다성 음악 신호의 템포 검출 기술

이동규 김기준 박호종
광운대학교 전자공학과
snapdog@kw.ac.kr

Tempo Detection of Polyphonic Music Signal

Donggyu Lee Kijun Kim Hochong Park
Dept. of Electronics Engineering, Kwangwoon University

요약

본 논문에서는 박자 분류 방법을 사용하여 다성 음악 신호의 템포 쌍을 검출하는 방법을 제안한다. 템포를 검출하는 방법은 음의 시작점을 추출하여 음악의 주기적인 흐름을 파악한 뒤, 그 주기를 템포로 변환하는 과정으로 구성된다. 제안한 기술은 템포로 추측되는 배수 관계의 템포 후보를 추출한 뒤, 템포 후보를 박자에 따라 분류하고 곡의 빠르기를 고려하여 최종 템포 쌍을 검출한다. 제안한 방법을 사용하여 높은 정확도로 템포 쌍이 검출되는 것을 확인하였다.

1. 서론

최근 음악 검색에 대한 관심이 높아짐에 따라 음악의 특징을 보다 정확하게 추출하기 위한 연구가 활발히 진행되고 있다. MIREX (Music Information Retrieval Evaluation eXchange)는 매년 음악 정보 분야별로 추출 알고리즘 성능에 대한 공개 평가를 진행하고 있다 [1]. 그 중 템포 (Tempo)는 2005년부터 평가 항목에 포함되어 있으며 음악의 3 요소로서 음악 검색에 있어서 중요한 특징 중에 하나이다.

템포는 주로 BPM(beats per minute)으로 나타내며 많은 템포 추출 알고리즘은 작곡가가 정한 템포와 비슷한 1개의 BPM만을 검출한다[2]. 대표적 방법으로 음의 시작점(onset)을 추출하기 위하여 DFT (discrete Fourier transform)로 스펙트럼을 구하고 스펙트럴 플럭스 (spectral flux)를 사용하여 검출 함수 (detection function)를 구한 뒤 자기 상관함수를 이용하여 템포를 찾는 방법이 개발되었다[3]. 그러나 악기는 주파수 대역 별로 다른 특성을 보이므로 대역을 나누는 과정이 필요하다. 이에 따라 입력 신호를 8개의 주파수 대역으로 나누어 각 대역별로 다운 샘플링 (down sampling)을 하고 자기 상관 함수를 구하여 가장 큰 3개의 피크 (peak) 값을 후보 피크로 정한 뒤, 최종적으로 1개의 템포를 검출하는 방법이 개발되었다[4]. 그러나 자기 상관 함수에서 템포를 결정하는 방법은 자기 상관 함수의 범위에 따라 최대 피크 값과 위치가 변하여 템포가 가변적으로 변하는 단점이 있다. 자기 상관 함수의 문제점을 해결하기 위하여 콤 템플릿 (comb template)을 사용하여 자기 상관 함수를 변형시키는 방법이 개발되었고[5], 이를 활용해 대역별로 자기 상관 함수를 구한 뒤 콤 템플릿을 적용한 방법이 개발되었다[6].

앞에서 언급한 알고리즘들은 모두 1개의 템포만을 검출하는데 1개의 템포를 그 음악의 빠르기라고 단정 짓기 어렵다. 이는 사람마다 같은 음악을 듣더라도 느끼는 빠르기가 다르며, 대다수의 사람이 인지한 템포가 작곡가가 정한 템포와도 다를 수 있기 때문이다. 실험을 통하여 같은 음악도 사람에 따라 템포를 다르게 인지한다는 사실을 확인하였고[7], MIREX의 평가 방식도 2개의 템포 쌍 (pair)을 찾는 방식을 선택하였다.

본 논문에서는 기존의 알고리즘과는 달리 검출 템포로 추측되는 배수 관계의 템포 후보를 추출한 뒤, 템포 후보를 음악의 예상 박자에 따라 분류(categorization)하고, 곡의 빠르기를 고려하여 최종 템포 쌍을 찾는 기술을 제안한다. 2장에서는 기존의 일반적인 템포 추출하는 방법에 대하여 설명하고 3장은 제안한 템포 쌍을 선택하는 방법, 4장에서는 제안한 방법의 측정 결과를 보여준다.

2. 기존 템포 추출 방법

템포를 추출하기 위하여 먼저 음의 시작점을 추출해야 한다. 이는 음의 시작점이 비트(beat)이며, 1분 동안의 비트수가 템포를 의미하기 때문이다. 이 때 추출된 음의 시작점들을 시작점 추출 함수라고 정의한다. 추출된 음의 시작점들의 주기를 검출하기 위하여 자기 상관 함수를 구하고 콤 템플릿(comb template)을 적용한 변형된 자기 상관 함수를 구하며 이를 주기 검출 함수라고 정의한다. 최종적으로 주기 검출 함수에서 템포 쌍을 추출한다.

입력된 신호를 50% 중첩을 가지는 92ms 단위의 프레임으로 나누어 윈도우를 적용한다. 이 후 프레임 단위로 N-포인트 DFT 하여 스펙트럼으로 나타낸다. 입력된 신호는 다성 음악으로 다양한 악기들이 존재하며 악기마다 주파수 대역에 따라 스펙트럼 성질이 다르다. 따라서 악기의 스펙트럼 특징을 대역별로 구분하기 위하여 3개의 대역으로 나눈다. 이 때 저대역은 0.2kHz 이하, 중대역은 0.2kHz ~5kHz, 고대역은 5kHz 이상으로 구분한다.

가. 시작점 추출 함수 (Onset detection function)

대역별로 효과적으로 시작점을 추출하기 위해서 대역별 특징에

따라 각기 다른 검출 방법을 사용한다. 저대역에서는 스펙트럼 에너지 변화뿐만 위상 정보를 활용하여 더 정확한 시작점을 추출할 수 있다[8]. DFT 계수는 식 (1)과 같다.

$$X(n,k)=\sum_{m=0}^{N-1} x(hn+m)w(m)e^{-2j\pi mk/N} \quad (1)$$

여기서 k는 주파수 변수, n은 프레임 번호를 의미하고 h는 홉 크기(hop size), $w(m)$는 윈도우, N은 DFT 크기를 나타낸다.

현재 프레임과 이전 프레임 사이의 에너지 차와 위상 차를 결합하면 식 (2)로 표현 할 수 있다[8].

$$\Gamma(n,k)=\begin{Bmatrix}[R(\widehat{X}(n,k))-R(X(n,k))]^2+\cdots \\ [I(\widehat{X}(n,k))-I(X(n,k))]^2\end{Bmatrix}^{1/2} \quad (2)$$

여기서 R과 I는 각각 실수부와 허수부를 나타내고, $\widehat{X}$는 이전 프레임의 DFT 계수를 의미한다. 따라서 현재 프레임의 OF(onset detection function)는 식 (3)으로 정의된다.

$$OF_{lb}(n)=\sum_{k=1}^{N/2}\Gamma_k(n) \quad (3)$$

이 때 lb은 저대역, n은 프레임 번호, N은 DFT 크기를 나타낸다.

중대역과 고대역에서는 스펙트럴 플럭스(spectral flux)를 사용하며, 식 (4)와 같다.

$$OF_{mb,hb}(n)=\sum_{k=1}^{N/2} H\left(|X(n,k)|-|\widehat{X}(n,k)|\right) \quad (4)$$

여기서 mb, hb는 각각 중대역, 고대역을 의미하고, $H(x)=(x+|x|)/2$ 를 의미하는 반파 정류 함수이다.

나. 주기검출함수 (Periodicity detection function)

앞에서 구한 시작점 추출 함수의 주기를 파악하기 위하여 대역별로 자기 상관 함수를 구하며 식 (5)로 표현된다.

$$r_i(D)=\sum_{n=1}^{L-D} OF(n)\,OF(n+D) \quad (5)$$

i는 대역을 의미하고, L은 OF의 길이, D의 범위는 BPM의 범위가 40 bpm에서 250 bpm 사이가 되도록 조절한다. 이는 비트(beat)간격이 너무 가깝거나 멀면 사람은 이를 템포로 인지하지 않기 때문이다.

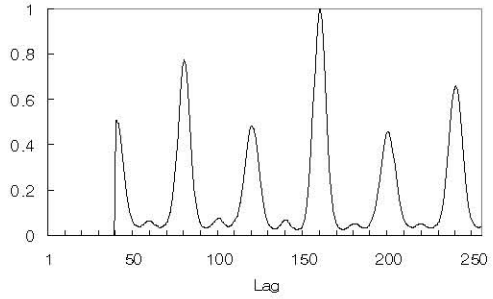
이렇게 구한 r값으로 콤 템플릿(comb template)을 사용하여 자기 상관 함수를 변형시킨 PF(periodicity function)를 구하며 식 (6)으로 정의된다[6].

$$PF_i(D)=\sum_{l=1}^{4}\frac{\max(r_i(D\times l+R))}{l} \quad (6)$$

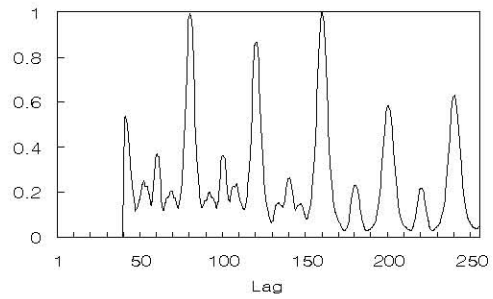
R = 1-l,…, l-1이다. 자기 상관 함수 D에서의 값과 D의 각 정수배에 위치하는 값의 합을 주기 검출 함수 D의 새로운 값으로 정의한다. 이 때 $r_i(D\times l)$는 R범위에 위치한 값 중에서 가장 큰 값을 정하며, l이 커질수록 더 큰 가중치로 나누어준다. 이렇게 대역별로 구한 PF를 정규화 한 후 더하면 최종 주기 검출 함수가 된다.

그림 1의 (a)와 (b)는 각각 자기 상관 함수와 주기 검출 함수를 나타내며, 가로축은 식 (5)와 (6)에서의 D를 의미한다. (a)와 (b)를 비교할 때, 피크 값의 크기가 바뀌면 최대 피크 값의 위치가 변한다. 이 때 템포는 식 (7)로 구한다.

$$tempo=\frac{fs\times 60}{\max Lag\times hopsize} \quad (7)$$



(a) 자기 상관 함수



(b) 주기 검출 함수

그림 1. MIREX 2011 practice data (train1)에 대한 자기 상관 함수와 주기 검출 함수

여기서 fs는 샘플링 주파수를 의미한다. 일반적으로 D가 1에서 40 사이일 경우에는 일반적으로 사람들이 템포로 인지하는 40 ~ 250BPM의 범위를 벗어난다. 그림 1을 예로 들면 샘플링 주파수가 44.1kHz, 홉 크기(hop size)가 256 샘플일 때 최대 피크 값을 갖는 래그(lag)는 160이므로 템포는 64.6BPM이다.

3. 제안한 템포 쌍 검출 방법

주기 검출 함수에서 최대값을 갖는 래그(lag)값을 선택하면 템포가 더블링 (doubling) 또는 하빙(halving)이 발생할 확률이 매우 높다. 그림 1의 (b)를 보면 래그가 160일 때 피크가 최대값을 가지지만, 80일 경우에도 상당히 큰 피크가 존재한다. 80번째 래그를 선택할 경우 템포는 129.1BPM이며, 이는 앞에서 구한 64.6 BPM의 더블링 된 템포이다. 그러나 Mckinney의 실험결과에 따르면 오히려 더 많은 사람들이 64.6BPM보다 129.1BPM을 선택하였다. 즉, 가장 큰 피크 값이 항상 템포는 아니지만 매우 높은 확률로 정답 템포의 배수에 위치함을 알 수 있다. 따라서 주기 검출 함수에서 피크 값과 위치 값을 동시에 이용하여 템포 쌍을 선택하는 방법이 필요하다.

제안한 방법에서는 템포 쌍을 선택하기 위하여 먼저 박자를 분류하는 과정을 진행한다. 2박자와 3박자를 구분하기 위해 템포 쌍 사이의 비율이 2배와 3배 차이를 갖는 경우로 분류(categorization)한다.

그림 2는 템포 쌍의 차이가 2배와 3배 차이가 나는 음악의 주기 검출 함수를 나타낸다. (a)의 경우 최대 피크 값 위치 160의 1/2, 1/4 위치인 80, 40에서 피크 값이 크며, 2/3, 1/3의 위치인 107, 53에서는 피크 값이 작다. (b)는 최대 피크값의 위치 150에서 2/3, 1/3의 위치에서 피크 값이 크며, 반대로 1/2, 1/4 위치에서는 피크 값이 작다. 이들을 2배와 3배 템포 쌍을 갖는 경우의 두 그룹으로 분류하여 템포로 추측되는 후보 피크로 지정한다. 이 때 최대 피크 위치에 따라 후보 피크의 위치도 변하게 된다. 그림 2의 (a)를 예로 들면, 최대 피크 위치가 80이라면 후보 피크의 위치는 1/2, 1/4 위치가 아닌 1/2, 2 위치가 된다. 이는 80의 1/4 위치인 20은 사람이 쉽게 인지할 수 있는 템포의 범위를 벗어나기 때문이다. 마찬가지로 최대 피크 위치가 40일 경우에는 2배, 4배 위치의 피크를 후보 피크로 지정한다. 이러한 경우에는 결과적으로 최대 피크의 위치에 관계없이 후보 피크가 같음을 알 수 있다.

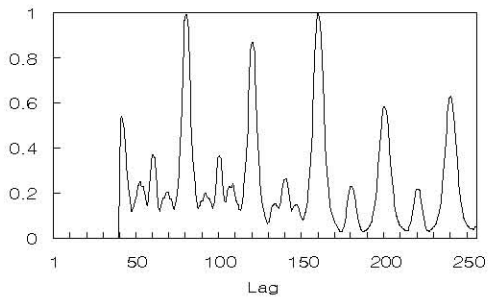
이와 같이 후보 피크 위치가 결정되면 피크 값과 위치를 동시에 이용하여 템포 쌍이 2배 또는 3배 차이를 갖는 경우인지 결정해야 한다. 이를 위하여 판별식을 정의하였으며 식 (8) 및 (9)와 같다.

$$TF_2 = \sum_{n=0}^{2} \frac{PF(cp_2(n))}{2^n} \tag{8}$$

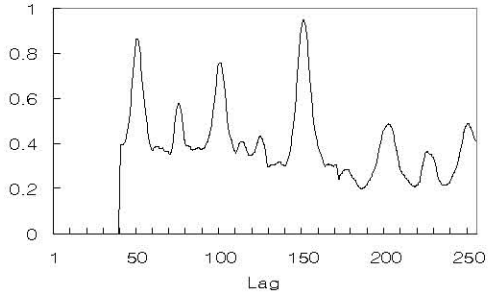
$$TF_3 = \sum_{n=0}^{2} \frac{PF(cp_3(n))}{3^n} \tag{9}$$

이 때 cp는 후보 피크(candidate peak)의 위치를 뜻하고 $cp(0)$이면 최대 피크의 위치이며 n이 증가할수록 최대 피크의 위치와 멀어지는 후보 피크의 위치이다. 표 1의 3배 템포 쌍을 예로 들면 $cp(0)$은 160, $cp(1)$은 107, $cp(2)$가 53이 된다. 그 후, TF_2가 크면 2배 템포 쌍, TF_3이 크면 3배 템포 쌍을 갖는 것으로 판별한다.

2배 또는 3배 템포 쌍을 갖는 것으로 판별되면 후보 피크 내에서 제 2의 템포를 찾아야 한다. 3배 템포 쌍을 갖는 경우로 판별되면 후보 피크 내에 3배 차이가 나는 템포 쌍이 한가지 밖에 존재하지 않으므로 추가 판단이 필요하지 않다. 그러나 2배 템포 쌍을 갖는 경우로 판단되면 또 한 번의 판단 과정을 거쳐야 한다.



(a) 템포 쌍의 차이가 2배인 주기 검출 함수



(b) 템포 쌍의 차이가 3배인 주기 검출 함수

그림 2. MIREX 2011 practice data (train1, train5)에 대한 주기 검출 함수 (a) train1 (b) train5

표 1. 그림 2 (a)의 박자 분류(categorization)에 의한 최대 피크와 후보 피크의 구분 예

	2배 템포 쌍일 경우	3배 템포 쌍일 경우
최대 피크 위치	160	160
후보 피크 위치	40, 80, 160	53, 107, 160

앞에서 언급했던 것처럼 템포는 BPM으로 표현되며, 이 수치가 클수록 빠르다는 것을 의미한다. 즉, 음악을 빠른 곡과 느린 곡으로 구분하여 곡의 빠르기에 따라 후보 피크 위치 중 2개를 선택한다.

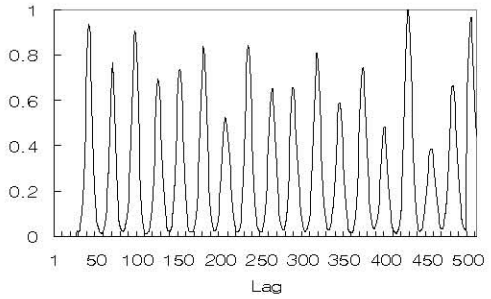
그림 3의 (a)와 (b)를 비교하면 피크 값과 개수가 차이가 나는 것을 확인 할 수 있다. 빠른 곡에는 비트(beat)가 빠르게 반복되므로 자기 상관 함수의 피크 개수도 많고 피크 값도 크다. 반대로 느린 곡은 비트가 천천히 반복되므로 피크의 개수도 적고 피크 값도 작다. 이러한 특성을 이용하여 빠른 곡과 느린 곡을 비교한다. 빠른 곡으로 판별 시에는 후보 피크 위치 중에 작은 위치 2개를 선택하고 느린 곡일 경우에는 후보 피크 위치가 큰 2개를 선택한다.

4. 성능 평가

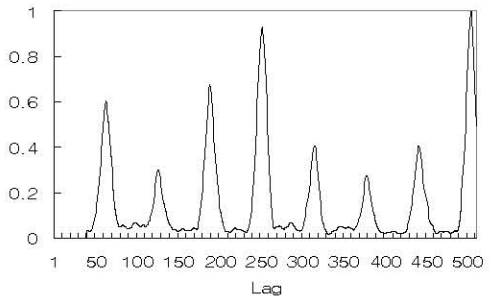
성능 평가를 위하여 MIREX 2011 practice data를 사용하였다. MIREX 2011 practice data는 총 20개, 30초 길이의 모노 데이터(44.1kHz 샘플링 주파수, 16bit PCM)로 구성되어 있다.

성능 평가 기준은 MIREX 2011 평가 기준과 동일하다. 정답 템포는 한 곡당 느린 템포와 빠른 템포로 구성되어 있는 템포 쌍이며, 검출된 템포쌍이 각각 정답 템포 쌍의 ±8% 이내의 범위일 경우에만 정답으로 인정한다. 또한 기존의 측정 방법들과는 달리 템포 쌍을 검출하므

로 [9]의 평가 방식처럼 더블링(doubling)과 하빙(halving) 템포는 정답으로 인정하지 않는다.



(a) 빠른 곡의 자기 상관 함수



(b) 느린 곡의 자기 상관 함수

그림 3. MIREX 2011 practice data (train19, train6)에 대한 곡의 빠르기 비교 (a) train19 (b) train6

대부분의 음악에서 높은 정확도로 템포 쌍을 찾는 것을 확인 할 수 있었다. 1개의 템포만 정답인 경우에도 다른 1개의 템포는 정답 템포의 더블링 또는 하빙 템포였다. 1개의 템포도 찾지 못한 음악의 장르는 클래식과 기타 연주로만 이루어진 락 음악이다. 이는 타악기 없이 멜로디 변화만 존재하여 주기적인 에너지 변화가 뚜렷하지 않기 때문이다.

표 2. 제안한 방법의 성능 평가 결과

At Least One Tempo Correct	Both Tempo Correct
85%	80%

5. 결론

본 논문에서는 다성 음악 신호의 박자를 분류하여 템포 쌍을 추출하는 기술을 제안하였다. 음의 시작점을 추출하여 주기적인 에너지 변화를 측정하였고, 이러한 변화에 대한 주기를 찾기 위하여 자기 상관 함수를 구한 뒤, 주기 검출 함수를 사용하여 후보 템포들을 분류하였다. 그 후 음악의 박자를 분류하여 2배 또는 3배의 템포 쌍을 갖는 경우로 판별하고 곡의 빠르기를 고려하여 최종 템포 쌍을 추출하였다. 기존의 템포 추출 알고리즘과는 달리 분류된 후보 템포 중에서 2개의 템포를 추출하여 음악에 대한 보다 구체적인 정보를 얻을 수 있었다. 제안한 방법을 사용하면 타악기가 포함되어 있는 음악의 템포의 검출 성능은 우수하지만 현악기만으로 연주되는 음악의 템포 검출 성능은 낮다. 이와 같은 문제점을 해결하여 연주되는 악기에 관계없이 우수한 템포 검출 성능을 보이는 연구를 할 예정이다.

참고문헌

[1] 김무영, 이석필, "MIREX 기술 동향," 전자공학회지, 제37권, 제1호, 88-102쪽, 2010년 1월

[2] J. Zapata and E. Gomez, "Comparative Evaluation and Combination of Audio Tempo Estimation Approaches," *Audio Engineering Society Conference: 42nd International Conference: Semantic Audio*, Ilmena, Germany, 2011.

[3] M. Alonso, B. David, and G. Richard, "Tempo and beat estimation of musical signals," in *Proc. Int. Conf. Music Information Retrieval*, 2004.

[4] S. Dixon and E. Pampalk, Classification of dance music by periodicity patterns," in *Proc. International Conference on Music Information Retrieval*, pp. 159-165, 2003.

[5] M. E. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 3, pp. 1009.1020, Mar. 2007.

[6] M. Gainza and E. Coyle, "Tempo Detection using a hybrid multi-band approach," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 2009.

[7] M. McKinney and D. Moelants, "Deviations of the resonance theory of tempo induction," *In Proceedings of the Conference of Interdisciplinary Musicology*, Graz, 2004.

[8] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "Complex domain onset detection for musical signals," in *Proc. 6th Int. Conf. on Digital Audio Effects*, London, U.K., Sep. 2003.

[9] F. Gouyon, A. Klapuri, S. Dixon, et al., "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 5, pp. 1832-1844, 2006.