거리 측정방법에 따른 쿼리-바이-싱잉/허밍 시스템의 성능 변화

장세진, *장달원, 이석필

전자부품연구원

*dalwon@keti.re.kr

Performance of Query-by-singing/humming system depending on the distance metric

Sei-Jin Jang, Dalwon Jang, and Seok-Pil Lee

Korea Electronics Technology Institute

요약

이 논문에서는 쿼리-바이-싱잉/허밍 (Query-by-singing/humming, QbSH) 시스템에서의 거리 함수를 다양화하면서 그 성능 변화를 살펴본다. QbSH는 디지털 음악의 사용이 보편화되면서 음악 검색의 방법으로 많은 연구가 진행되어 왔으며, 많은 경우, dynamic time warping (DTW) 방법으로 사용해서 정합하고 있다. 그러나, DTW에서 사용하는 거리에 대해서는 특별한 관심을 가지지 않았으며, 일반적으로 절대적 차이값이나 그것의 제곱값을 많이 사용해 왔다. 이 논문에서는 여러 가지 거리에 대해서 성능을 측정하였다. 성능측정은 특정한 시스템에서 이루어진 것이기 때문에 일반성을 가지지 않을 수 있으나, DTW에서 사용하는 거리를 기존의 것과 다른 것으로 변화시켜서 성능을 향상시킬 가능성이 있음을 이 논문에서는 밝힌다. 본 논문에서는 10-12초 길이의 1000번의 쿼리 (Query)에 대해서 28시간 정도의 데이터베이스에서 실험한 결과, 논문에서 제안하는 거리가 기존의 절대적 차이값을 사용한 것보다 제1후보 검출 정확도가 10% 가량 상승함을 확인할 수 있었다.

1. 서론

디지털 음악의 유통과 사용이 많아지면서 대용량의 음악 데이터베이스 (database, DB)에 저장된 음악 정보들을 효율적으로 정리하고 검색하는 기술에 대한 많은 수요가 있어왔다. 특히 최근 스마트폰과 스마트 TV의 보급은 음원 시장에 대한 쉬운 접근을 가능하게 하였고, 그에 따라서 디지털 음악의 구입이 활성화되고 있다. 그런 이유로 음악 정보 검색 (Music information retrieval, MIR) 분야가 많은 관심을 받고 있다 [1,2]. 이 분야에는 장르 분류 (music genre classification)[9], 멜로디 표기 (melody transcription) [10], 오디오 인식 (audio identification) [11], 쿼리-바이-싱잉/허밍 (QbSH) [3-8] 등의 연구분야가 존재한다. 이 중에서 QbSH는 사용자의 소리 입력을 받아들인다는 점에서 다른 연구분야들과 차별성을 보인다.

이 논문에서는 QbSH 시스템을 구축하는데 있어서 반드시 필요한 요소인 정합 과정을 다루고 있으며, 그 중 DTW 알고리즘을 사용한 정합 과정을 다루고 있다. DTW 알고리즘은 여러 QbSH 시스템에서 사용하는 방법으로[3-8] 시간적으로 변화가 있는 두 개의 서로 다른 시퀀스를 정합시킬 시에 많이 사용되는 방법이다. 이 방법은 각 시퀀스의 원소 값 사이의 거리를 구하고, 원소값들 사이의 거리의 합을 최소화시키는 원소값들 사이의 패스를 결정하면서 두 시퀀스의 정합 정도를 살핀다. 이 때 시퀀스의 원소 값들 사이의 거리를 결정하는 것은 주로차이의 절대값이나 그것의 제곱값이 많이 사용되곤 하였다 [5,6,8]. 하지만, 시퀀스의 특정에 따라서 다른 거리 함수가 더 나은 결과를 나을 수 있다. 따라서 시스템에 따라서 다양한 거리 함수를 실험해 보는 것이 성능 향상에 도움이 될 것이다.

논문은 다음과 같이 구성된다. 2장에서는 우리의 QbSH 시스템을 개략적으로 설명하고, 3장에서는 다양한 거리 함수를 설명한다. 4장에서는 실험결과를 보이고 5장에서 결론을 보이면서 마친다.

2. QbSH 시스템의 구조

가. 전체적인 구조

그림 1.에 우리의 QbSH 시스템의 구조를 도식화하였다. 기존의 QbSH 시스템들과는 다르게, 우리의 시스템은 MIDI 입력에 의존하지 않고 MP3 등의 다음 (polyphonic) 음악에 기반을 두고 있다. 다음음악에 대해서 피치 시퀀스 (pitch sequence) 를 추출하고 이를 특징 DB (feature DB)에 저장한다. 사용자에게서 쿼리 (query) 입력을 받아서 이로부터 피치 시퀀스를 추출한다. 이 피치 시퀀스와 특징 DB에 저장된 피치 시퀀스를 정합하여 보고 특징 DB에 있는 피치 시퀀스 중 가장 유사한 피치 시퀀스로부터 노래의 정보를 추출해낸다.

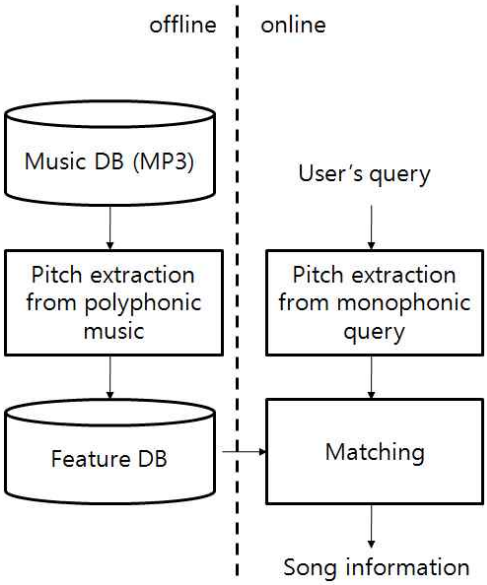

그림 1. QbSH 시스템의 구조

나. 피치 시퀀스 추출

피치 시퀀스를 신호로부터 추출하는 것은 64ms 길이의 하나의 프레임 당 하나의 피치 값이 나오도록 하여 구하였다. 음악 입력은 8kHz로 다운샘플링되고 64ms 길이의 프레임으로 나눠진다. 하나의 프레임에서 하나의 실수 값이 나온다. 사용자 입력으로부터 피치 추출은 시간과 주파수 조합영역에서의 선형 보간법을 이용해서 구해진다 [13]. 다음 음악으로부터 피치를 추출하는 방법은 평균 하모닉 구조 (average harmonic structure, AHS)에 기반을 두었다 [14,15]. 다음 음악으로부터 다수개의 기본 주파수 후보를 선정하고, 그 것들 중에서 AHS 값에 기반하여 프레임 당 하나의 피치 값을 결정하였다.

다. 정합과정

정합과정은 DB에 들어있는 시퀀스 중에서 사용자의 쿼리에서 추출한 피치 시퀀스와 가장 유사한 것을 찾는 과정이다. 이를 위해서 DB에 들어있는 모든 시퀀스에 대해서 거리를 측정하는 과정이 필요하며, 거리는 DTW에 기반하여 결정된다.

DTW는 [6]에서와 같이 비대칭적으로 (Asymmetric) 구현되었다. 입력으로 DB에 들어있는 피치 시퀀스와 사용자의 쿼리에서 추출한 피치 시퀀스, 두 가지 시퀀스를 받아들이고, 사용자 쿼리로부터 추출한 시퀀스에 정합되는 DB 시퀀스의 특정 부분을 찾는 과정을 반복한다.

정합의 성능을 높이기 위해서 피치 시퀀스는 chroma 레벨로 표현하고, 음의 높낮이에 대한 보정을 하여준다[12].

3. 거리

본 논문에서는 다섯 가지의 거리 함수를 사용하였다. 두 개의 실수 값 a,b 가 있을 때 a와 b 사이의 거리를 다음의 다섯 가지 방법으로 나타내었다.

$$d_{|.|}(a,b) = |a-b|$$

$$d_{|.|^2}(a,b) = |a-b|^2$$

$$d_{HINGE}^{(\lambda)}(a,b) = \begin{cases} |a-b| & \text{if } |a-b| < \lambda \\ \lambda & otherwise \end{cases}$$

$$d_{LOG}(a,b) = \log(1+|a-b|)$$

$$d_{SIG}(a,b) = \frac{1}{1+\exp(-|a-b|)} - 0.5$$

처음에 표기된 두 가지 거리를 흔히 많이 사용되는 방법이고 [5,6,8], 아래에 사용된 세 가지는 논문에서 제안하는 방법이다. 첫 번째 거리 함수는 차이의 절대값을 가지고, 두 번째 거리 함수는 차이값의 제곱을 가진다. 아래의 세 가지 거리는 큰 차이값에 대해서 덜 민감한 특징을 가지고 있다. 이러한 거리는 특정한 한 값이 큰 에러를 가질 때 그 에러에 대한 영향을 줄여줄 수 있다.세번째 거리 함수는 입력의 차이값이 특정한 상수 λ보다 작을 때는 그 값을 그대로 가지고, λ보다 클 때에는 값이 λ로 제한된다. 네 번째 거리 함수는 log 스케일에 기반한 값이며 다섯 번째 거리 함수는 sigmoid 함수를 사용하였다.

위의 다섯 가지 거리 함수를 그림으로 표기하면 다음의 그림 2.와 같다. 그림에서 보기 편하게 하기 위해서 $d_{SIG}(.)$의 경우, 4배를 해주어 레벨을 맞춰 주었다. 거리값에 상수를 곱하는 것은 결과에 영향을 주지 않는다. 그림에서와 같이 $d_{HINGE}^{(\lambda)}$와 d_{SIG}같은 경우 특정값에서 포화되는 형태를 가지고 있으며, d_{LOG}는 기울기가 점점 줄어드는 형태를 가지고 있다.

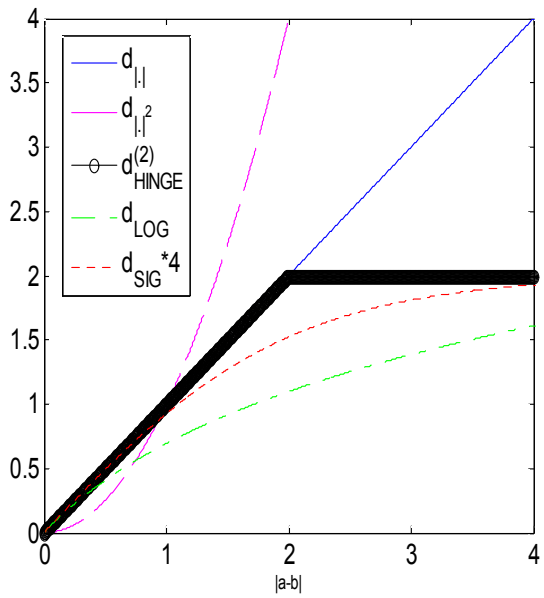

그림 2. 거리 함수 비교

4. 결과

실험에 사용한 DB에는 28시간 분량의 총 450곡의 노래가 들어가 있다. 여기에는 동요, 댄스, 팝, 트로트 등 다양한 장르의 노래가 들어가 있으며, 노래에 따라서 약 34초에서 396초 길이의 다양한 노래가 들어있다. 우리는 노래의 제목을 찾는 것을 목표로 하였으며, 오디오 인식에서와 같이 노래에서의 위치를 찾는 것은 목표로 하지 않았다 [11]. 테스트 데이터에는 총 1000개의 쿼리 입력이 들어가 있으며 각각은 약

10초에서 12초 정도의 길이를 가진다. 이 입력은 32명의 지원자에게서 모은 것이며 지원자는 14명의 여자와 18명의 남자로 구성된다. 1000개의 입력에는 298개의 허밍 입력과 702개의 싱잉 입력이 존재한다.

성능측정은 4가지 요소를 사용하였다. 먼저 1개의 노래제목을 결과로 냈을 때 정답이 검출되는 확률, 10개에서 검출되는 확률, 20개에서 검출되는 확률, 그리고 순위 역의 평균 (Mean reciprocal rank, MRR)을 사용하였다. MRR은 QbSH 시스템의 성능을 평가하기 위해서 많이 사용되는 지표이다.

거리	MRR	Top1	Top 10	Top 20
$d_{\|.\|}(.)$	0.627	0.569	0.646	0.794
$d_{\|.\|^2}(.)$	0.505	0.449	0.620	0.673
$d^{(1)}_{HINGE}(.)$	0.666	0.610	0.779	0.822
$d^{(2)}_{HINGE}(.)$	**0.713**	**0.668**	**0.808**	**0.843**
$d^{(3)}_{HINGE}(.)$	0.696	0.643	0.790	0.834
$d^{(4)}_{HINGE}(.)$	0.674	0.615	0.786	0.822
$d_{LOG}(.)$	0.674	0.618	0.780	0.814
$d_{SIG}(.)$	0.704	0.656	0.799	0.838

표 1. 거리 함수에 따른 결과

표에서 보이는대로 기존에 많이 사용되던 차이의 절대값과 그 제곱값은 좋지 않은 성능을 보인다. 제1후보에 대해서 검출 확률이 0.569에서 0.668로 약 0.1 정도 성능향상이 있었다. $d^{(2)}_{HINGE}(.)$가 실험에 사용된 거리 함수 중 가장 좋은 성능을 보였다. 거리 함수에 따라서 성능차가 많이 날 수 있음을 볼 수 있다.

5. 결론

본 논문에서는 DTW를 정합방법으로 사용하는 QbSH 시스템에서 거리 함수에 따른 성능 변화를 살펴 보았다. 본 논문에서는 기존의 차이의 절대값과 그 제곱 이외의 세 가지 거리 함수를 제안하고 성능을 비교한 결과 기존의 결과에 비해서 많은 향상이 있음을 알 수 있다. 이 결과는 우리의 QbSH 시스템에서 실험한 것으로 일반적이지 않을 수 있다. 하지만, 시스템에 따라서 다양한 거리 함수를 실험해 보는 것이 성능 향상에 도움이 된다는 것을 확인할 수 있었다.

6. 참고문헌

[1] J. S. Downie, "Music information retrieval," *Annual Review of Information Science and Technology*, 37:295-340, 2003.

[2] R. Typke, F. Wiering, and R. Veltkamp, "A survey of music information retrieval systems," *Proc. ISMIR,* 2005, pp. 153-160.

[3] A. Ghias, J Logan, and D Chamberlin, "Query by humming:musical information retrieval in an audio database," *Proc. of ACM Multimedia,* 1995, pp. 231-236.

[4] J. -S. R. Jang and M. Y. Gao, "A query-by-singing system based on dynamic programming," *International Workshop on Intelligent Systems Resolution (the 8th Bellman Continuum),* Hsinchu, Taiwan, pp 85-89, Dec. 2000.

[5] J. -S. R. Jang and H.-R. Lee, "A general framework of progressive filtering and its application to query by singing/humming," *IEEE Trans. on Audio, Speech, and language Processing,* vol. 16, no. 2, pp 350-358, Feb., 2008

[6] Y. Zhu and D. Shasha, ""Warping indexes with envelope transforms for query by humming,"" *In Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp. 181-192, 2003

[7] L. Wang, S. Huang, S. Hu, J. Liang, and B. Xu, "An effective and efficient method for query by humming system based on multi-similarity measurement fusion," *Proc. ICALIP*, 2008

[8] H. M. Yu, W. H. Tsai, and H. M. Wang, "A queryby-singing system for retrieving karaoke music," *IEEE Trans. on multimedia,* vol. 10, no. 8, 2008, pp. 1626-1637.

[9] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," *Proc. Int. Conf. Music Information Retrieval,* Bloomington, 2001, pp. 205-210.

[10] G. Poliner, D. Ellis, A. Ehmann, E. Gomez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Trans. Audio, Speech, Language Process.,* vol. 15, no. 4, pp. 1247.1256, Apr. 2007.

[11] D. Jang, C. D. Yoo, S. Lee, S. Kim, and T. Kalker, "Pairwise Boosted Audio Fingerprint," *IEEE Trans. Information Forensics and Security,* vol. 4, no. 4, pp. 995-1004, Dec. 2009.

[12] D. P. W. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," ICASSP 2007

[13] 김기출, 박성주, 이석필, 김무영, "선형 보간법을 이용한 시간과 주파수 조합영역에서의 피치 추정 방법," 전자공학회 논문지, 2010.

[14] 이세원, 윤제열, 심동규, 박성주, 이석필, 박호종, "예측 알고리즘을 이용한 다중 기본 주파수 측정 기술," 신호처리합동 학술대회 제22권 1호, 2009

[15] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," *IEEE Trans. Audio Speech Language Process.*, Vol. 16, No. 4, pp. 766-778, 2008.