피크레벨을 이용한 비트 분석 및 음악 장르구분

김윤호* · 조재영*
*목원대학교 IT 공학부

Analysis of Bit and music genre using Peak Level

Yoon-ho Kim* · Jae-young Jo*
*Mokwon University Div. of computer multimedia eng.
E-mail : yhkim@mokwon.ac.kr & racemaster@mokwon.ac.kr

요 약

기존의 음악 재생기들은 폴더별 저장 등의 방법을 통해 사용자가 임의로 장르구분을 해서 저장한 경우에 한하여 장르별로 음악을 들을 수가 있다. 본 논문에서는 드럼 등의 타악기 소리에서 검출되는 피크레벨 주파수를 시간별 피크레벨수로 분석하여 빠른 곡과 느린 곡의 구분을 사용자가 직접 해줄 필요 없이 재생기 내에서 버튼 하나로 구분되어 재생될 수 있도록 하는 것을 목표로 한다.

ABSTRACT

This report shows the new music player's system which could separate music's tempo by analysis Peak level frequency by time from some percussion instruments. After this process, the new music player could classify some fast or slow genre's musics without a user's order, then we should listen to a fast or slow genre's music by a button.

키워드

오디오, 주파수, 피크레벨, 비트

Ⅰ. 서 론

아날로그 신호에서 디지털 신호로 전자제품들의 변화가 일어나면서 음악을 저장하는 매체나 그 매체를 재생하는 재생기의 종류도 다양해졌다. 과거의 아날로그 테입이나 턴 테이블 같은 아날로그 저장매체는 CD, MD, DVD 등의 디지털 매체로 변화되었고 그 새로운 매체의 재생을 위한 재생기 또한 많은 변화와 발전을 거듭했다. 특정 저장매체를 사용하지 않고 재생기 내의 메모리에 MPEG1에서 규정한 압축기술로 압축된 압축데이타를 실시간으로 해제하여 들을 수 있는 MP3 플레이어 또한 많은 사용자를 확보하고 있는 재생기의 대표 중 하나다. 또한 청취자 자신의 취향에 맞게 CD-Recordable를 통한 mp3나 wav 파일의 기록 및 청취가 가능하게 되었다. 하지만 청취자가 CD-R을 통해 자신이 원하는 음악파일들을 저장한다고 해도 기존의 재생기 에서는 한곡의 반복 플레이, 또는 음악을 무작위 플레이 시키는 렌덤 플레이등의 기능만이 있었으며 최근 MP3의 확산으로 인한 다수곡의 재생을 위해 청취자가 폴더별로 저장해놓은 곡들을 폴더별로 청취하는 것들이 가능하게 되었다. 하지만 그러기 위해서는 물론 곡을 저장할 때 곡들을 직접 선곡하여 폴더별로 분류해야하는 단점이 있다. 본 연구에서는 거의 모든 장르의 대중음악에서 검출되는 타악기 소리에 의해서 형성되는 피크레벨 주파수를 특정 초당 피크레벨의 수로 측정 및 분석하여 사용자가 폴더별 저장 등의 방법을 거치지 않고도 무작위로 저장시킨 다수의 음악들을 대상으로 사용자가 원할 때 버튼 하나만으로 빠른곡과 느린곡으로 분류가 가능한 피크레벨 측정 프로그램을 구현하는 것을 목적으로 한다.

II-1. 피크레벨의 정의 및 표현

녹음작업이 완료 된 mp3나 wav 파일에서 진폭이 크게 나타나는 부분을 피크레벨이라 한다.

일반적으로 드럼의 베이스 북소리와 같은 경우 주파수가 낮은 반면 진폭은 크다. 그런 베이스 북소리와 같은 소리들이 녹음될 경우 높은 진폭으로 표현되는데 그 진폭의 꼭대기 부분이 한계에 근접하거나 넘어갈 경우 음의 왜곡이 생긴다. 보통 모든 녹음작업을 마친 후에 노멀라이즈라는 작업을 해주는 것이 보통인데 이 작업은 피크레벨이 있는, 즉 가장 큰 크기의 진폭을 가진 레벨의 피크레벨 값이 한계값을 넘지 않는 범위내에서 최대한 볼륨을 키워주는 작업을 의미한다 [1][2].

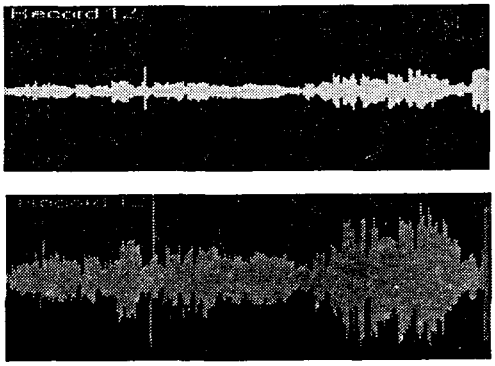

그림 1. 피크레벨값의 표현과 해당 피크레벨 기준의 노멀라이즈의 예

일정시간동안의 피크레벨의 개수를 측정하기 위해서는 우선 어느 레벨의 진폭을 가진 주파수부터를 피크레벨로 정의할 것인가가 문제다. 따라서 본 연구에서는 노멀라이즈를 했다는 전제하에 최고점에 이르는 주파수를 100%로 보고 시작점의 주파수를 0으로 본 상태에서 최고 레벨의 주파수 값 10% 내의 주파수 까지를 비트측정 피크레벨로 정의하기로 한다. 노멀라이즈는 가장 높은 진폭이 한계값을 넘지 않는 범위내에서 최대한의 볼륨값의 크기를 늘려주기 때문에 베이스 드럼등의 소리를 기준으로 노멀라이즈 되는 경우가 대부분이다. 대부분의 대중가요는 곡의 비트를 결정짓는 드럼등의 타악기 소리가 피크레벨을 형성하고 모든 대중가요는 장르의 구분없이 그런 타악기 소리가 포함되어있다. 그러므로 주파수의 최고점을 측정하면 예외 없이 그 곡의 비트를 결정짓는 타악기나 전자악기음이 측정된다[3][4]. 물론 녹음환경에 따라서 예상치 않은 잡음이 피크레벨을 형성하는 경우도 있지만 대중가요 음반으로 제작되어 있는 음악파일의 경우 최상의 녹음 환경에서 작업되었기에 그런 외부 잡음이 피크레벨을 형성하는 경우는 드물다.

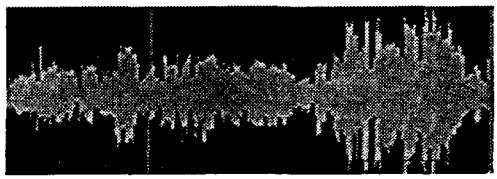

그림 2. 측정 피크레벨값의 표현

기존의 발라드곡의 비트는 일반적으로 70 에서 90 사이의 비트를 사용한다. 반면에 빠른 비트의 대표라고 할 수 있는 댄스나 힙합 장르의 곡은 110에서 250비트 사이의 비트를 사용하는 경우가 대부분이다. 90에서 110까지의 중간영역의 비트는 많이 쓰이지 않는 비트 이므로 본 연구의 실험대상에서는 제외시켰다. 실험에 쓰이는 곡은 70에서 90 사이의 비트를 사용한 곡 30곡과 110에서 250 비트를 사용한 곡 30곡을 섞어서 사용한다.

II-2. 피크레벨 측정 및 방법

대부분의 대중가요 곡에는 드럼이나 전자타악기의 소리가 포함되어 있다. 하지만 피크레벨을 형성하는 타악기의 강한 소리가 존재하는 구간은 시간적인 관점에서 볼 때 곡의 처음부터 끝까지 항상 존재하는 것은 아니다. 댄스나 힙합등의 장르의 곡들의 경우에는 처음부터 피크레벨을 형성하는 강한 타악기의 소리들이 존재하는 경우도 흔히 볼 수 있지만 발라드나 락발라드 등의 느린 곡의 경우 처음 도입부나 중반부까지는 타악기 소리 없이 피아노나 스트링 등의 악기소리만으로 구성되어져 있는 경우가 더 많은것을 실험을 통해 알 수 있었다. 하지만 본 연구에서 사용된 모든 곡들을 조사한 결과 발라드와 락발라드 장르의 모든 곡들이 2분에서 2분 30초 사이에 곡의 클라이막스 및 간주 부분이 삽입되어 있었고 그 클라이막스나 간주 부분에 있어서는 100% 피크레벨이 생성되는 타악기 소리가 포함되어 있는것을 알 수 있었다. 그 외 댄스나 힙합등의 빠른곡은 곡의 도입부부터 타악기 소리가 포함되어 있는 곡이 14곡, 중반부부터 포함되어 있는 곡이 16곡이었다. 그러므로 실험 대상이 되는 곡에 있어서 피크레벨 측정 범위는 각 곡의 2분에서 2분 30초 사이의 피크레벨 수를 측정하는 방법을 사용하였다.

시스템 구현을 위해서 개인용 컴퓨터(pentium-4 1.80GHz, 512RAM)에 오디오 매직 2 사운드/오디오 카드를 설치하였다. 운영체제는 Window XP로 하였으며, 피크레벨 측정 프로그램 구현을 위한 사용언어는 JAVA SDK_1.4를 사용하였다. WAV 파일 노멀라이즈나 기타 WAV 파일의 편집을 위해서 디지털 오디오 편집기인 Cool Edit 2000와 Cakewalk Sonar2.0 버전을 사용하였다. 피크레벨 측정 프로그램은 WAV 파일

의 전처리 과정, 피크레벨 수 측정 과정, 비트수에 따른 곡 분류과정으로 나누어지며 이를 그림 3에 나타내었다.

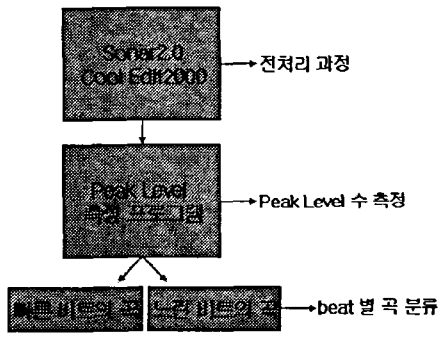

그림 3. 피크레벨 측정 방법 블럭도

전처리 과정(Pre-processing)은 환경 적응, 시작점 검출, 반향 제거, 잡은 제거 와 같은 오디오 신호를 본격적인 분석과정에 들어가기 전에 처리하는 과정이다. 전처리 과정을 거친 WAV 파일은 자바로 구현된 peak level 측정 프로그램을 통해 곡의 2분에서 2분30초에 해당하는 부분에 한하여 피크레벨을 측정하게 된다. 실험 방법은 두 가지 설정을 두었는데 그 중 한 가지는 2분에서 2분 30초까지의 30초간의 피크레벨의 수를 측정하는 방법이었고 또 다른 하가지 방법은 2분 10초에서 2분 15초까지의 5초간의 피크레벨의 수를 측정하는 방법이었다. 후자의 경우 CD player에서 곡을 검색하는 시간을 줄일 수 있으므로 같은 시간에 더 많은 곡의 분류가 가능하다는 장점이 있다 [5][6].

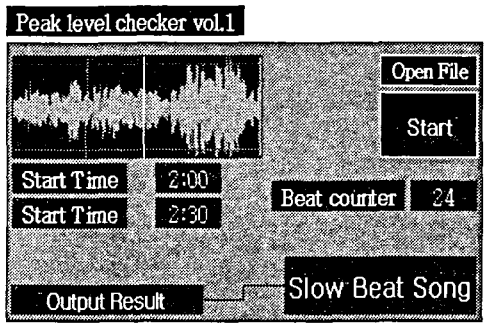

그림 4. 피크레벨 측정 프로그램

우선 사용자는 open file 버튼을 눌러 원하는 wav 파일을 불러들인다. 그 후 피크레벨 개수 체크가 시작 될 시점과 끝나는 시점을 지정해준다. 마지막으로 start 버튼을 누르면 피크레벨 측정 프로그램은 타임지시선이 피크레벨 체킹 끝점에 도달하는 순간까지(위 그림에서는 2분 30초) 10배속의 빠르기로 파일을 읽는다. 그 후 사용자가 지정해준 범위내의 측정된 피크레벨 개수를 beat counter 표시란에 표시한 후 그 개수를 분석하여 output result 표시란에 빠른 곡인가 느린 곡인가의 여부를 표시한다. 만약 두 결과의 범주에 포함되지 않은 경우 error 라는 표시가 뜬다 [7][8][9][10].

III. 결론

본 연구에서는 타악기 소리 입력 시 생기는 피크레벨 주파수의 개수 측정을 통해 곡의 빠르기를 측정하는 프로그램을 구현하였다. 표.1은 실험을 통한 결과로 2분에서 2분30초의 범위를 측정했을 경우 빠른 비트의 곡과 느린 비트의 곡 모두 93퍼센트 이상의 높은 성공률을 보였고 2분10초에서 2분 15초의 범위를 측정했을 경우 전자의 실험결과에 약간 못 미치는 약 86퍼센트 이상의 성공률을 보였다. 이는 느린 비트의 곡에 있어서 좀 더 높은 오류율이 있다는 것을 보여준다. 짧은 범위의 측정의 경우 약간의 오차율이 큰 영향을 미친다는 것을 보여주는 결과라 할 수 있다. 30초의 범위를 측정했을 경우 평균 피크레벨 개수와 5초의 범위를 측정했을 경우 평균 피크레벨 개수는 항상 해당 초에 비례적으로 증감하지 않는다는 것을 보여주는 결과이기도 하다. 음악파일 속에 섞여있는 잡음이라든지 약간의 의도적이지 않은 소음들로 인한 피크레벨 증가현상이 생길 수 있기 때문이다. 빠른 비트의 경우 250의 비트수에 가까운 곡은 드물기 때문에 잡음이나 소음들에 의한 피크레벨이 생긴다 하여도 250 비트수의 피크레벨 개수를 벗어나는 경우가 없어 실험결과에는 영향을 미치지 않았으나 느린 비트의 경우 단 5초의 피크레벨 개수를 측정하여 결과를 내는 상황에서 단 하나의 잡음으로 인한 피크레벨 개수의 증가는 중간비트 영역의 오류 비트수를 가지는 경우가 생겼다. 또한 피크레벨 측정 전 전처리 과정의 한 부분인 노멀라이즈 과정에서 타악기 소리 중 유난히 큰 소리가 있을 경우 그 피크레벨을 기준으로 노멀라이즈 처리가 되기 때문에 다른 타악기 소리들이 피크레벨의 범주에 들어가지 못하게 되는 경우도 있었다.

표 1. 피크레벨 특성 분석

측정영역 / 템포	2:00 ~ 2:30 (min. sec)	2:10 ~ 2:15 (min. sec)
Slow beat	93%	86%
Fast beat	100%	100%
Middle beat (error beat)	13%	6%

그 문제는 전처리 과정에서의 노멀라이즈 과정 전에 미리 너무 높은 파형을 가지는 피크레벨은 미리 sonar 프로그램 상에서 3DB Quieter 과정을 거쳐 평균화 시켜주는 작업을 해줌으로서 해결할 수 있었다. 본 실험에서의 피크레벨 개수 측정 프로그램은 정해진 범위의 시작시간까지 10배속이지만 그래도 플레이를 해야 한다는 단점이 있었으나 향후 연구를 통해 측정할 필요가 없는 곡의 도입부는 플레이 할 필요 없이 정해주는 범위의 시작점과 끝점만을 바로 측정할 수 있게 한다면 시간적인 측면에서 커다란 이점을 가질 수 있을 것이라 본다. 또한 일정 시간동안의 피크레벨의 총 개수만을 가지고 곡의 빠르기를 결정하는 방법이 아닌 일정하지 않은 피크레벨 주기를 분석 및 파악하여 향후 과제로는 단지 느린비트와 빠른비트의 구분만을 하는 프로그램이 아닌 신경망학습과 퍼지이론 등의 도입으로 각 곡들의 연주패턴이나 비트특성들을 분석하여 그 분석된 데이터를 바탕으로 곡들의 장르별 분류 까지도 가능하게 하는 프로그램을 구현하는 것이 향후 과제로 남아있다. 마지막으로 기존의 시디 플레이어나 mp3 플레이어등의 재생기 구동원리를 파악하여 선 분류 후 감상의 방식이 아닌 음악을 듣는 중에 장르별 분류가 가능할 수 있도록 하는 것을 최종 목적으로 한다.

참고문헌

[1] Sohubha Kadambe & G. Faye Boudreau (1992), Applications of Wavelet Transforms for Pitch Detection of Speech Signals, IEEE Trans on Information Theory 38.

[2] W. Hess (1983), Pitch Determination of Speech Signal, Spriger-verlag

[3] Wong et al. (1979), Least squares glottal inverse filtering from the acoustic speech waveform, IEEE. Trans. ASSP 27,pp.350~355

[4] Wilpon, J. G., L. R. Rabiner and T. B Martin (1984), An improved word-detection algorithm for telephone-quality speech incorporating both synthetic and semantic constraints, AT&T Tech, pp.476~497

[5] 박일서, 김대현, 조철우(2001), 실시간 음성분석도구의 Matlab 구현, 창원대학교,

[6] 지정규, 오해석(1997), 선율 입력에 의한 음곡선 검색 알고리즘, 한국정보과학회,

[7] Triton, 'Music workstation/sampler-Voice Name List',KORG

[8] Rack ATTACK User's manual, waldorf, 2002,

[9] 1604-VLZ PRO 16-channel owner's Manual, 2000

[10] PC2 Kurzweil Musician's Guide, 2002