색의 경우 정확한 정보를 알지 못하더라도 노래를 허밍 할 수 있을 정도의 기억만 있다면 원하는 노래를 검색할 수 있다. 예를 들면 노래방에서 음악을 선곡할 때나 길을 걷다가 들려오는 노래를 벨소리로 다운로드 받으려 할 때 허밍 검색 시스템을 사용할 수 있다.

이와 같은 이점에도 불구하고 허밍입력은 개인마다 그리고 상황에 따라서 음의 높이나 박자가 다르고 허밍 자체의 오류도 많이 내재되어 있어 정확한 검색 결과를 얻기란 쉬운 일이 아니다. 따라서 정확한 허밍음의 추출과 효과적인 멜로디 표현 방법 그리고 불안정한 허밍 데이터 유사도 측정 방법에 대한 연구가 요구되고 있다.

본 논문에서는 그중 허밍 검색시스템에서 사용하는 허밍 표현을 개선할 수 있는 방법을 제시한다. 본 논문은 다음과 같은 구성으로 되어 있다. 2장에서 먼저 효과적인 멜

로디 표현을 위한 연구 결과들을 살펴보고 3장에서 허밍 검색 시스템에서 검색이 어떻게 처리 되는지 알아본다. 4장에서 허밍검색 시스템에서 정확한 결과를 얻을 수 있도록 허밍 표현 방법을 제안하고 5장에서는 실험을 통하여 연구 결과를 분석하고 6장에서 결론과 추후 연구방향을 제시한다.

2. 관련 연구 분석

허밍검색에서 정확한 결과를 얻기 위해서는 정확한 허밍 입력이 필요하다. 허밍의 특징상 개인마다의 특징이 반영되어 절대적인 수치로 정량화 하기란 쉽지 않다. 허밍 데이터를 검색에 얼마나 효과적으로 반영할 수 있는지는 검색 정확성 면에 많은 영향을 준다. 허밍에서 음표를 추출하고 효과적으로 반영할 수 있는 멜로디로 표현하는 방법이 중요하다는 것을 의미 한다 .

기존에 연구되어온 허밍 검색 시스템으로 “QBH:Query By Humming[1]” 와 “MELDEX:Melody Indexing[3]”가 있다. 이 시스템들은 UDR(UDS)라는 문자열로 멜로디를 표현하고 있다. 음 변화에 따라 이전 음보다 높은 음이라면 U(up) 낮은 음이라면 D(down) 같은 높이의 음이라면 R(Repeat) 또는 S(Same)의 3개의 문자열로 변환하여 사용한다. UDR은 음높이의 변화 정보만을 담고 있는데 실제 허밍검색에서 사용하기에는 정보의 양이 부족하다.

위와 같은 UDR 형식의 문제점을 해결하고자 했던 방법으로 Themefinder[4]는 음의 변화 방향뿐만 아니라 높이 변화량을 포함 하여 검색의 정확도를 향상시켰다. 다른 연구로는 음 변화에 대한 히스토그램을 기준으로 일정 비

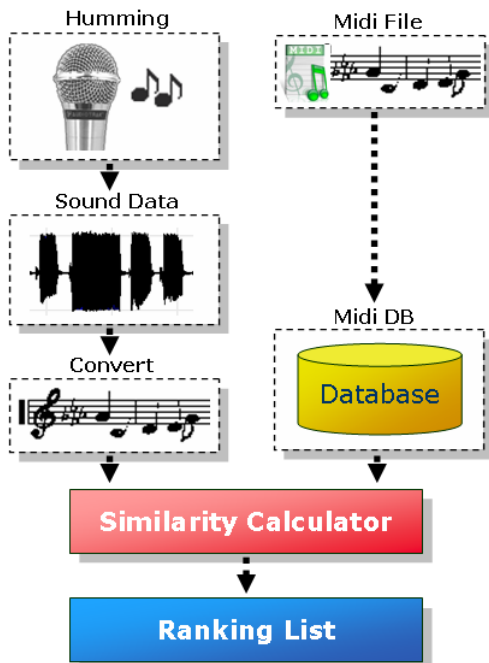
⁺이 논문은 서울시 산학연 협력사업의 지원에 의한 논문임.

율로 UDR을 분류해주는 방법[5]도 있다.

UDR 형식은 허밍에서 나타나는 개인차를 고려하지 않은 멜로디 표현방법이다. 허밍은 같은 노래라도 사람마다 다른 음높이와 음길이가 추출되는 개인차를 보이므로 이를 보완해 줄 수 있는 방법이 필요하다. 음성 인식에서 사용되고 있는 HMM(Hidden Markov Model)은 불안정한 허밍의 음 변화를 학습하여 확률 통계적으로 사용할 수 있는 방법[6]이다. HMM을 통해 불안정한 허밍의 단점을 보완해 줄 수는 있지만 학습을 하기 위해서는 많은 사람들의 노력과 시간이 필요하다. 그렇게 구한 데이터조차 모든 사람들에게 적용되지 않다는 단점을 지니고 있다.

3. 허밍 검색 시스템 구조

시스템은 크게 허밍에서 신호처리를 통해 음표 데이터를 추출해주는 부분과 미디파일에서 음표데이터만을 추출해주는 부분 그리고 두 음원간의 유사도를 계산하여 순위 리스트를 구해주는 부분으로 구성된다.



(그림 1) 허밍 검색 시스템 구조도

허밍입력이 발생하면 FFT(Fast Fourier Transform)를 통해 음성 신호를 구하게 되고 미디 파일에서 사용되는 음표[7]로 변환하게 된다. 음표로 추출하는 과정에서 숨소리가 검출되거나 음의 시작하는 부분과 끝나는 부분에서 튀는 음들이 발생하였는데 이를 제거해준 후에 멜로디 표현 방법으로 변환하였다.

미디 파일은 허밍 음과는 달리 절대적인 음표 정보를 가지고 있다. 허밍 검색에서 음높이 데이터만을 사용하고 있어 미디 파일에서도 음 높이가 데이터만을 추출한다. 같은 방법으로 모든 음악에서 음표 정보를 추출하고 난 다음 DB에 저장한다.

위와 같은 방법으로 허밍과 모든 미디 파일은 동일한

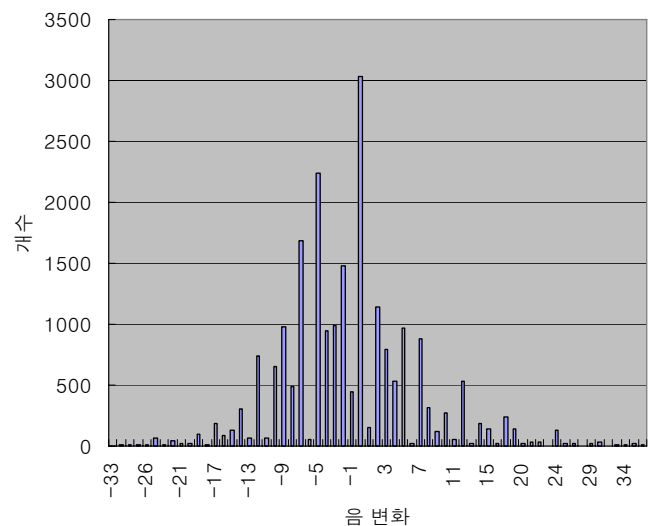
도메인 영역의 음표 데이터를 추출 할 수 있다. 추출된 데이터들은 유사도를 측정할 때 사용하는데 앞에서 구한 음표 데이터를 그대로 사용하지는 않고 이 논문에서 제안하고 있는 그룹 허밍 표현으로 변환하여 사용된다.

기존 UDR 형식으로 멜로디를 표현하게 되면 일반적인 스트링 매칭 알고리즘을 적용하여 유사도를 계산 할 수 있다. 하지만 본 시스템서는 UDR 형식을 세분화 하고 가중치를 대입하기 때문에 일반적인 스트링 매칭 알고리즘을 사용할 수 없다. 대신 음악과 허밍에서 가중치의 차이 값을 음악의 끝부분까지 쉬프트 해가며 구하였고 그중 최소가 되는 값을 선택한다. 선택 된 차이 값이 작을수록 허밍과 음악사이의 유사도가 크다는 것을 의미하며 그중 최소가 되는 값이 허밍과 음악사이의 유사도를 의미한다.

허밍과 모든 음악과의 유사도 값이 구해지고 나면 유사한 음악 순서대로 순위가 매겨진 순위 리스트가 출력되는 것으로 음악검색을 마치게 된다.

4. 효과적인 UDR 표현 방법

UDR 형식의 멜로디로 변환할 때 좀 더 효과적인 표현 방법을 실험을 통해 찾아보고자 한다. 기존 멜로디 표현 방법의 문제점은 저장할 수 있는 정보의 양이 너무 적다는 것이다. 따라서 좀 더 세밀하게 음들을 분리한다면 정확한 검색결과를 얻을 수 있을 것이다. 하지만 세분화를 많이 한다고 해서 정확도가 비례해서 증가하는 것은 아니다. 오히려 허밍 입력 오류에 민감해져 오차가 증가하는 현상이 발생한다. 입력 오류의 오차 크기와 검색 결과의 정확도는 반비례하여 나타나는데 적절한 세분화 범위 설정에 대해 타협점을 찾아야 한다.



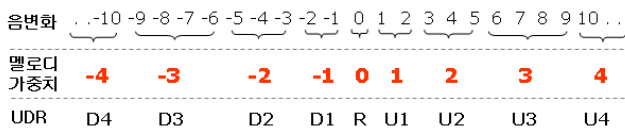
(그림 2) MIDI 파일에서의 음 변화 분포

제안하고자 하는 방법은 음의 변화를 적절히 그룹으로 분류하고 차등을 두어 가중치를 할당하는 것이다. 여기서 그룹으로 분류할 때 어느 정도가 적절한 그룹 범위인지를 선택하는 것은 중요한 문제이다. 최적의 범위 선택을 위해

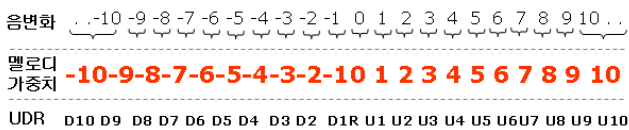
다양한 경우의 수를 실험 하였고 그 결과를 토대로 최적의 모델을 제시하고자 한다.

실험에 앞서 음 높이의 변화가 실제 MIDI 음원에서 어느 정도의 분포 범위를 이루고 있는지를 알아봐야 한다. 현재 검색에 사용되고 있는 미디 파일들을 가지고 음 변화의 분포를 조사 하였고 결과는 그림2와 같다. 대체적으로 음높이의 변화는 상승하는 음높이와 하강하는 음높이의 변화가 각각 2옥타브 정도 내에서 이루어지는 것으로 확인 되었다.

따라서 위의 결과를 가지고 여러 경우로 나누어 실험을 하였고 그 중 대표적인 방법 2가지를 제시한다. 방법1은 음의 변화가 작은 것은 그룹의 크기를 작게 하고 음의 변화가 크면 그룹의 크기도 크게 하는 것이다. 이는 음의 변화가 작으면 허밍을 할 사람이 정확하게 허밍 할 가능성이 많아지고 크면 클수록 허밍이 부정확해 질 가능성이 높다고 판단되기 때문이다. 방법 2는 그룹으로 나누지 않고 현재 음의 변화 값을 가중치로 사용하여 세밀하게 나누는 방법이다. 다음은 각 방법에 따라 멜로디 표현 형식 구조를 나타낸 그림이다.



방법1. 음 변화율에 비례하여 그룹으로 분류



방법2. 그룹하지 않고 음 변화의 값을 가중치로 선택

5. 실험 결과

대부분의 미디 파일들은 여러 악기음으로 구성되고 그 중 보컬에 해당하는 중심 악기음이 존재한다. 본 논문에서는 실험을 위해 동요와 가요로 구성된 100개의 미디파일을 단음으로 변환하여 데이터베이스를 구축 하였다. 허밍은 발음을 “나”로 통일 하여 한곡 당 12~15초 정도를 16비트, 모노 방식으로 샘플링 하여 녹음하였다. 실험에 사용한 허밍음의 개수는 45개이며 검색 결과 10위권 내로 검색된다면 검색이 성공했다고 판단한다.

<표1> 허밍검색 실험 결과

표현방법	UDR	방법1	방법2
검색 성공 개수	15개	26개	17개
평균 검색 순위	23위	19위	21위

허밍에서 음표 데이터를 추출하기 위해 Audio 라이브

러리[8]를 사용하였다. 추출된 허밍 데이터는 숨소리나 튀는 음도 포함하고 있어서 그런 오류들을 제거하기 위해 추가적인 도구를 제작하였다.

위의 <표1>의 결과를 보면 UDR로 멜로디를 표현한 방법보다는 제안하는 방법이 더 좋은 검색 결과를 보여준다. 제안하고 있는 방법 중에서 방법1의 경우가 음의 변화량을 그대로 가중치로 사용하는 방법2보다는 좋은 결과를 보여주고 있다. 방법2와 같이 세밀하게 나누는 방법은 허밍이 정확했을 경우 최고의 매칭 결과를 보여주겠지만 허밍의 오류에는 민감하게 반응하기 때문이다.

6. 결론

실험결과 정확도를 위해 단순히 음 높이를 세밀하게 나누어 준다고 정확도가 향상되는 것은 아니었다. 적절한 크기의 그룹으로 멜로디를 표현하는 것이 오히려 오류에 둔감해지므로 정확한 검색을 얻는데 효과적이다.

본 논문에서 그룹의 범위를 임의적으로 나누어 사용하고 있는데 추후 명확한 기준이 제시되어야 필요성이 있다. 현재 우리가 참고하는 음 변화율에 대한 그래프는 미디파일에서 조사한 결과이다. 허밍에서는 음 변화가 다르게 나타날 수 있으므로 실제 사람이 얼마나 정확하게 음 변화를 표현할 수 있는지에 대한 통계 자료가 필요하다. 그 결과를 기반으로 허밍과 미디파일에서의 멜로디 변환 기준을 달리 할 수도 있고 허밍에 특징적인 요소를 추가해 볼 수도 있을 것이다.

이번 연구 결과를 기반으로 위에서 언급한 추가적인 요구 사항들에 대해 지속적인 연구를 진행하고자 한다.

참고문헌

[1] A. Ghias, J. Logan, D. Chamberlin and B. C. smith, "Query By Humming: Musical Information Retrieval in Audio Database" In Proc. ACM Multimedia. 1995
 [2] <http://www.midomi.com/>
 [3] R. J. McNab, L. A. Smith, I.H. Witten, C. L. Henderson and, S. J. Cunningham, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input" Digital Libraries, 1996.
 [4] K. Andreas, "Themefinder: A Web-based Melodic Search Tool," Melodic Similarity: Concepts, Procedures and Applications, Computing in Music, 1998
 [5] Sonoda, Goto, Muraoka "A WWW-based Retrieval System" IEICE Transactions on Information and Systems, 1999.
 [6] Shifrin, Bryan Pardo, Colin Meek, William Birmingham "HMM-Based Musical Query Retrieval" ACM InternationalJoint Conference on Digital Libraries, 2002.
 [7] <http://www.harmony-central.com/MIDI/Doc/table2.html>
 [8] <http://aubio.piem.org/>