
Music Information Retrieval(MIR)을 활용한 음악적 리듬의 시각화 연구

Onset 검출(Onset Detection) 알고리즘에 의한 시각화 어플리케이션

A Study on Visualization of Musical Rhythm Based on Music Information Retrieval

*최수환, Swann Che

요약 이 글은 Music Information Retrieval(MIR) 기법을 사용하여 오디오 콘텐츠의 리듬 정보를 자동으로 분석하고 이를 시각화하는 방법에 대해 다룬다. 특히 MIR을 활용한 간단한 시각화(sound visualization) 어플리케이션을 소개함으로써 음악 정보 분석이 디자인, 시각 예술에서 다양하게 활용될 수 있음을 보이고자 한다. 음악적 정보를 시각 예술로 담아내려는 시도는 20세기 초 아방가르드 화가들에 의해 본격적으로 시작되었다. 80년대 이후에는 컴퓨터 기술의 급속한 발전으로 사운드와 이미지를 디지털 영역에서 쉽게 하나로 다룰 수 있게 되었고, 이에 따라 다양한 오디오 비주얼 예술작품들이 등장하였다. MIR은 오디오 콘텐츠로부터 음악적 정보를 분석하는 DSP(Digital Signal Processing) 기술로 최근 디지털 콘텐츠 시장의 확장과 더불어 연구가 활발히 진행되고 있다. 특히 웹이나 모바일에서는 이미 다양한 상용 어플리케이션이 적용되고 있는데 query-by-humming과 같은 음악 인식 어플리케이션이 대표적인 경우이다. 이 글에서는 onset 검출(onset detection)을 중심으로 음악적 리듬을 분석하는 알고리즘을 살펴보고 기본적인 조형원리에 따라 이를 시각화하는 어플리케이션의 예를 소개한다.

핵심어: *Sound Visualization, Music Information Retrieval, Rhythm Analysis, Onset Detection, Audio-Visual*

1. 서론

음악 정보를 시각 예술로 변환하려는 시도는 20세기 초 아방가르드 화가들에 의해 시작되었다. 특히 몬드리안과 같은 De Stijl 화가들은 음악이 순수하게 추상적인 요소들로 구성되어 있음에 주목하고 화성법, 대위법과 같은 음악의 규칙에서 조형 요소들을 구성하는 원리를 찾으려고 노력하였다.[1]

Music Information Retrieval(MIR)은 디지털 신호 처리(DSP, Digital Signal Processing)의 한 분야로 오디오 콘텐츠 분석과 관련하여 최근 활발히 연구되고 있다. 특히 웹이나 모바일에서는 이미 상용 어플리케이션에 많이 적용되고 있는데 query-by-humming과 같은 자동 음악 인식 어플리케이션이 대표적인 경우이다.

이 글은 MIR 기법을 사용하여 오디오 콘텐츠로부터 비트 추출 및 리듬 분석을 자동으로 실행하고 이를 시각화하는 방법에 대해 살펴 본다. MIR과 관련한 연구가 주로 상용 어플리케이션 적용을 목적으로 하는 것과 달리 이 글은 MIR 기법의 예술적 적용 가능성에 초점을 맞추고 있다. 따라서 MIR로 분석된 리듬 정보를 조형 원리에 따라 시각적으로 재구성하는 방법들을 주로 다룬다.

2. 음악적 리듬

리듬은 다양한 변수들이 혼합된 개념이기 때문에 정확하게 정의하기는 힘들다. 리듬의 어원은 그리스어 $\rho\nu\theta\mu\omicron\zeta$ (rhythm)과 $\rho\epsilon\omega$ (to flow)이다. 형태를 의미하는 그리스어 Rhythmos는 이오니아 철학의 중요한 용어 중 하나였다. Rhythmos는 특정한 방식의 흐름을 의미하는데 플라톤은 수를 사용하여 표현할 수 있는 몸의 움직임이라고 하였다. 그는 『향연』에서 “시스템은 빠름과 느림의 결과이다. 처음에는 반대쪽에서 시작하지만 차츰 조화를 이룬다.” 라고 하였고, 『국가』에서는 리듬을 “움직임 속의 질서” 라고 정의하였다.[2]

리듬은 일반적으로 음악의 기본이 되는 규칙적인 펄스 또는 음악에서 전반적으로 유지되는 반복적 패턴을 의미한다. 이때 일정한 간격으로 발생하는 펄스를 비트라고도 하며, 비트의 강약이 변화하여 만들어지는 패턴을 박자라고 한다. 박자 시스템은 서양 음악의 중요한 요소 중 하나이지만 일부 현대음악이나 민속음악에서는 박자 개념이 없는 경우도 있다.

Barratt는 리듬은 ‘반복되지만 어떤 두 요소도 동일하지 않은 비트 혹은 흐름’ 이라고 정의하며, 전통적으로 예술 작품에 리듬감을 부여하기 위해서는 규칙성 속에서 의미 있는 변화가 있어야 한다고 말한다.[3]

컴퓨터를 사용한 리듬 분석을 위해 Blimes는 박자 구조(metric content), 박자가 없는 구절(ametrical phrases), 빠르기 변화(tempo variation), 이벤트 오차(event shifting)의 네 가지를 리듬의 요소로 제시했다.[4] 이 중에서 음악적 리듬의 박자 구조는 일정한 간격을 지닌 시간축의 그리드에 표시할 수 있는 연속적인 리듬 이벤트를 말한다. 서양 음악의 기보법은 박자 구조를 표현하는 대표적인 예이며, 음악 소프트웨어에서 사용하는 시퀀서(sequencer), 퀀타이저(quantizer) 등도 박자 구조를 활용한다. 앞서 말한 것처럼 서양 음악은 박자 시스템에 기반한 리듬 구조를 가지고 있으므로 박자 구조의 분석이 서양 음악의 리듬 분석을 위해서는 필수적인 사항이라고 할 수 있다.

한편 Toussaint은 기하학적 관점에서 음악적 리듬을 분석하였다.[5] Toussaint은 특히 아프로-쿠바 음악의 클라베(clave)¹⁾ 리듬에 주목하여 이를 기하학적 형태로 표현하여 그 구조를 파악하고자 하였다. Toussaint은 4/4 박자를 가진 클라베의 onset을 16등분한 원에 배치한 후 onset 간 간격으로 리듬의 수학적 구조를 도형으로 나타내었다.

3. MIR 기법을 사용한 리듬 분석

MIR은 오디오 콘텐츠로부터 음악적 정보를 추출, 분석하는 기술로 오디오 신호를 매개변수화(parametrization)하여 얻어지는 특성 벡터(feature vector)를 패턴 매칭(pattern matching)등의 기법으로 분석한다.

3.1 리듬 분석 알고리즘

컴퓨터를 사용한 음악적 리듬의 자동 분석은 80년대 이후 연구되기 시작하였다. 초기에는 제한적인 컴퓨터 성능으로 인해 MIDI와 같이 기호화된 음악 데이터에 대한 분석이 주로 이루어 졌으나 최근에는 DSP를 사용하여 오디오 데이터를 직접 분석하는 것이 가능해졌다. 음악적 리듬의 분석을 위해서는 시간 영역, 스펙트럼 영역의 다층적인 분석이 필요한데 비트 추적(beat tracking) 알고리즘은 가장 기본적인 분석 방법 중 하나이다.

Hainsworth는 비트 추적 알고리즘을 그 목적에 따라 규칙기반 기법(rule-based), 자동상관 기법(autocorrelative), 오실레이팅 필터 기법(oscillating filter), 히스토그램 기법(histogramming), 다중 에이전트 기법(multiple agent), 확률적 기법(probabilistic) 등으로 분류하였다.[6]

Scheirer는 공진 콤파터(resonating comb-filter)를 사용한 비트 추적 알고리즘을 제시하였다.[7] 이 알고리즘은 입

1) 클라베(clave)는 살사(salsa), 룸바(rumba) 등과 같이 아프로-쿠바 음악에 자주 사용되는 리듬 패턴이다.

력 오디오 신호를 필터뱅크(filterbank)를 사용하여 6개의 대역으로 나눈 후 인벨롭 검색기(envelope detector)를 사용하여 각 밴드의 데이터를 줄인다. 이후 대역별로 약 150여개의 콤팩터를 사용하여 오디오 신호의 빠르기(주기)를 검출해 낸다.

Goto는 비트 추적을 위해 다중 에이전트(multiple agent) 모델을 사용하였다.[8] 입력 신호는 필터뱅크를 거쳐 대역별로 분리된 후 이산 onset 검출(discrete onset detection)에 의해 IOI(inter-onset interval) 벡터로 변환된다. 그리고 IOI 벡터의 자동상관, 교차상관(cross correlation)에 의해 리듬 패턴을 유도하고, 다중 에이전트를 사용하여 최적의 빠르기를 산출한다.

Klapuri는 Hidden Markov Model(HMM)을 사용한 알고리즘을 제안하였다.[9] Klapuri의 알고리즘은 Scheirer와 유사한 평가 단계를 거치지만, HMM을 사용하여 과거의 데이터를 현재의 빠르기 산출에 반영한다.

정확한 리듬 분석을 위해서는 알고리즘의 정확도 뿐만 아니라 음악 스타일에 대한 이해가 필요하다. 예를 들어 레게나 스카 음악의 경우 오프비트(offbeat)에 강박이 발생하는데 이러한 정보를 미리 입력할 경우 비트 검출 알고리즘의 정확도를 더 높일 수 있다. 따라서 리듬 분석 알고리즘의 정확도를 높이기 위해 콘텐츠로부터 자동으로 분석되는 데이터 외에 수동으로 입력된 메타데이터를 활용하는 경우도 있다.

3.2 Onset 검출 알고리즘

Onset은 소리에서 음량이 0인 지점에서 최초 최대점까지 증가하는 시작 부분을 말하는데 소리의 전이 상태에 해당하는 부분이다. onset 검출은 음악 신호의 분석과 인덱싱 기법에서 중요한 역할을 하며 리듬 비트의 검출에 효과적으로 사용할 수 있다.[10]

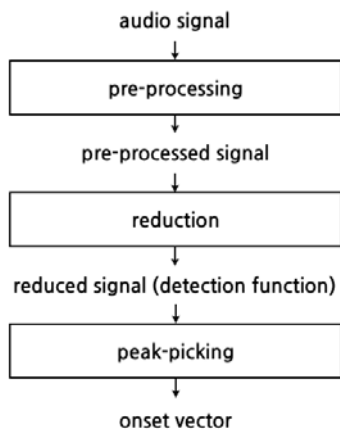


그림 1. onset 검출 알고리즘의 흐름도

일반적으로 onset 검출 알고리즘은 그림1과 같이 3가지 단계로 이루어진다. 입력 신호는 노이즈 제거와 같은 전 처리기(pre-processor)를 거친 후 특성 추출(feature extraction)과 같은 reduction 기법에 의해 검출 함수(detection function)를 도출한다. 검출 함수에서 매 프레임에 해당하는 시간의 y값은 onset의 확률로 해석할 수 있는데 최고점을 찾아내는 peak-picking 알고리즘으로 onset의 위치를 시간 축에 표시할 수 있다.

3.2.1 전처리(preprocessing)

onset 검출 알고리즘에서 전처리는 원본 신호의 특정한 측면을 분석 목적에 따라 증폭시키거나 제거하는 것을 의미한다. 따라서 알고리즘에 따라 전처리는 선별적으로 적용할 수 있는데 일반적으로 다중대역 분할과 transient / steady-state 분리가 많이 적용된다.

많은 onset 검출 연구에서 다중 대역 분할을 사용하고 있는데 이는 onset 검출을 부 대역(sub-band)으로 나눔으로써 검출 능력을 높이기 위함이다. 다중 대역 분할을 위해서는 주로 필터뱅크를 많이 사용하며 Goto, Scheirer, Klapuri 등이 이러한 방식을 사용하였다.

transient/steady-state 분리는 오디오에서 음색 요소를 제거하기 위해 필요하다. 예를 들어 오르간의 경우 sustain 구간이 길기 때문에 onset 검출이 힘들 수 있는데 이러한 음색 요소를 제거함으로써 onset 검출을 위한 transient 상태만 남기는 전처리를 하게 된다.

3.2.2 Reduction

Onset 검출에서 reduction은 오디오 신호를 검출 함수(detection function)로 변환하는 것으로 onset 검출을 위해 필요한 부분(전이 상태) 이외의 데이터를 제거하는 과정이다. reduction은 크게 신호 특성(signal feature)을 이용하는 방법과 확률적 모델을 사용하는 방법으로 나눌 수 있다.

MIR에서 많이 사용하는 특성(feature)는 크게 시간 특성(temporal feature)과 스펙트럼 특성(spectral feature)로 나눌 수 있다. 시간 특성은 오디오 신호의 크기(amplitude)나 에너지 및 그 변화값(미분값)을 사용하여 비교적 간단하게 구현이 가능하며, 스펙트럼 특성은 Short-Time 푸리에 변환(STFT)이나 MFCC¹⁾ 등의 방법을 사용한다. 여기서는 각각 시간 특성, 스펙트럼 특성을 대표하는 envelope follower와 STFT 기법에 대해 살펴보도록 한다.

1) MFCC(Mel-Frequency Cepstral Coefficients), 인간의 청각 시스템을 반영한 mel-scale에 기반한 스펙트럼 분석 방법으로 음성 인식 알고리즘에서 중요한 역할을 한다.

1) 인벨롭 추적(envelope following)

간단한 신호의 경우 신호의 크기 변화를 관찰함으로써 onset을 쉽게 찾아낼 수 있다. 초기의 onset 검출 방법은 신호의 크기 변화를 추적하는 검출 함수를 사용하였는데 이는 envelope follower에 해당한다. envelope follower는 입력 신호를 정류(rectification)한 후 smoothing(low-pass filtering)함으로써 쉽게 구현할 수 있다.

$$E_0(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} |x(n+m)|w(m) \quad (1)$$

$w(m)$: 중심 $m = 0$ 인 N -포인트 윈도우(smoothing 커널이라고도 한다.)

신호의 크기를 추적하는 방법 대신 다음과 같이 에너지를 추적하는 방법을 사용할 수도 있다.

$$E_0(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} [x(n+m)]^2w(m) \quad (2)$$

envelope follower에 의해 얻어진 검출 함수는 smoothing을 사용했지만 여전히 최대값 선택(peak picking)에 의한 onset 검출에 적합하지 않다. 표준 onset 검출 알고리즘에서는 에너지를 그대로 사용하지 않고 에너지의 미분값을 사용한다. 이 경우 onset에서 흔히 발생하는 에너지의 급격한 변화가 미분값에서 꼭지점으로 나타나게 된다. 그 외에 방법으로 소리 크기가 로그함수로 인지된다는 음향심리학의 실험 결과를 이용하기도 한다.

2) Short-Time Fourier Transform

스펙트럼 특성을 사용한 검출 함수는 전처리를 간소하게 하고 여러 개의 악기로 되어 있는 다성(polyphonic) 신호의 분석에 적합하다. 입력 신호 $x(n)$ 에 대하여 Short-Time 푸리에 변환(STFT)는 다음과 같이 나타낼 수 있다.

$$X_k(n) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} x(nh+m)w(m)e^{-\frac{2j\omega mk}{N}} \quad (3)$$

$w(m)$: N -포인트 윈도우, h : 근접한 윈도우 간의 hop(time shift)의 크기.

스펙트럼 영역에서 전이 상태와 관련된 에너지의 증가는 광대역으로 나타나는 경향이 있다. 신호의 에너지는 대체로 저역대에 집중되어 있기 때문에 전이 상태에 의한 에너지 변화는 고역대에서 더 쉽게 관찰할 수 있다. 이런 특성을 이용하기 위해 다음과 같이 주파수에 따라 가중치를 적용할 수 있다.

$$\tilde{E}(n) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} W_k |X_k(n)|^2 \quad (4)$$

W_k : 주파수에 따른 가중치

4. 시각화 어플리케이션의 구현

이 장에서는 앞서 살펴본 onset 검출 알고리즘을 활용한 사운드 시각화(sound visualization) 어플리케이션의 예를 소개한다. 사운드 시각화 어플리케이션의 구조는 그림과 같다.

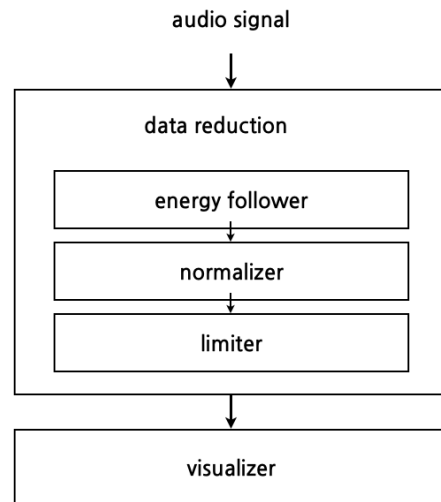


그림 2. 사운드 시각화 어플리케이션의 구조

onset 검출을 통해 얻어진 리듬 비트 정보는 행렬 또는 벡터로 저장된다. 이 예에서 사용한 onset 검출 알고리즘은 입력 신호를 대역별로 분리하지 않았으므로 reduction을 거친 검출 함수는 벡터 형태로 나타난다. reduction의 단계에서는 입력 신호의 에너지를 추적(energy following)한 후 노멀라이징(normalizing) 및 한계값(threshold)을 이용한 리미팅(limiting) 과정을 거쳤다. 에너지 추적에서는 Hanning

윈도우를 사용한 필터링을 적용하였다.

입력 신호 샘플의 인덱스를 i , 검출 함수의 샘플 인덱스를 j 라고 할 때, i 와 j 는 다음과 같은 관계를 가진다.

$$i = h \times j \quad (5)$$

h 는 검출 함수의 벡터 크기를 줄여주는 매개변수로 프레임 크기로도 이해할 수 있다. 예를 들어 320BPM¹⁾의 빠르기를 가진 음악의 경우 32분음표에 해당하는 시간은 0.0234초로 샘플링 비가 44,100Hz인 오디오 신호의 경우 약 1,024 프레임에 해당한다. 즉, $h=1,024$ 에서 320BPM인 음악의 32분음표 단위까지 비트의 검출이 가능함을 의미한다. 앞에서 소개한 Scheirer, Goto, Klapuri의 경우 비트 검출 이후 BPM과 관계있는 주기 산출을 위한 알고리즘을 포함하고 있지만 이 알고리즘은 onset(비트)의 검출만 다룬다. 검출된 onset 벡터는 기본적으로 위치(벡터의 인덱스, 샘플 프레임)와 크기에 대한 정보를 가진다. 다중 대역이나 다중 특성을 사용할 경우 n 차원 벡터를 얻을 수 있다.

사운드 시각화(sound visualization) 어플리케이션은 검출 함수인 onset 벡터를 기본적인 조형 요소들로 재구성한다. 이 어플리케이션은 Processing API를 사용하여 Java로 구현되었다. Processing은 자바 기반의 언어로 다양한 드로잉 API가 제공되어 프로그래밍에 익숙하지 않은 시각 디자이너가 쉽게 사용할 수 있는 어플리케이션 개발 환경이다.[11]

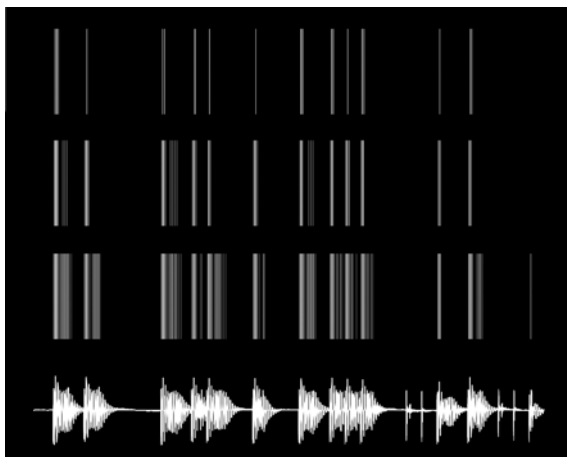


그림 3. onset 검출의 예. 513 샘플의 Hanning 윈도우, 256 샘플의 hop 크기가 사용되었다. 맨 아래의 이미지가 원본 오디오의 파형이고, 위에서부터 각각 0.7, 0.5, 0.3의 한계값(threshold)이 사용되었다.

1) BPM(Beat Per Minute), 음악의 빠르기를 나타내는 단위로 분당 비트 수를 의미한다. 일반적인 곡의 빠르기는 120BPM인데 이는 분당 4분음표 120개가 연주되는 빠르기에 해당한다.

시각화를 위한 조형 요소로는 점, 선, 면과 같은 기본 도형을 사용하였고 구성의 명확함을 위해서 그레이 모드를 적용하였다. 각 조형 요소는 벡터 또는 행렬의 값에 따라 각각 회전, 크기, 간격, 명도 등의 속성을 조정하였다.

〈그림 3〉은 검출된 onset을 밝기가 다른 수직선으로 시각화한 경우이다. 리미터(limiter)의 한계값 설정에 따라 onset 검출의 정확도가 달라지며 상대적으로 많은 에너지를 가지고 있는 저역대 onset의 검출이 더 정확하다. onset 검출의 정확도는 이 외에도 윈도우의 모양, 크기, hop 크기 등에 의해서도 많은 차이를 나타낸다.

〈그림 4〉-〈그림 9〉는 onset 검출 알고리즘으로 추출된 onset 벡터를 시각화한 예이다. 〈그림 4〉, 〈그림 5〉는 onset 벡터의 크기를 수직선의 밝기나 길이로 적용한 경우이다. 〈그림 6〉~〈그림 9〉는 onset 벡터를 그리드 형태로 시각화하였는데, 각각 셀의 폭, 원의 밝기, 셀의 회전 각도 등으로 적용하였다.

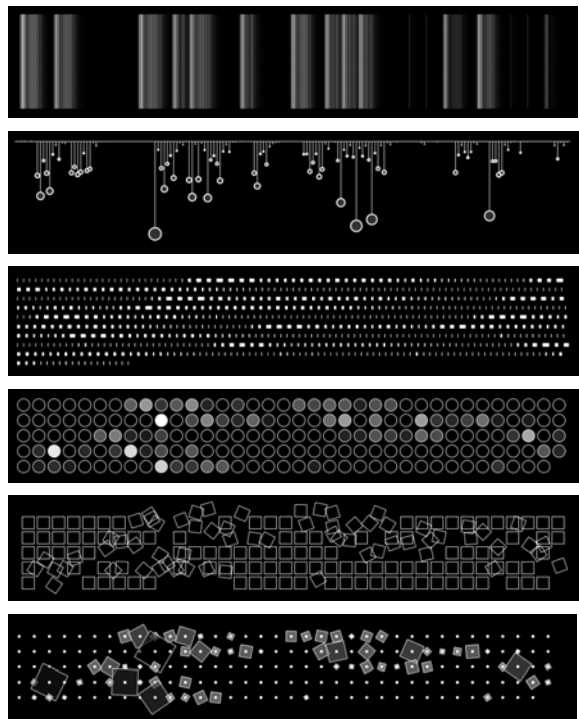


그림 4-그림 9. 검출된 onset에 대한 다양한 시각화의 예.

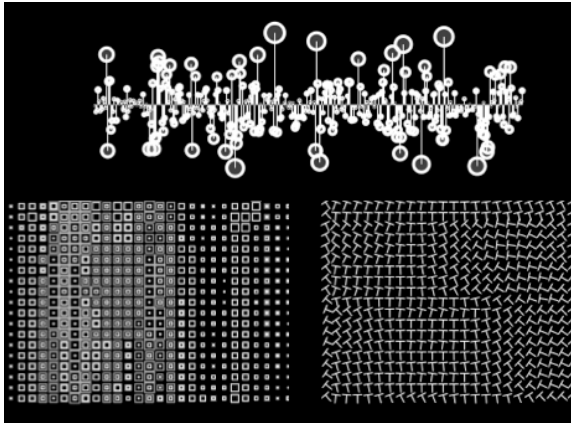


그림 10. 푸리에 변환을 사용한 사운드 시각화의 예. 푸리에 변환 후 대역별로 각 프레임의 에너지를 자동상관(autocorrelation) 알고리즘을 사용한 후 시각화하였다.

5. 결론

이 글에서는 리듬 정보 분석을 위한 다양한 MIR 기법들을 살펴보고 onset 검출 알고리즘을 활용한 사운드 시각화 어플리케이션의 예를 통해 디자인, 시각 예술 분야에서의 활용 가능성을 검토해 보았다. 이 글에서 사용한 사운드 시각화 어플리케이션은 다양한 시각적 표현 가능성에 초점을 두었기 때문에 일반적인 MIR 알고리즘이 추구하는 분석의 정확도 문제는 다루지 않았다. 따라서 실질적인 적용을 위해서는 정확도, 연관성과 같은 알고리즘의 타당성에 대한 검토도 필요하다. 또한 음악적 리듬은 onset으로 대표되는 시간적 구조 뿐만 아니라 화성과 같은 스펙트럼 구조와도 관련이 있기 때문에 보다 의미있는 사운드 시각화 알고리즘을 위해서는 오디오에 대한 다층적인 분석이 필요하다. <그림 10>은 푸리에 변환을 통해 얻어지는 스펙트로그램(spectrogram)을 시각화한 것으로 이러한 예에 해당할 것이다.

현재 MIR에 대한 연구 결과는 상용 어플리케이션에서 많이 적용되고 있어 그 연구 규모나 범위가 확대되고 있다. 이러한 연구들은 주로 웹, 모바일 환경에서 디지털 콘텐츠의 활용을 쉽게 하는 것이 목적이기 때문에 음악 정보 분석의 정확도를 높이는 방향으로 이루어지고 있다. 하지만 20세기 초 아방가르드 화가들처럼 사운드와 이미지의 보편적인 미를 예술적으로 구현하려는 시도들도 최근 미디어 아트 영역에서 많이 나타나고 있다.

리듬은 사운드와 이미지에서 공통으로 발견되는 중요한 속성이다. 따라서 이를 분석하고 변환하는 알고리즘은 음악 뿐만 아니라 디자인, 시각 예술 등에서 다양하게 활용될 수 있다. 그리고 이러한 알고리즘을 구현할 때 각 분야가 구축해 놓은 리듬, 더 나아가서 시간을 다루는 원리에 대한 이해는 중요한 역할을 할 것이다.

참고문헌

- [1] K. V. Maur, *The Sound of Painting*, Prestel Books, New York, USA, 1999.
- [2] D. Deutsch, *Psychology of Music*, Academic Press, San Diego, USA, 1999.
- [3] K. Barratt, *Logic & Design: in Art, Science and Mathematics*, Design Books, USA, 1980.
- [4] J. Blimes, "A Model for Musical Rhythm", *Proceedings of the International Computer Music Conference*, 1992.
- [5] G. Toussaint, "The Geometry of Musical Rhythm", *Proceedings of the Japan Conference on Discrete and Computational Geometry*, 2005.
- [6] S. W. Hainsworth, "Techniques for the Automated Analysis of Musical Audio", PhD thesis, University of Cambridge, UK, 2004.
- [7] E. Scheirer, "Tempo and Beat Analysis of Acoustic Musical Signals", *Journal of the Acoustical Society of America*, vol. 103, no. 1, 1998.
- [8] M. Goto, "A Study of Real-time Beat Tracking for Musical Audio Signals", PhD thesis, Waseda University, 1998.
- [9] A. Klapuri, "Signal processing methods for the automatic transcription of music", PhD thesis, Tampere University of Technology, 2004.
- [10] J. P. Bello et al., "A Tutorial on Onset Detection in Music Signals", *IEEE Transactions on Speech and Audio Processing*, 2005.
- [11] C. Reas, B. Fry, *Processing: A Programming Handbook for Visual Designers and Artists*, MIT Press, Cambridge, USA, 2007.