



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0130833
(43) 공개일자 2009년12월24일

(51) Int. Cl.

G06F 3/00 (2006.01) G10H 1/26 (2006.01)

G06F 9/00 (2006.01)

(21) 출원번호 10-2009-0053420

(22) 출원일자 2009년06월16일

심사청구일자 없음

(30) 우선권주장

12/140,133 2008년06월16일 미국(US)

(71) 출원인

임머슨 코퍼레이션

미국, 캘리포니아 95131, 산조세 폭스레인 801

(72) 발명자

울리히, 크리스토퍼 제이.

미국 93003 캘리포니아주 벤투라 팔로마스 에비뉴 227

랭크, 스티븐 디.

미국 95119 캘리포니아주 산 호세 바히아 코트 201

바키르시오글루, 무니메 엠.

미국 95130 캘리포니아주 산 호세 파세오 올리보스 5023

(74) 대리인

양영준, 백만기

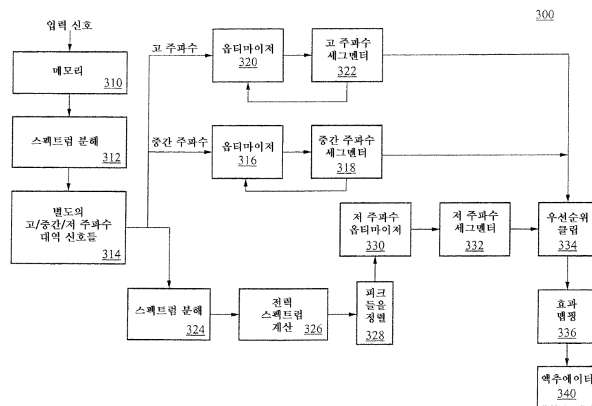
전체 청구항 수 : 총 26 항

(54) 디지털 오디오 파일로부터 햅틱 이벤트들을 자동으로 생성하는 시스템 및 방법

(57) 요약

일 실시예에서, 복수의 채널에서의 구조화된 표현 시퀀스의 복수의 이벤트를 햅틱 이벤트들로 자동으로 변환하는 시스템 및 방법이 개시된다. 그 방법은 하나 이상의 채널에서의 시퀀스의 각 이벤트에 대한 이벤트 스코어를 계산하는 것을 포함한다. 또한, 그러한 방법은 하나 이상의 채널에서의 이벤트 스코어들에 기초하여 누적 스코어를 계산하는 것을 포함한다. 그 방법은 하나 이상의 선택된 채널에서의 이벤트 스코어들에 기초하여 햅틱 이벤트들을 이벤트에 선택적으로 지정하는 것을 포함하며, 햅틱 이벤트들은 햅틱 액추에이터에 의해 출력된다. 이것은 햅틱 이벤트들을 효율적으로 생성하기 위해, 시스템이 사운드의 속성들을 계산하거나 또는 그러한 속성들과 연관된 기존의 값들을 취함으로써 행해질 수 있다.

대표도



특허청구의 범위

청구항 1

복수의 주파수 서브-대역 신호들로 특성화가능한 데이터를 갖는 디지털 오디오 파일을 처리하는 단계;

적어도 하나의 서브-대역에서 복수의 오디오 이벤트들을 식별하는 단계; 및

상기 이벤트들 중 적어도 하나에 햅틱(haptic) 효과를 할당하여, 액추에이터(actuator)에 의해 출력될 햅틱 이벤트를 생성하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 디지털 오디오 파일은 상기 복수의 오디오 이벤트들이 식별되는 상기 주파수 서브-대역 신호들을 포함하는 방법.

청구항 3

제2항에 있어서,

상기 디지털 오디오 파일은 Mp3 포맷으로 된 방법.

청구항 4

제1항에 있어서,

상기 오디오 파일을 처리하는 단계는 오디오 신호의 스펙트럼 분해(spectral decomposition)를 계산하는 단계를 더 포함하는 방법.

청구항 5

제4항에 있어서,

상기 주파수 서브-대역 신호들 중 적어도 하나를 상기 복수의 이벤트들로 분할하는(segmenting) 단계, 및 이벤트들 사이에 최소 분리 시간을 지정하는 단계를 더 포함하는 방법.

청구항 6

제4항에 있어서,

상기 분할하는 단계는 상기 햅틱 이벤트 이전에 최소 액추에이터 시동 시간을 지정하는 단계를 더 포함하는 방법.

청구항 7

제4항에 있어서,

상기 분할하는 단계는 복수의 햅틱 이벤트들에서의 각각의 햅틱 이벤트에 대해 최소 지속 기간(minimum duration of time)을 지정하는 단계를 더 포함하는 방법.

청구항 8

제1항에 있어서,

상기 할당하는 단계는 이벤트들 사이에 최소 분리 시간을 지정하는 단계를 더 포함하는 방법.

청구항 9

제1항에 있어서,

상기 할당하는 단계는 제1 햅틱 이벤트 이전에 최소 액추에이터 시동 시간을 지정하는 단계를 더 포함하는 방법.

청구항 10

제4항에 있어서,

상기 서브-대역 신호로부터 분할된 상기 이벤트들은 초당 햅틱 효과들의 사용자 지정 평균 수(user specified average number)에 대응하는 방법.

청구항 11

제4항에 있어서,

상기 서브-대역 신호로부터 분할된 상기 이벤트들은 최대 앨리어스 타겟 값(maximum alias target value)보다 작은 앨리어스 값을 갖는 방법.

청구항 12

제4항에 있어서,

하나 이상의 서브-대역 신호들의 파라미터에 대해 지정된 시간 기간에 걸쳐 평균을 계산하는 단계를 더 포함하는 방법.

청구항 13

제12항에 있어서,

상기 적어도 하나의 서브-대역 신호를 분할하는 단계는,

상기 서브-대역 신호의 파라미터의 단기 값(short term value)이 상기 파라미터에 대한 상기 평균의 배수를 초과하는 경우 이벤트를 지정하는 단계를 더 포함하는 방법.

청구항 14

제2항에 있어서,

상기 오디오 신호의 상기 스펙트럼 분해의 전력 스펙트럼을 계산하는 단계;

상기 전력 스펙트럼에 적어도 부분적으로 기초하여 주파수를 선택하는 단계; 및

상기 선택된 주파수에 적어도 부분적으로 기초하여 이벤트를 식별하는 단계를 더 포함하는 방법.

청구항 15

제1항에 있어서,

상기 햅틱 이벤트 데이터는 네트워크를 통해 클라이언트와 통신할 수 있는 서버 상에 저장되고, 상기 서버는 상기 햅틱 이벤트를 상기 클라이언트에게 전달하는 방법.

청구항 16

복수의 주파수 서브-대역 신호들로 특성화가능한 데이터를 갖는 디지털 오디오 파일을 처리하는 회로;

적어도 하나의 서브-대역에서 복수의 오디오 이벤트들을 식별하는 회로;

상기 이벤트들 중 적어도 하나에 햅틱 효과를 할당하여 햅틱 이벤트를 생성하는 회로; 및

상기 햅틱 이벤트를 출력하도록 구성된 액추에이터

를 포함하는 장치.

청구항 17

제16항에 있어서,

상기 서브-대역 신호들 중 적어도 하나를 상기 복수의 오디오 이벤트들로 분할하는 회로를 더 포함하며,
상기 분할하는 회로는 오디오 신호의 스펙트럼 분해를 계산하는 수단을 더 포함하는 장치.

청구항 18

제17항에 있어서,
상기 분할하는 회로는 이벤트들 사이에 최소 분리 시간을 지정하는 수단을 더 포함하는 장치.

청구항 19

제17항에 있어서,
상기 분할하는 회로는 제1 햅틱 이벤트 이전에 최소 액추에이터 시동 시간을 지정하는 수단을 더 포함하는 장치.

청구항 20

제17항에 있어서,
상기 분할하는 회로는 복수의 햅틱 이벤트들에서의 각각의 햅틱 이벤트에 대해 최소 지속 기간을 지정하는 수단을 더 포함하는 장치.

청구항 21

제16항에 있어서,
상기 할당하는 회로는 이벤트들 사이에 최소 분리 시간을 지정하는 것을 더 포함하는 장치.

청구항 22

제16항에 있어서,
상기 서브-대역 신호 내의 상기 이벤트들은 초당 햅틱 효과들의 사용자 지정 평균 수에 대응하는 장치.

청구항 23

제16항에 있어서,
상기 서브-대역 신호 내의 상기 이벤트들은 최대 앨리어스 타겟 값보다 작은 앨리어스 값을 갖는 장치.

청구항 24

제16항에 있어서,
하나 이상의 서브-대역 신호들의 파라미터에 대해 지정된 시간 기간에 걸쳐 평균을 계산하는 수단을 더 포함하는 장치.

청구항 25

제24항에 있어서,
상기 분할하는 회로는 상기 서브-대역 신호의 파라미터의 단기 값이 상기 파라미터에 대한 상기 평균의 배수를 초과하는 경우 이벤트를 지정하는 것을 더 포함하는 장치.

청구항 26

제16항에 있어서,
상기 오디오 신호의 상기 스펙트럼 분해의 전력 스펙트럼을 계산하는 회로;
상기 전력 스펙트럼에 적어도 부분적으로 기초하여 주파수를 선택하는 회로; 및
상기 선택된 주파수에 적어도 부분적으로 기초하여 이벤트를 식별하는 회로를 더 포함하는 장치.

명세서

발명의 상세한 설명

기술분야

<1> 관련 출원

<2> 본 출원은 본원과 공동 소유된 미국 가특허 출원 제60/792,137호(2006년 4월 13일자로 제출되고, 발명자가 Stephen D. Rank 및 Christopher J. Ullrich이고, 발명의 명칭이 "Music to Haptic Code Conversion"임)에 기초한 우선권의 이익을 주장하는 미국 특허 출원 제11/515,243호(2006년 8월 31자 제출됨)의 일부 계속 출원이다.

<3> 기술분야

<4> 본원에 개시된 주제는 일반적으로 햅틱 분야에 관한 것으로, 더 구체적으로는 디지털 오디오 또는 오디오/시각 파일에 대응하는 햅틱 트랙이나 햅틱 효과 제어 신호를 생성하는 것에 관한 것이다.

배경 기술

<5> 사용자 인터페이스 디바이스에서는 종종 촉각 진동(tactile vibrations)이 바람직하다. 촉각은 공간 감각: 세계를 통해 자신의 길을 조종하기 위해 사용하는 것-시각, 청각 및 촉각을 단일화한다. 촉각은 밀리초(ms) 단위로 반사-레이트 응답을 생성하여 시각과 청각이 함께 대체할 수 없는 완전함을 제공한다.

<6> 사용자 인터페이스 디바이스에서는, 예를 들어 음악 파일이 휴대폰에서 재생되고 있을 때 촉각 진동들이 사운드와 일치하게 하는데 유용하다. 최종-사용자 디바이스들에 대한 오디오는 2가지 기본 방식: 재생되는 사운드들을 공지의 심볼들의 스트링으로 부호적으로 인코딩하는 프로토콜(예를 들어, MIDI(Musical Instrument Digital Interface)과 같은 구조화된 표현 포맷으로; 및 오디오 파형의 주기적인 샘플들의 세트인 디지털 샘플링된 오디오로서 인코딩될 수 있다.

<7> 현재, 샘플링된 오디오 파일들은 수동 변환을 통해 변환될 수 있다. 인간 설계자는 샘플링된 오디오 파일을 듣고 분석하고 판단하여 햅틱 효과들을 생성하기 위한 샘플링된 오디오의 특징들을 선택한다. 그러나, 인간 설계 간섭은 부자연스러운 리소스이다.

발명의 내용

해결 하고자하는 과제

<8> 음악 파일들을 햅틱 효과들로 변환하는 전자동 또는 반자동 방법이 요구된다.

과제 해결수단

<9> 일 실시예에서, 복수의 주파수 서브-대역 신호들로 특징화가능한 데이터를 갖는 디지털 오디오 파일을 처리하는 단계; 적어도 하나의 서브-대역에서 복수의 오디오 이벤트를 식별하는 단계; 및 적어도 하나의 이벤트에 햅틱 효과를 할당하여 액추에이터에 의해 출력되는 햅틱 이벤트를 생성하는 단계를 포함하는 방법이 제공된다.

<10> 일 실시예에서, 복수의 주파수 서브-대역 신호들로 특징화가능한 데이터를 갖는 디지털 오디오 파일을 처리하는 회로; 적어도 하나의 서브-대역에서 복수의 오디오 이벤트를 식별하는 회로; 및 적어도 하나의 이벤트에 햅틱 효과를 할당하여 햅틱 이벤트를 생성하는 회로; 및 햅틱 이벤트를 출력하도록 구성된 액추에이터를 포함하는 장치가 제공된다.

<11> 일 실시예에서, 디지털 오디오 신호를 조작하는 장치로서, 상기 오디오 신호를 복수의 서브-대역 신호로 분리하는 수단; 상기 서브-대역 신호들 중 적어도 하나를 복수의 이벤트로 분할하는 수단; 및 적어도 하나의 이벤트에 햅틱 효과를 할당하여 액추에이터에 의해 출력되는 햅틱 이벤트를 생성하는 수단을 포함하는 장치가 제공된다.

<12> 일 실시예에서, 상기 방법 및 시스템은 오디오 신호의 스펙트럼 분해를 계산한다. 일 실시예에서, 상기 방법 및 시스템은 이벤트들 사이에 최소 분리 시간을 지정한다. 일 실시예에서, 상기 방법 및 시스템은 제1 햅틱 이벤트 이전에 최소 액추에이터 시동 시간을 지정한다. 일 실시예에서, 상기 방법 및 시스템은 복수의 햅틱 이벤트에서의 각 햅틱 이벤트에 대해 최소 지속기간을 지정한다. 일 실시예에서, 상기 방법 및 시스템은 이벤트들

사이에 최소 분리 시간을 지정한다. 일 실시예에서, 상기 방법 및 시스템은 제1 햅틱 이벤트 이전에 최소 액추에이터 시동 시간을 지정한다. 일 실시예에서, 서브-대역 신호 내의 이벤트들은 초당 햅틱 효과들의 사용자 지정 평균 수에 대응한다. 일 실시예에서, 서브-대역 신호 내의 이벤트들은 최대 앨리어스 타겟 값 아래의 앨리어스 값을 갖는다. 일 실시예에서, 상기 방법 및 시스템은 하나 이상의 서브-대역 신호의 파라미터에 대해 지정된 시간 기간에 걸쳐 평균을 계산한다.

<13> 일 실시예에서, 상기 방법 및 시스템은 서브-대역 신호의 파라미터의 단기 값이 파라미터에 대한 평균의 배수를 초과하면 이벤트를 지정한다. 일 실시예에서, 상기 방법 및 시스템은 오디오 신호의 스펙트럼 분해의 전력 스펙트럼을 산출하고; 전력 스펙트럼에 적어도 부분적으로 기초하여 주파수를 선택하고; 선택된 주파수에 적어도 부분적으로 기초하여 이벤트를 식별한다. 일 실시예에서, 상기 방법 및 시스템은 네트워크를 통해 클라이언트와 통신할 수 있는 서버에 햅틱 이벤트를 저장하며, 상기 서버는 햅틱 이벤트를 상기 클라이언트에 전달한다.

발명의 실시를 위한 구체적인 내용

<14> 본 명세서의 일부에 포함되고 그 일부를 구성하는 첨부 도면은 본 발명의 하나 이상의 실시예들을 도시하고, 상세한 설명과 함께, 발명의 주제의 원리 및 구현을 설명한다.

<15> 주제의 실시예들은 자동 오디오-햅틱 변환을 위한 시스템의 맥락에서 본원에 기술된다. 본 기술분야의 통상의 기술자는, 본원의 다음의 상세 설명은 단지 예시일 뿐, 임의의 방법으로 제한하는 것으로 의도되지 않는다는 것을 이해할 것이다. 본원의 다른 실시예들은 본 개시의 이익을 갖는 기술자들에게 그 실시예들 자체를 쉽게 제안할 것이다. 첨부하는 도면들에 도시된 바와 같이 본원의 구현들에 대한 참조가 상세히 이루어질 것이다. 동일한 참조 표시는, 동일 또는 유사 부분들을 지칭하기 위해, 도면들 및 다음의 상세 설명에 걸쳐 사용될 것이다.

<16> 명료성을 위해, 본원에 기재된 구현들의 일상의 특징들이 모두 도시되고 기술되는 것은 아니다. 물론, 임의의 그러한 실제 구현의 개발에 있어, 응용 및 사업 관련 제약들에 따르는 등의 개발자들의 특정 목표들을 성취하기 위해, 다수의 구현-특정 결정들이 이루어져야 하며, 이러한 특정 목표들이 하나의 구현에서 다른 구현까지, 그리고 한 개발자에서 다른 개발자까지 다르다는 것이 이해될 것이다. 또한, 그러한 개발 노력은, 복잡하고 시간 소모가 되지만, 그럼에도 불구하고, 본원의 이익을 갖는 본 기술분야의 통상의 기술자들에게 있어 엔지니어링의 일상 책임일 수 있다는 것이 이해될 것이다.

<17> 본원에 기술된 주제에 따르면, 컴포넌트들, 시스템들, 디바이스들, 프로세스 단계들 및/또는 데이터 구조들은 다양한 유형의 운영 시스템, 컴퓨팅 플랫폼들, 컴퓨터 프로그램들 및/또는 범용 머신들을 사용하여 제조, 동작 및/또는 구현될 수 있다. 부가적으로, 본 기술분야의 통상의 기술자들은, 하드와이어드(hardwired) 디바이스들, FPGA(Field programmable gate array)들, ASIC(application specific integrated circuit)들 등과 같은 범용 성질이 적은 디바이스들이 본원에 기술된 발명 개념들의 범위 및 사상에서 벗어남 없이 사용될 수 있다는 것을 이해할 것이다. 포함하는 방법이 컴퓨터, 디바이스 또는 머신(machine)에 의해 구현되고, 그 방법이 머신에 의해 판독가능한 명령어들로서 저장될 수 있는 경우, 명령어들은, ROM(Read Only Memory), PROM(Programmable Read Only Memory), EEPROM(Electrically Erasable Programmable Read Only Memory, 플래시 메모리, 점프 드라이브 등), 자기 저장 매체(예를 들어, 테이프, 자기 디스크 드라이브 등), 광 저장 매체(예를 들어, CD-ROM, DVD-ROM, 종이 카드 및 종이 테이프 등) 및 다른 공지된 유형의 프로그램 메모리를 포함하지만 이에 한정되지 않는 컴퓨터 메모리 디바이스와 같은 유형 매체 상에 저장될 수 있다. 부가적으로, 그 방법은 인간적 또는 창조적 판단을 요구하지 않고 소프트웨어 도구들의 선택을 사용하여 인간 오퍼레이터에 의해 실행될 수 있다는 것이 이해되어야 한다.

<18> 일 실시예에서, 오디오 파일은, 시스템이 주파수 서브-대역 신호들뿐만 아니라 적합한 햅틱 이벤트들을 출력하는데 사용되는 그와 연관된 계수들(coefficients)을 처리, 식별 및 산출하는 샘플링된 디지털 또는 디지털화된 아날로그 오디오를 포함한다. 일 실시예에서, 본원에 기술된 시스템 및 방법은 디지털 파일들을 수신하고, 디지털 오디오 파일 내에서 이미 주파수 서브-대역 계수들과 동기화된 햅틱 이벤트들을 자동으로 출력하는 것에 관한 것이다. 이러한 시스템에 의한 덜 계산적인 집약적 접근법은, 시스템이, 실시간 및/또는 오프-라인 모드로 햅틱 이벤트들을 자동으로 출력할 수 있게 한다. 다른 실시예에서, 수신된 파일들은 구조화된 표현 파일들(예를 들어, MIDI 파일들)이다. MIDI 파일들이 구조화된 표현 파일들(structured representation files)의 관점에서 본원에 기술되었지만, 임의의 다른 프로토콜이 시스템에 의해 사용될 수 있고, 이에 의해, 프로토콜은 오디오, 비디오 또는 다른 신호들을 출력할 수 있는 구조화된 표현 파일들을 활용한다는 것을 주목해야 한다. 변환된 햅틱 이벤트들은, 사용자가, 엔드 디바이스 및/또는 다른 디바이스에 의해 출력된 오디오에 대응하는 만

족하는(pleasing) 햅틱 경험을 경험하는 엔드 디바이스에 의해 처리되고 출력된다. 엔드 디바이스는 오디오 플레이어들, 모바일 폰들, 스마트 폰들, 비디오 플레이어들, PDA들, 컴퓨터들, 텔레비전들, 비디오 게임 콘솔들, 컴퓨터 주변기기들(예를 들어, 마우스, 조이스틱, 볼러 볼, 키보드), 스티어링 휠들(steering wheels), 가구, 슬라이더들, 손잡이들, 헤드폰들, 터치스크린들, 이동식 또는 비-이동식 라디오들, 단방향/양방향 페이지들 및 다른 적합한 사용자 조작가능한 전자 디바이스들을 포함하지만, 이에 한정되지 않는다.

<19> 시스템 및 방법은, 엔드 디바이스에 의해 출력되는 것이 본원에 기술되지만, 복수의 엔드 디바이스들은 실시예에 따라, 동일 햅틱 이벤트들 및/또는 동일 햅틱 이벤트들의 상이한 부분들을 동시에 또는 다른 시간에 출력할 수 있다는 것도 주목해야 한다. 엔드 디바이스는 그 내에 하나 이상의 액추에이터들을 포함할 수 있다는 것을 주목해야 한다. 엔드 디바이스 내의 다수의 액추에이터들의 실시예에서, 다수의 액추에이터들에 의해 출력될 햅틱 이벤트들은 각각의 햅틱 이벤트에 대해 동시에, 선택적으로, 교대로 및/또는 개별적으로 행해질 수 있다.

<20> 도 4는 실시예에 따른 시스템을 구현한 전체 아키텍처를 도시한다. 도 4에 도시된 바와 같이, 다양한 소스들은 입력 신호(99)를 시스템(300)에 제공할 수 있다. 예를 들어, 디지털 샘플링된 오디오 신호 또는 구조화된 표현 신호는 서버(402), 오디오 신호원(CD, DVD, DAT 등), 호스트 컴퓨터(406), 모바일 디바이스(408) 및/또는 커머셜 키오스크(commercial kiosk)(410)로부터 수신될 수 있다. 입력 신호는 유선 또는 무선 개인 또는 공용 네트워크(98)를 통해 시스템(300)의 수신 모듈(412)에 송신된다. 그 후, 신호는 보다 상세히 아래 설명되는 바와 같이 프로세서(414)에 의해 처리된다. 일단 처리되면, 시스템(300)은, 하드-와이어 또는 무선으로 포트(416)를 통해 하나 이상의 엔드 디바이스들(420)에 처리된 정보를 출력한다. 그 후, 엔드 디바이스(420)는, 출력 파일을 수신하면, 옵션의 오디오 스피커(424)와 함께 하나 이상의 액추에이터들(422)을 사용하여 햅틱 이벤트들을 출력할 수 있다. 도 4에 기술된 아키텍처는 실시예이고, 여기서, 추가적 및/또는 대안적인 컴포넌트들이 본원에 기술된 주제의 범위에서 벗어남 없이 활용될 수 있다는 것을 주목해야 한다.

<21> 이제 도면들을 참조하면, 도 1은 음표(notes), 화음(chords) 및 쉼표(rests)의 시퀀스를 포함하는 전통 낱장 악보 스타일(sheet music style)을 이용하여 시각적으로 인지가능하게 나타낸 악곡(musical piece)의 일부(100)를 도시한다. 악곡은 예를 들어, 디지털 샘플 형태 또는 구조화된 표현 형태로 플레이되고 레코딩될 수 있다. 각각의 음표는 기계적 진동의 주파수를 나타낸다. 음표의 유형들(즉, A, A#, B, C, C#, D, D#, E, F, F#, G, G#)은 피치(pitch)의 관점에서 서로 상이하다. 온음표들, 반음표들, 1/4 음표들, 1/8 음표들, 1/16 음표들 등은 각각의 음표 또는 주파수에 대한 플레이의 지속시간을 나타낸다. 마찬가지로, 온쉼표들, 반쉼표들, 1/4 쉼표들, 1/8 쉼표들, 1/16 쉼표들 등은 어떠한 음표도 플레이되지 않는 지속시간을 나타낸다. 음표들 및 그들의 지속기간의 시퀀스는 어떤 주파수들이 어떤 시간에 플레이될 것인지에 대한 표현을 결정한다.

<22> 본 발명의 실시예는 디지털 샘플링된 오디오 파일에 햅틱 이벤트들을 자동으로 맵핑하는 것에 관한 것이다. WAV, MP3, MP4, AAC, ASX, WMA 포맷들을 갖지만 이에 한정되지 않는, 샘플링된 오디오 소스들(및 파일들)은 모두 기본적으로 샘플 레이트 및 시간 순서의 샘플들의 세트(종종 좌우 채널들)와 같은 동일 유형의 정보를 제공한다. MP3 및 AAC 포맷과 같은 소정의 압축 파일들은 서브-대역 주파수들 뿐만 아니라 그들의 서브-대역 주파수들의 오디오 이벤트들의 크기들 및 계수들과 관련된 정보를 포함한다. 그러나, 구조화된 표현 파일들과 다르게, 악기, 피치, 템포 또는 다른 음악적 내용에 관한 어떠한 명백한 정보도 비-구조화된 표현 파일들에서 이용가능하지 않고, 단지 전체적으로 렌더링된 사운드들만이 이용가능하다. 예로서 샘플링된 오디오 신호의 시간-진폭 그래프가 도 2에 도시된다.

<23> 부가적으로, 또는 택일적으로, 시스템은 구조화된 표현 신호(예를 들어, MIDI)로부터 햅틱 이벤트들을 자동으로 맵핑하는 것이 가능하다. 특히, 오디오 파일의 특정 구조화된 표현이 주어지면, 시스템은 구조화된 표현 신호에 의해 고유하게 제공된 데이터를 분석함으로써 햅틱 이벤트들을 자동으로 맵핑할 수 있다. 그러나, 어느 한 유형의 파일을 자동으로 맵핑하여, 입력 신호는 신중히 처리되고, 엔드 디바이스에서의 액추에이터들의 성능 제한들로 인해 분석이 단순화되어, 파일의 오디오 신호들로부터 만족하는 개별 햅틱 시퀀스들을 제공한다.

<24> 현재의 햅틱 기술은 약 4개의 동시적 햅틱 이벤트까지 플레이하고 약 1ms 제어 루프로 동작할 수 있다. 이는 존재하는 액추에이터들의 제한들로 인한 것이다. 존재하는 ERM(Eccentric Rotating Mass) 모터 기술의 제한들이 주어지면, 동시적인 햅틱 효과들을 플레이하는 것은 예기치않은 액추에이터 거동(behavior)을 초래할 수 있다. 또한, ERM 모터들은 초당 약 4개의 햅틱 이벤트들의 최대치까지 햅틱 이벤트 주파수를 제한하는 비교적 낮은 기계적 대역폭들을 갖는다. ERM 모터들은 또한 피크 가속(peak acceleration)이 달성되기 전에 약 50-80 ms의 시동 지연을 갖고, 이 시동 지연은 그러한 ERM 모터들을 구동하기 위해 햅틱 제어 신호들을 형성하는 것이 고려되어야 한다. ERM 모터들이 본원에 기술된 주제의 맥락에서 기술되지만, 엔드 디바이스, 햅틱 효과

해상도, 변환될 오디오의 디지털 품질, 및 사용에 있어서의 애플리케이션의 유형에 따라 다른 유형의 모터들 또는 액추에이터들이 사용될 수 있다는 것이 고려된다. 단지 예를 들어, 액추에이터는 ERM 모터들에 대해 우세한 해상도 성능들을 갖는 선형 공명 액추에이터(linear resonant actuator, LRA)일 수 있다. 선형 전류 제어 모터들, 스테퍼(stepper) 모터들, 자기 액추에이터들, 공기/수압(pneumatic/hydraulic) 활성 액추에이터들, 토크들(torquers)(제한된 각도 범위를 갖는 모터), 및 보이스 코일 액추에이터들을 포함하는 다른 유형의 액추에이터들이 고려되지만 이에 한정되지 않는다.

- <25> 햅틱적 인에이블형 엔드 디바이스(haptically enabled end-device)와 접촉하여 만족하는 햅틱 경험을 사용자에게 제공하기 위해, 시스템은, 시간 기간 내에 너무 많은 햅틱 이벤트들을 생성해서는 안된다. 부가적으로, 햅틱 이벤트들은 하나의 일정한 진동 또는 '블러(blur)'를 생성하지 않도록 충분히 떨어진 공간에 있어야 하지만, 그럼에도 불구하고 음향 효과들, 음악적 템포 및/또는 멜로디들과 일치해야 한다. 키(key)의 변화, 멜로디 전환(transition) 등과 같은 예외적인 이벤트들은 종종, 햅틱 이벤트에 의해 강조되어야 한다. 음악적 리듬에는 종종, 오디오 프로그램 신호의 템포 및 오디오 프로그램 신호의 시작에 대한 그 템포의 위상 오프셋(phase offset)을 식별함으로써 악센트를 주어야 한다. 사전에 상당량의 침묵(silence)을 갖는 오디오 이벤트들은 음악에서의 그들의 악센트(accent)로 인한 유사한 햅틱 이벤트들(예를 들어, 음향 효과로서의 자동차 충돌, 심벌즈)일 수도 있다.
- <26> 도 3은 실시예에 따른 수신된 디지털화된 샘플링된 오디오 신호의 햅틱 효과 맵핑을 위한 시스템 및 방법을 도시하는 프로세스 블록도이다. 일반적으로, 실시예에 따르면, 수신된 샘플링된 오디오 입력은 시스템에 의해 수신되고, 자동으로 "변환" 또는 "맵핑"되어, 신호의 식별된 오디오 이벤트들의 재생과 동시에 엔드 디바이스에서의 햅틱 효과들의 재생을 제어하는 햅틱 제어 신호 또는 햅틱 이벤트를 형성한다. 도 3의 블록들은, 하드웨어 회로들, 집적 회로들 및 ASIC들을 포함하지만 이에 한정되지 않는 아날로그 또는 디지털 하드웨어일 수 있는 일반적인 컴포넌트들로서 지정된다. 부가적으로 또는 택일적으로, 도 3의 블록들은, 프로세서들 및 메모리들에 의해 활용되거나 또는 그에 구현되는 루틴들, 서브루틴들, 코드, 스레드들 등과 같은 소프트웨어 코드 컴포넌트들이다.
- <27> 도 3의 실시예에 따르면, 수신된 샘플링된 오디오 신호는 블록(310)에서 계산적 디바이스의 메모리에 로드된다. 계산적 디바이스는 종래의 개인용 컴퓨터, 처리 모듈, 서버, 클라이언트, 상기 자체 정의된 엔드 디바이스, 또는 충분한 처리 능력들을 갖는 임의의 다른 전자 디바이스일 수 있다. 샘플링된 오디오 신호는 CD, 미니-디스크, DVD, DAT, 하드-드라이브 또는 다른 매체로부터 취해질 수 있다. 언급한 바와 같이, 오디오 신호는 압축된 디지털 오디오 포맷(예를 들어, MP3, AAC 등)으로 되어 있을 수 있다. 일 실시예에서, 오디오 신호는 이 후 엔드 디바이스에 의해 변환되어 플레이할 수 있는 상용 또는 비-상용 웹사이트로부터 직접(유선 또는 무선으로) 다운로드될 수 있다. 다른 실시예에서, 오디오 트랙들은 원격으로 변환된 후 서버에 저장되고, 이에 의해, 변환된 햅틱 트랙들은, 서버로부터 플레이될 하나 이상의 엔드 디바이스들로 직접 다운로드될 수 있다. 일 실시예에서, 수신된 파일은 시스템 오프-라인에 의해 처리되고, 이에 의해, 시스템은 일부 또는 전체 파일이 수신된 후 그것을 처리한다. 일 실시예에서, 수신된 파일은 실시간 방식으로 시스템에 의해 처리되고, 이에 의해, 시스템은 아래에서 더 상세히 논의될 바와 같이, 파일이 수신됨에 따라 데이터를 처리하여 출력한다.
- <28> 수신된 파일은, Microsoft Windows Media Player®, RealPlayer®, iTunes®, MusicMatch Jukebox®, 또는 서버 및/또는 엔드 디바이스 상의 다른 사유 프로그램(proprietary program)과 같은 상용 컴퓨터 소프트웨어 프로그램에 포함된 파일 리더/플레이어로 입력될 수 있다. 대안적으로, PCM형 출력을 발생시키는 임의의 코덱(encoder-decoder)이 시스템과 함께 사용될 수 있다.
- <29> 일 실시예에서, 시스템은 단일(단청의(monaural)) 입력 신호로서 샘플링된 오디오를 수신한다. 일 실시예에서, 시스템은, 예를 들면 프리믹싱(pre-mixing) 또는 서브셀렉팅(sub-selecting)에 의해 변환되거나 맵핑될 수 있는 스테레오 또는 멀티 채널 입력들로서 샘플링된 오디오를 수신한다. 또한, 복수의 햅틱 트랙들은 소망된다면 동시에 또는 다른 시각에 처리될 수 있다.
- <30> 논의되는 바와 같이, 수신된 신호는 시스템에 의해 계산적으로(computationally) 분석될 수 있고, 그에 의해 시스템은 서브-대역들에서 오디오 이벤트들을 분석하고 식별할 때 주파수 서브-대역 계수들을 효과적으로 계산한다. 그 후 일 실시예에서 입력 신호가 시스템의 선택 메모리(optional memory)로 로드된다. 시스템은 메모리로부터 신호를 검색하고 주파수 및 그 시간당 그의 에너지 용량(energy content)에 관해서 입력 신호를 분석한다(블록 312). 일 실시예에서, 시스템은 윈도우드 FFT(windowed Fast Fourier Transform) 알고리즘을 사용하여 오디오 신호의 에너지 분포 데이터를 생성한다(블록 312). 일 실시예에서, 웨이블릿 변환(wavelet

transform)을 포함하지만 그에 제한되지는 않는 임의의 다른 적당한 절차 또는 알고리즘을 이용하여 신호의 에너지 분포 정보를 생성한다. 스펙트럼 분해(spectral decomposition) 분석은, 신호를 처리하는 데 있어서 시스템에 의해 이용될 수 있는 복소 계수(complex coefficient)로서 인코딩된 위상 정보를 생성할 수도 있다.

- <31> 일 실시예에서, 시스템은 주파수 서브-대역들의 분할(segmentation) 전에, 스펙트럼 분해 계산에 대한 보상 요인(compensation factor)을 이용하여 인간 귀 전달 함수 및 지각 요인(Human Ear transfer function and perceptual factor)들을 설명한다. 예를 들면, 인간의 귀는 60Hz의 사운드들과는 반대로 3-4kHz의 사운드들에 더욱 민감하다. 예에서, 시스템은 60Hz 주위의 주파수 용량(frequency content)을 대략 40dB만큼 감쇠시키지만, 3kHz와 4kHz 사이의 주파수들의 용량은 감쇠시키지 않는다. 그리하여, 60Hz 비트(beat)에 상당히 더 많은 에너지가 있을 수 있지만, 사용자는 감쇠로 인해 더 현저하게 높은 주파수 대역의 비트를 인식할 가능성이 있다.
- <32> 유사하게, 목표 햅틱 디바이스에 대한 스피커들의 오디오 전달 함수(audio transfer function)가 고려되고 분할 전에 스펙트럼 분해에 적용될 수 있다. 예를 들면, 대부분의 셀 폰 스피커들에 있어서 주파수 응답은 200Hz 미만인 주파수들은 전송되지 않는 것이다. 그리하여, 그러한 스피커들에 의해 렌더링되는 오디오 신호를 수신할 때, 200Hz 아래의 비트를 강조하는 것은 예기치 않은 햅틱 효과들을 야기할 수 있다. 처리 시에 스펙트럼 분해의 이러한 주파수 대역들을 무시하는 것은 더욱 일관된 결과들을 줄 수 있다.
- <33> 일 실시예에서, 시스템은 스펙트럼 분해 분석으로부터의 복소 계수들의 절대값을 이용해서 주파수에 대하여 오디오 신호의 에너지 분포를 맵핑하여 오디오 신호의 크로스오버 주파수들 및 서브-대역 신호들의 지정(designation)을 허용한다. 이것은 도 3의 블록(314)에 표시된다.
- <34> 도 3에 도시된 바와 같이, 블록(314)에서 시스템 및 방법은 입력 오디오 신호의 에너지 분포로부터 다수의 서브-대역 주파수들을 결정한다. 전송된 바와 같이, 시스템은 소프트웨어, 아날로그 또는 디지털 하드웨어 및/또는 펌웨어를 이용하여 이러한 분석을 수행할 수 있다. 일 실시예에서, 시스템은 신호의 에너지 분포를 분석하고 주파수들에 기초하여 3개의 서브-대역 카테고리들 에너지를 지정한다. 일 실시예에서, 서브-대역 카테고리들은 고 주파수 서브-대역들, 중간 주파수 서브-대역들, 및 저 주파수 서브-대역들이다. 도 5는 입력 오디오 신호의 고 주파수 서브-대역 신호(500) 및 중간 주파수 서브-대역 신호(502)를 나타내는 예시적인 필터링된 시간-주파수 그래프를 도시한다. 일 실시예에서, 시스템 및 방법은 입력 오디오 신호의 에너지 분포를 3개보다 많은 또는 3개보다 적은 서브-대역 카테고리들로 지정한다.
- <35> 일 실시예에서, 저 주파수 서브-대역은 저 주파수 값들의 범위 내에 있도록 시스템에 의해 지정되는 반면, 고 주파수 서브-대역은 고 주파수 값들의 범위 내에 있도록 지정된다. 시스템은 중간 주파수 서브-대역이 저 서브-대역의 최고 주파수와 고 서브-대역의 최저 주파수 사이가 되도록 지정한다. 각각의 주파수 서브-대역 범위들은 논의된 것들에 제한되지 않고, 각 서브-대역은 설계자의 소망에 기초하여 임의의 주파수 상한 및 주파수 하한을 갖도록 지정될 수 있다는 것을 주의해야 한다. 일 실시예에서, 저, 중간 및 고 주파수 범위들은 전체 신호 또는 전체 신호의 분할된 부분들의 전력 스펙트럼의 전역적 평균(global average)에 기초하여 시스템에 의해 결정된다. 저, 중간 및 고 주파수 범위들은 임의의 다른 적당한 방법들을 이용하여 시스템에 의해 결정될 수 있다는 것이 고려된다.
- <36> 특히, 시스템 및 방법은 입력 오디오 신호를 서브-대역 크로스오버 주파수들에 관하여 분석한다. 일 실시예에서, 시스템은 저 주파수 서브-대역과 중간 주파수 서브-대역 사이의 하나 이상의 서브-대역 크로스오버 주파수들 f_m 을 계산한다. 또한, 시스템 및 방법은 중간 주파수 서브-대역과 고 주파수 서브-대역 사이의 하나 이상의 서브-대역 크로스오버 주파수들 f_h 를 계산한다. 예에서, f_m 은 대략 300Hz 내지 대략 1000Hz의 범위에 있도록 지정된다. 예에서, f_h 는 대략 3500Hz 내지 대략 5000Hz의 범위에 있도록 지정된다. 이러한 범위들은 단지 예들이고 다른 주파수 범위들이 시스템 및 방법의 범위 내에서 생각된다는 것을 주의해야 한다.
- <37> 일 실시예에서, f_m 및 f_h 주파수들은 스펙트럼 롤오프(spectral rolloff)가, 오디오 입력 신호의 전체 에너지의 특정 백분율이 존재하는 주파수 미만인 스펙트럼 롤오프의 15 및 85 백분위수(percentile)의 평균값들로서 연산된다. 이들 특정 백분율들은 블록(312)에서 계산된 신호의 에너지 분포로부터 연산될 수 있다. 다른 스펙트럼 롤오프 백분위수들이 생각되고 반드시 15 및 85 백분위수로 제한되지는 않는다는 것을 주의해야 한다. 일 실시예에서, 시스템 및 방법은 각 시간 윈도우(time window)의 지정된 주파수들에 대한 에너지를 정규화하고 특정 백분위수에 대응하는 누적값(cumulative value)을 계산한다. 이것은 크로스오버 주파수들을 계산하는 하나의 방법이고 크로스오버 주파수들을 계산하는 다른 방법들이 생각된다는 것을 주의해야 한다. 예를 들면, 일 실시

예에서, 서브-대역 신호 크로스오버 주파수들은 전체 신호의 FFT를 계산함으로써 그리고 톨오프 값들을 계산함으로써 연산될 수 있다. 그 후 각 서브-대역의 에너지 값들을 합산함으로써 입력 오디오 신호의 에너지 분포(예를 들면, 스펙트럼 사진(spectrogram))로부터 3개의 서브-대역 신호들이 구축될 수 있다. 44KHz 시간 정보(원래의 오디오 샘플 레이트)가 스펙트럼 분해 연산 동안 128Hz 시간 데이터로 변환될 수 있기 때문에, 이러한 절차는 입력 신호를 다운(down) 샘플링하는 데 사용될 수도 있다.

- <38> 일 실시예에서, 고, 중간, 및 저 주파수 서브-대역 신호들은 각각 시스템에 의해 분할되고 처리된다. 도 3에 도시된 바와 같이, 중간 주파수 서브-대역 신호는 블록(316)에 의해 지정된, 중간 대역 옵티마이저로 보내진다. 중간 대역 옵티마이저(316)는 소망하는 수의 EPS(effects per second)에 대응하는 개별(discrete) 이벤트들의 트랙을 발생시킨다. 액추에이터의 해상도, 음악의 종류, 엔드 디바이스의 종류 및 시스템의 처리 능력들 또는 사용자의 소망을 포함하지만 그것으로 제한되지 않는 소정의 요인들에 기초해서 EPS가 규정될 수 있지만, 일 실시예에서는, 목표로 정해진 EPS는 사용자에게 의해 규정된다.
- <39> 시스템은 햅틱 이벤트들 사이의 최소 분리를 실시하여, 엔드 디바이스의 액추에이터가 적당한 시각에 그것이 출력하는 각 햅틱 효과를 개시하고 단계적으로 축소하는 것을 허용한다. 이러한 최소 분리는 엔드 디바이스의 액추에이터가 개별적인 다수의 햅틱 이벤트를 생성할 수 있게 하고 두 개 이상의 햅틱 이벤트가 겹치는 것, 또는 서로 겹쳐서 다수의 햅틱 효과들의 "흐릿한(blurred)" 느낌의 부분들을 생성하는 것을 방지한다. 일 실시예에서, 햅틱 이벤트들 사이에서 제한된 시간의 양은 50ms이고, 특정 지속기간은 서브-대역에 따라 100-250ms 사이이다. 시스템은 이러한 값들에 제한되지 않고 다른 값들이 생각된다는 것을 주의해야 한다.
- <40> 중간 대역 옵티마이저(316)는 오디오 입력 신호의 보컬 멜로디에 대응하도록 개별 햅틱 이벤트들의 트랙을 발생시킴으로써 보컬 이벤트들의 멜로디를 캡처하려고 한다. 그리하여 시스템은 클리핑 파라미터(clipping parameter)를 이용하여 소망하는 EPS의 목표 범위 내의 가장 높은 에너지량을 갖는 이벤트들을 식별한다.
- <41> 도 3에 도시된 바와 같이, 클리핑된 입력 신호가 중간 대역 옵티마이저(316)로부터 출력되면, 신호는 로컬 세그멘터(318)로 입력된다. 일반적으로, 로컬 세그멘터(318)는 클리핑된 입력 신호의 장기 평균(long time-average)과 클리핑된 입력 신호의 단기 값(short-time value)을 비교하여 중간 주파수 서브-대역 신호를 각 이벤트에 대한 일련의 시작 및 종료 시각들로 분할하는 알고리즘을 이용한다. 도 3에 화살표에 의해 도시된 바와 같이, 그 후 로컬 세그멘터(318)는 이러한 정보를 옵티마이저(316)에 다시 제공하고, 이로써 옵티마이저(316)는 개별적인 이벤트들의 분할이 소망하는 EPS의 목표 범위에 있을 때까지 하나 이상의 클리핑 파라미터를 단조롭게(monotonically) 조정하도록 1차원 라인 탐색을 적용한다. 일 실시예에서, 이러한 처리는 신호에 대하여 소망하는 수의 EPS에 도달될 때까지 반복된다. 일 실시예에서, 분할 처리는 미리 설정된 수의 되풀이(iteration) 동안 반복된다. 분할 처리의 상세들은 이제 논의될 것이다.
- <42> 도 6은 실시예에 따라 중간 주파수 서브-대역 신호를 개별적인 햅틱 이벤트들로 분할하는 데 사용되는 평균 전력 및 단기 전력(short-time power)의 표현을 나타내는 시간-신호 그래프를 도시한다. 도 6의 그래프는 20초 윈도우 내의 단기 신호 값들(602)뿐만 아니라 중간 주파수 서브-대역 신호의 시간-평균(600)을 도시한다. 다른 지속 기간 윈도우(time duration window)들이 시스템에 의하여 이용될 수 있다는 것을 주의해야 한다.
- <43> 일 실시예에서, 단기 신호값은 입력 오디오 신호를 5Hz 버터워스 필터(Butterworth filter)를 통하여 저역 필터링함으로써 획득될 수 있다. 입력 신호가 아날로그 또는 디지털 영역(domain)의 임의의 다른 디바이스 또는 방법을 이용하여 필터링된다는 것을 주의해야 한다. 일 실시예에서, 시스템은 단기 평균값 신호가 그 시점에 대한 특정 임계값을 초과하는 소정 시각에 새로운 이벤트를 만든다. 도 6의 그래프에 관해서, 이벤트는 단기 평균값들이 장기 평균을, 소망하는 방향으로(예를 들면, 윗방향으로 또는 아랫방향으로) 가로지르는 점들에서 시작하고 끝난다. 예를 들면, 도 6에서, 로컬 세그멘터(318)는 단기 평균(602)이 장기 평균선(600)을 윗방향으로 가로지르는 곳을 식별함으로써 이벤트가 시작하는 곳(점(604))을 결정한다. 로컬 세그멘터(318)는 단기 평균(602)이 장기 평균선(600)을 아랫방향으로 가로지르는 곳(점(606))을 식별함으로써 이벤트가 끝나는 곳을 결정할 수도 있다. 상기는 분할 처리의 예일 뿐이며 다른 분할 처리들 및 알고리즘이 시스템에 의해 사용될 수 있음이 명백하다는 것을 주의해야 한다.
- <44> 저 및 고 주파수들의 경우, 햅틱 이벤트들로 간주되는 이벤트들은, 엔드 디바이스의 액추에이터가 초당 효과적으로 생성할 수 있는 것보다 많은 것이 보통이다. 중간 주파수 서브-대역들에 대해 전송된 최적화 처리는, 보통 고 주파수 및 저 주파수 서브-대역에 고유한 다수의 가깝게 간격을 둔(closely spaced) 주파수 피크들로 인해 고 주파수 및 저 주파수 서브-대역에 대하여는 선호되지 않는다(도 5). 이러한 다수의 가깝게 간격을 둔 주파수 피크들은 액추에이터가 입력 오디오 신호의 템포와는 조화되지 않는 앨리어싱된 결과(alias result)들을

생성하게 할 수 있다.

- <45> 이러한 문제를 다루기 위해서, 시스템은 고 주파수 및 저 주파수 서브-대역의 신호들에 대해 중간 주파수 서브-대역들의 신호들과 상이한 최적화 및 분할 접근법을 이용한다. 도 3은 고 주파수 및 저 주파수 서브-대역들이 시스템에 의하여 다르게 처리된다는 것을 도시한다. 고 주파수 서브-대역과 관련하여, 앨리어싱(aliasing)은 시스템이 신호를 처리할 때 고려하는 관심사이다. 따라서, 이러한 앨리어싱을 고려하지 않고 생성된 햅틱 이벤트들은 음악과 일치하지 않는 비트 또는 리듬을 따를 수 있다.
- <46> 일반적으로, 도 3에 도시된 바와 같이, 시스템은 고 주파수 신호들에 대한 하나 이상의 클립핑(clipping) 파라미터들을 조절하기 위해 1차원(1-D) 라인 탐색을 수행하는 고 주파수 옵티마이저(320) 블록을 포함한다. 옵티마이저 블록(320)은 출력 신호에서의 앨리어싱의 양에 대한 측정을 이용하여, 클립핑 값을 조절하며, 그 클립핑 값은 그 다음에 세그먼트 블록(322)에 제공된다. 고 서브-대역 세그먼트(322)는 각 햅틱 이벤트의 시작 및/또는 종료 시간을 결정하며, 화살표에 의해 도시된 바와 같이, 그 결과를 옵티마이저 블록(320)으로 다시 제공한다. 그 후, 옵티마이저(320)는, 타겟 앨리어스 레벨에 도달하거나 또는 소정의 갯수의 사전정의된 스텝들이 발생될 때까지, 그에 응답하여 클립핑 값을 조절한다.
- <47> 앨리어싱은 생성된 햅틱 이벤트들의 시작 시간들 사이의 차이들의 분포를 검사함으로써 측정된다. 이러한 분포가 액추에이터에 의해 지시된 최소 효과 분리와 동일한 델타(delta)에서 매우 큰 피크를 갖는 경우, 분할 결과는 높은 앨리어싱을 갖는 것으로 고려된다. 합리적인 타겟 앨리어스 값은 30%일 수 있으며, 그것은 효과들의 30%가 정확하게 최소 분리 값만큼 상이한 시작 시간들을 가짐을 의미한다.
- <48> 세그먼트 블록(322)에서 이용된 알고리즘은 일 실시예에서의 중간 서브-대역(320)에 의해 이용된 것과 동일할 수 있다. 일 실시예에서, 블록(322)에 의해 고 주파수 서브-대역 신호들을 처리하는데 이용된 알고리즘은 중간 주파수 서브-대역 신호들에 대해 블록(318)에 의해 수행된 알고리즘과 유사하지만, 몇 가지의 차이점이 있다. 일 실시예에서, 고 주파수 신호들에 대해 장기 평균(long time average)은 50ms이다.
- <49> 일 실시예에서, 입력 신호는 신호대 잡음비(S/N)를 증가시키도록 전처리된다. S/N은 신호의 최대 값에 대한 클립핑된 신호의 평균의 비율이도록 계산될 수 있으므로, 처리된 신호는 입력 신호로부터 스케일링된(scaled) 1.5 초 이동 평균(moving average)을 감산하고, 음의 값들을 0으로 설정함으로써 계산된다. 이것은 단지 일례일 뿐이며, 시스템은 신호에서의 신호대 잡음비를 향상시키기 위해 다른 방법을 이용할 수 있음을 주지해야 한다.
- <50> 일 실시예에서, 250ms보다 긴 것으로 측정되는 햅틱 효과들은, 고 주파수 신호들에 대해 250ms의 최대치로 클립핑된다. 일 실시예에서, 클립핑된 신호가 가장 최근의 50ms 윈도우에서 최고 피크보다 큰 경우에만, 새로운 세그먼트들(segments)이 생성된다. 전술한 내용은 예시적인 것이며, 전술한 요인(factor)들로 한정되지 않음을 주지해야 한다. 또한, 전술한 요인들 중 하나 이상은 알고리즘을 수행시에 이용될 수 있음을 주지해야 한다. 전형적으로, 고 주파수 서브-대역 분할의 결과는, 입력 오디오 신호의 템포(tempo)에 대략적으로 대응하는 짧은 햅틱 이벤트들의 세트이다.
- <51> 도 3을 다시 참조하면, 일 실시예에서, 저 주파수 서브-대역 신호들은 중간 주파수 서브-대역 신호들과는 상이하게 처리된다. 전형적으로, 입력 오디오 신호의 저 주파수 부분은 음악의 리듬에 대응하며, 또한, 다양한 유형의 타악기 및/또는 매우 낮은 주파수의 보컬들(예를 들면, 베이스 보컬들)에 의해 지배된다. 고 주파수의 경우와 같이, 시스템 및 방법은 저 주파수 서브-대역 신호들을 처리하여, 리듬과 일치하는 햅틱 이벤트들을 생성하며, 과도한 앨리어싱 아티팩트(artifact)를 생성한다. 음악에서의 드럼 비트(drum beat)들은 저 주파수 서브-대역 스펙트럼 내의 어떤 주파수 범위 내에 있는 것이 일반적이다. 따라서, 일 실시예에서, 시스템은 비트들을 식별하기 위해, 저 주파수 서브-대역에서의 좁은 범위의 주파수들을 고려한다.
- <52> 일 실시예에서, 시스템은 블록(312)에서의 윈도우와 비교하여 훨씬 긴 시간 프레임 윈도우를 이용하여 원래의 입력 신호의 저 주파수 서브-대역 신호의 스펙트럼 분해 분석(블록 324)을 처음에 수행함으로써 이러한 특정 범위의 주파수들을 식별한다. 일 실시예에서, 시간 프레임 윈도우는 1/32초 또는 31ms이지만, 다른 지속기간도 고려되며, 31ms로 한정되지는 않는다. 시간 프레임 윈도우는 신호의 스펙트럼 롤오프(spectral rolloff)를 평가하기 위해, 저 주파수 범위에서, 블록(312)에서 계산된 스펙트럼 분해보다 상당히 높은 해상도를 제공한다. 시스템 및 방법이 보다 높은 해상도 에너지 분포를 계산(블록 324)하면, 시스템은 신호의 전력 스펙트럼을 계산한다. 이것은 블록(326)에 의해 수행되며, 여기서, 시스템은 지정된 시간 윈도우에 걸쳐 스펙트럼 분해로부터 에너지 분포를 합산한다. 입력 신호는 2개의 상이한 스펙트럼 분해 분석(도 3에서의 블록 312 및 324)을 거치는 것으로 도시되지만, 저 주파수 서브-대역 신호의 스펙트럼 분해는 블록(312)에서 대안적으로 수행될 수 있다.

므로, 그 결과는 저 주파수 서브-대역 신호의 전력 스펙트럼을 계산시에 블록(326)에 의해 이용된다.

- <53> 시스템 및 방법은 저 주파수 서브-대역 신호에 대해 전력 스펙트럼 계산(블록 326)을 이용하여, 전력 레벨에 기초하여 주파수들을 정렬한다(블록 328). 몇 개의 주파수들이 전력 스펙트럼에서의 동일 피크에 대응할 수 있는 것이 일반적이므로, 주파수 피크들의 보다 효율적인 정렬을 허용하도록, 중복 주파수들이 제거될 것이다.
- <54> 도 7은 소정의 실시예에 따라 저 주파수 서브-대역 신호를 분할하는데 이용된 주파수-전력 그래프(즉, 전력 스펙트럼)를 도시한다. 전력 스펙트럼은 설정된 수의 주파수 빈(bin)들로 입력 신호에서 제공된 에너지의 양을 나타낸다. 이러한 정보는 주파수를 전체 전력의 빈(bin)들로 정렬하기 위해 시스템에 의해 이용된다. 예를 들어, 2.1과 2.8(x 10⁴) 사이의 전체 전력의 빈은, 내림차순의 전체 전력으로 정렬된 640Hz, 672Hz, 416Hz, 320Hz 및 608Hz의 주파수 피크들을 갖는다.
- <55> 동일한 비트와 같은 에너지 또는 전력의 정확히 동일한 시간적 분포를 수반할 수 있는 다른 피크들을 제거할 뿐만 아니라, 이들 중복 주파수들을 제거하기 위해, 각각의 주파수 빈과 관련된 시간적 신호들 사이의 교차 상관 매트릭스(cross-correlation matrix)가 시스템에 의해 계산되며, 이에 대해서는 보다 상세히 후술된다. 일 실시예에서, 상관 매트릭스는 주파수 빈 fi 및 fj에서의 시간 신호가 포지티브 상관되는(positively correlated) 경우, 위치(i,j)에서 대략적인 높은 값을 가질 것이다. 소정의 실시예에 따르면, 0.5보다 높은 상관 계수를 갖는 모든 주파수 빈들이 보다 높은 관련 전력을 갖는 빈으로 클립핑된다. 즉, 시스템은 최고 에너지 신호를 유지할 것이며, 클립핑된 상관 계수보다 높은 저 에너지 신호들을 이용하지 않는다. 상관 계수는 0.5로 제한되지 않으며, 0.5 이외의 값들이 고려됨을 주지해야 한다. 나머지 주파수들의 경우, 블록(324)에서 이용된 알고리즘은, 비록 다른 타겟 백분율이 고려되기도 하지만, 15%의 최대 엘리어스 타겟을 갖는 옵티마이저 및 세그멘터(블록 330, 332)를 이용한다.
- <56> 일 실시예에서, 시스템은 저 주파수 대역에서의 모든 연속하는 주파수 빈들 사이의 상관 계수들을 계산한다. 주기적 스내어 드럼 비트(periodic snare drum beat)들, 핸드 클랩(hand clap)들 등에 대응하는 오디오 세그먼트들은 통상적으로 높은 대역폭 이벤트들이며, 최대 상관 계수들을 탐색함으로써 가장 쉽게 식별된다. 다음 주파수에 대해 최고 상관 계수를 갖는 주파수 대역이, 저역 통과 검출 알고리즘에서의 또다른 처리를 위해 시스템에 의해 선택된다. 2개의 랜덤 변수들 x, y의 상관 계수들은 다음과 같이 주어진다.

수학식 1

$$cor(x, y) = E(x, y) - \frac{E(x)E(y)}{\sqrt{\sigma_x^2 \sigma_y^2}}$$

- <57>
- <58> 진술한 바와 같이, 고 주파수 신호, 중간 주파수 신호 및 저 주파수 신호가 클립핑되고 분할된 후에, 시스템 및 방법은 서브-대역 신호들 각각에 대한 후보 햅틱 이벤트들의 리스트를 생성할 것이다. 특히, 도 3에 도시된 바와 같이, 분할된 고 주파수 서브-대역 신호들은 고 주파수 세그멘터 블록(322)으로부터 우선순위 클립 블록(334)으로 전달된다. 또한, 분할된 중간 주파수 서브-대역 신호들은 블록(318)으로부터 우선순위 클립 합산 블록(334)으로 전달된다. 더욱이, 분할된 저 주파수 서브-대역 신호들은 블록(332)으로부터 우선순위 클립 블록(334)으로 전달된다.
- <59> 시스템 및 방법은 햅틱 이벤트들을 우선순위 클립 블록(334)을 통과시키지 않고 직접 간단히 출력할 수 있지만, 이렇게 하는 것은 아마도 엔드 디바이스의 액추에이터에 의해 혼란되거나 또는 흐릿한 햅틱 느낌을 나타낼 수 있다. 이것은 시간적으로 중복되는 몇몇 햅틱 이벤트들에 기인한 것일 수 있다. 이것을 극복하기 위해, 시스템은 우선순위 클립 블록(334)을 사용하여 각각의 서브-대역 신호들로부터 후보 햅틱 이벤트들을 우선순위화 한다.
- <60> 특히, 우선순위 클립 블록(334)은, 주파수 서브-대역들이 서로의 특정한 시간 주기내에 있으면, 동일하거나 또는 상이한 주파수 서브-대역들로부터, 2 이상의 햅틱 이벤트가 액추에이터에 의해 출력되는 것을 방지할 수 있다. 또한, 이것은 중복하는 날카로운(sharp) 이벤트 및 강한(strong) 이벤트가 서로를 캔슬하는 것을 방지한다. 예를 들어, 날카로운 이벤트 및 강한 이벤트 모두가 액추에이터에 의해 동시에 또는 실질적으로 동시에 출력되면, 브레이킹 펄스(braking pulse)와 같은 날카로운 이벤트(아래 참조)는 강한 이벤트를 너무 이르게 종결시킬 수 있다.
- <61> 일 실시예에 따르면, 저 주파수, 중간 주파수 및 고 주파수 햅틱 이벤트들의 임의의 우선순위는 그것들의 중복

에 관계없이 우선순위 클립 블록(334)에 의해 출력될 수 있다. 일 실시예에서, 주파수들의 서브-대역은, 상이한 서브-대역들로부터의 2개의 햅틱 이벤트들이 서로 중복되는 것으로 검출될 때 우선순위가 할당될 수 있다. 이 우선순위는 음악의 특정한 장르; 노래의 템포; 액추에이터의 해상도; 및/또는 입력 오디오 신호의 정보량(예를 들어, 손실 대 무손실)에 기초하여 할당될 수 있다.

<62> 도 3에 도시된 바와 같이, 우선순위 클립 블록(334)은 효과 맵핑 블록(336)에 우선순위화된 햅틱 이벤트들을 출력한다. 일 실시예에서, 효과 맵핑 블록은 우선순위화된 햅틱 이벤트들을 수신하고, 이들 개별 이벤트들에 햅틱 효과들을 할당한다. 일 실시예에서, 시스템은 주기 의존적이고 크기 의존적인 2개의 기본 햅틱 효과 유형들을 사용한다. 주기적인 효과들은 진동의 주기 또는 속도가 크기에 관계없이 명시적으로 제어될 수 있게 한다. 주기적인 효과들은 디바이스에 따라 진동 주파수들을 약 30Hz까지 정확하게 생성한다. 크기 의존적인 효과들은 크기 의존적인 레이트(rate) 및 디바이스 의존적인 레이트로 진동한다.

<63> 또한, 이 2개의 효과 유형들은 강한, 부드러운 및 날카로운의 3개의 액추에이션 스타일들 또는 효과들로 특성화된다. 각각의 스타일은 사용자에게 약간 상이한 느낌을 일으키는 상이한 제어 매커니즘을 갖는다. 강한 효과들은 사용자에게 가장 강한 느낌을 주는 효과들이다. 부드러운 효과들은 주기적인 효과들에 대하여 고 주파수들을 달성하기 위해 효과의 세기가 감소하는 특성을 갖는다. 크기 의존적인 효과들에 대하여, 효과의 세기는 감소하지 않는다. 날카로운 효과들은, 세기가 크기 의존적인 효과들에 대하여 상당히 감소하더라도, 세기를 감소시키지 않고 주기적인 효과들에 대하여 최고 주파수들을 달성한다.

<64> 각각의 효과는 시스템 또는 사용자에게 의해 설정될 수 있거나 조정가능한 파라미터들을 포함한다. 주기적인 햅틱 효과들에 대한 이러한 파라미터들은 크기, 주기/주파수, 지속기간, 어택 시간, 어택 레벨, 페이드 시간, 페이드 레벨 및 엔벨로프(envelope)를 포함하지만, 이에 한정되지 않는다. 크게 의존적인 햅틱 효과들에 대한 파라미터들은 크기, 지속기간, 및 엔벨로프를 포함하지만, 이에 한정되지 않는다. 기술된 햅틱 효과들 및 상기 도시된 할당들은 단지 예시적이며, 햅틱 효과들 및 할당들의 크기를 사용할 수 있다는 것을 인식해야 한다. 또한, 생성된 효과들은 세그먼트에 걸쳐 보통 신호의 최대값(0-10000으로 정규화됨)으로 취해지는 선택적인 크기 규격을 갖는다.

<65> 도 8은 전송된 입력 오디오 신호와 연관된 악곡에 대한 햅틱 이벤트들의 시퀀스의 예시적인 시간-심볼 표현을 도시한다. 도 8에 도시된 바와 같이, 타임-심볼 표현(800)은 입력 오디오 신호(802)의 시간-주파수 그래프를 포함한다. 또한, 그 표현(800)은 베이스(Bass)(804)로 지정된 저 주파수 서브-대역; 하이 패스(High Pass)(806)로 지정된 고 주파수 서브-대역; 및 음성(Vocal)(808) 및 미드 악센트(Mid Accent)(810)로 지정된 2개의 중간 주파수 서브-대역들을 포함한다.

<66> 도 8에 도시된 바와 같이, 시스템은 한번에 하나의 햅틱 이벤트만이 발생하고 지정된 지속기간에 의해 인접하는 햅틱 효과들로부터 분리되도록 각각의 서브-대역에서 햅틱 이벤트들을 맵핑한다. 일 실시예에서, 시스템은 각각의 이벤트에 대한 햅틱 효과 프로파일(profile)이 서로 유사하지 않은 경우에 2 이상의 햅틱 이벤트가 한번에 발생할 수 있게 한다. 예를 들어, 도 8에 도시된 바와 같이, 시스템은, 강한 크기 의존적인 효과들(808A, 808B) 및 짧은 주기적인 효과들(804A, 806A)이 사용자에게 의해 2개의 개별 이벤트들로서 효과적으로 느껴지는 것을 고려하여, 그것들이 중복될 수 있게 한다. 도 8에 도시된 예시적인 표현에서, 시스템은 고 주파수 햅틱 이벤트들보다 저 주파수 햅틱 이벤트들에 우선순위를 부여하도록 구성되는 한편, 중간 주파수 이벤트들은 고 주파수 이벤트 및 저 주파수 이벤트와 중복되도록 허가된다. 시스템은 임의의 주파수 서브-대역 또는 주파수 서브-대역들 모두에 대한 우선순위 및 중복하는 권한들을 부여하도록 사용자, 엔드 디바이스 또는 입력 오디오 신호 자체에 의해 구성가능하며, 도 8에 도시된 것에 한정되지 않는다는 것에 유의해야 한다.

<67> 이상은, 시스템이 샘플링된 오디오 신호에 대응하고 사용자에게 기분 좋은 방법으로 출력되는 햅틱 이벤트들을 자동으로 수신하고, 식별하고, 생성할 수 있는 하나 이상의 실시예들을 논의하고 있다. 또한, 일 실시예에서, 수신된 파일은 시스템에 의해 실시간 방법으로 처리되며, 그에 의해 파일은, 시스템이 임베디드 데이터로부터 햅틱 이벤트들을 더 쉽고 신속하게 처리하고 출력하도록 압축되고 내부에 햅틱 이벤트 정보를 포함하는 것이 가능하다. 특히, 임베디드 데이터는 바람직하게는 MP3, MP4, AAC, ASX, Ogg 및 WMA 포맷들을 포함하는 압축된 디지털 오디오 파일 포맷으로 되지만, 이에 한정되지 않는다. 압축된 디지털 오디오 포맷들은 엔드 디바이스에 의한 자동화된, 실시간의 및/또는 오프라인 사용에 적합하며, 그에 의해 압축된 오디오 파일은 햅틱 이벤트들을 처리하고 생성하기 위해 시스템에 의해 사용되는 주파수 서브-대역 정보를 저장할 수 있다.

<68> 오디오 파일에 이미 들어 있는 서브-대역 주파수 계수들을 사용함으로써, 시스템은 도 13과 관련하여 기술된 집약적(intensive) 계산들을 수행할 필요가 없다. 즉, 시스템은 오디오 파일 내의 이미 존재하는 주파수 서브-대

역 계수들을 사용하여, 디지털 오디오 파일의 이미 존재하는 서브-대역 값들을 사용함으로써 신속한 방법으로 헵틱 이벤트들을 생성할 수 있다. 도 12는 임베디드 데이터와 주파수 서브-대역 계수들을 사용하여 헵틱 이벤트들을 출력하는 시스템의 블록도를 도시한다. 특히, 시스템(1200)은 바람직하게는 프로세서(1202), 메모리(1204), 디코더(1206), 헵틱 플레이백 엔진(1208) 및 액추에이터(1210)를 포함한다. 도 12에 도시된 바와 같이, 압축된 오디오 파일은 디바이스(1200)에 의해 무선으로 또는 유선 접속을 통해 수신되며, 그에 의해 오디오 파일은 바람직하게는 메모리(1204)에 저장된다. 오디오 파일은 바람직하게는 서브-대역 주파수 계수 정보의 형태로 임베디드 데이터를 포함한다. 바람직하게는, 임베디드 데이터는 각각의 프레임에 대하여 서브-대역 주파수 계수들을 추출하기 위해 엔드 디바이스 내의 디코더(1206)에 의해 디코딩된다. 대안적으로, 디코딩은, 원격 서버와 같은 엔드 디바이스 이외의 디바이스에서 수행될 수 있다. 추출된 정보는, 추출된 정보를 저장되거나 또는 미리 정의된 헵틱 효과들과 연관시키는 헵틱 플레이백 엔진(1208)에 제공된다. 바람직하게는, 디바이스(1200)는 액추에이터(1210)에 신호들을 출력하기 위해 헵틱 효과들을 연관된 오디오 이벤트들과 동기화시키며, 그에 의해 액추에이터(1210)는 헵틱 이벤트들을 출력한다. 일 실시예에서, 파일이 디바이스(1200)에 의해 일종의 미세 튜닝 또는 자기 보정 피드백 방법으로서 플레이되기 때문에 파일의 임베디드 데이터는 디바이스(1200) 자체에 의해 수정된다. 이것은 출력 사운드 피쳐들이 디바이스의 스피커들에 의해 플레이백 될 때 마이크로폰을 통해 출력 사운드 피쳐들을 분석하는 디바이스(1200)에 의해 행해질 수 있다. 예시에서, 시스템은 헵틱 이벤트들의 크기들이 디바이스(1200)에 의해 처리되고 플레이된 후에 그것들을 수정하기 위해 서브-대역 주파수 계수들을 헵틱 플레이백 엔진(1208)의 커널에 전송할 수 있다. 다음으로, 수정된 크기들은 디바이스(1200)에 의해 저장될 수 있으며, 그에 의해 다음번에 파일이 디바이스(1200)에 의해 플레이될 때, 디바이스(1200)는 메모리로부터 수정된 값들을 검색하고 수정된 데이터를 사용하여 헵틱 이벤트들을 플레이한다.

<69> 디코더(1206)가 오디오 파일을 수신하면, 디코더는 서브-대역 계수들을 분석하고 추출하여 헵틱 이벤트들의 빠른 처리 및 출력을 허가한다. 일례에서, MP3 파일의 최저 서브-대역이 일반적으로 100Hz보다 작은 주파수들을 포함한다는 것을 고려하면, MP3 파일의 최저 서브-대역은 비트 또는 리듬 검출에 유용하다. 시스템(1200)은 액추에이터에 신호들을 출력하도록 헵틱 플레이백 엔진(1208)에 지시하기 위해 임의의 적절한 알고리즘을 사용하며, 그에 의해 액추에이터는 오디오 이벤트들과 연관된 헵틱 이벤트들을 출력할 것이다. 특히, 헵틱 플레이백 엔진은 알고리즘에 의해 설정된 임계 주파수를 가로지르는 모니터링된 오디오 이벤트들에 기초하여 어떤 오디오 이벤트들이 대응하는 헵틱 효과를 수신하는지를 판정할 것이다. 예시적인 알고리즘은 샘플 오디오 데이터를 분석하는 것과 관련하여 사용된 전술된 알고리즘이다. 헵틱 플레이백 엔진(1208)은 메모리(1204)로부터 저장된 헵틱 효과 정의들을 검색할 수 있으며, 그에 의해 정의들은 액추에이터에 의해 출력될 효과들의 유형(예를 들어, 진동, 충격(jolt))뿐만 아니라 이벤트 지속기간, 크기 및/또는 주파수를 포함할 수 있다. 액추에이터(1210)는, 헵틱 플레이백 엔진(1208)으로부터 정보를 수신하면, 연관된 사운드의 플레이백과 자동으로 동기화되는 정의된 헵틱 특성을 갖는 헵틱 이벤트들을 출력할 것이다. 즉, 시스템(1200)은, 모든 베이스 또는 저 주파수 비트들이 그것들과 연관된 일관된 헵틱 효과를 갖는 것을 보장하도록, 저 주파수 서브-대역에서의 각각의 헵틱 이벤트에 대하여 정의된 헵틱 효과들을 실시간으로 또는 오프라인 모드로 출력할 것이다.

<70> 보컬 및 그외 고 피치 사운드 등의 고 주파수 사운드에 대하여, 시스템(1200)은 복수의 서브-대역으로부터 에너지를 부가하여, 고 주파수 사운드에 부합하는 헵틱 이벤트를 생성한다. 특히, 오디오 내의 기본적인 중간 영역 사운드들 및 보이스(voice)에 대해서는 일반적으로 100Hz보다 높고 400Hz보다 낮은 주파수 서브-대역이 매우 적합하다. 시스템(1200)은 바람직하게는, 고 주파수 서브-대역에서 에너지에 따라 그 크기가 변하는 낮은 크기-의존 효과를 출력한다. 시스템(1200)은 바람직하게는, 미처리된 에너지(raw energy)에 저역 필터링을 적용하여 에너지의 의사(spurious) 피크들을 제거함으로써, 시스템은 소프트웨어를 이용하여 실시간으로 에너지를 저역 필터링할 수 있다. 특히, 시스템은, 헵틱 플레이백 엔진(1208)이 동작하는 주파수에 그 길이가 기초하는, 비중복(non-overlapping) 윈도우들에 대한 에너지를 평균함으로써, 저역 필터링된 에너지를 얻는다. 일 실시예에서, 헵틱 플레이백 엔진은 200Hz에서 동작하고, 신호는 5ms에 대해 평균될 수 있으며, 헵틱 플레이백 엔진(1208)에는 크기-의존 헵틱 효과의 크기를 나타내는 대응값이 송신될 수 있다. 저역 필터링 후의 잔여 피크들은 평균값의 임계치 위의 헵틱 이벤트들을 트리거링하는 데 이용될 수 있으며, 액추에이터(1210)에 의해 출력되도록 명령되는 헵틱 이벤트들의 지속기간(duration)은 피크들 자신의 지속기간에 의존할 것이다. 예를 들어, 시스템(100)에 의해 피크의 긴 지속기간을 보이스 또는 기타 음(guitar note)의 유지 음(sustaining note)이 되게 상정함으로써, 헵틱 플레이백 엔진(1208)은 그 특정 유지 음에 대응하는 지속기간을 갖는 헵틱 효과를 출력하도록 액추에이터(1210)에 명령한다.

<71> 그 외 400Hz보다 훨씬 더 높은 서브-대역들의 에너지가 존재할 수 있기 때문에, 시스템(1200)은 이들 잔여 서브-대역들의 피크들을 고려한다. 이는 시스템(100)이 심벌즈(cymbals) 및 그 외 고 피치 사운드들에 대응하는 헵

틱 효과를 출력하는 것을 보장한다. 전술한 바와 같은 임계치 알고리즘이 적절하지만, 다른 알고리즘들이 고려될 수 있다. 본 실시예는 도 3에 도시된 우선순위 클립(334) 및 효과 맵핑(336) 모듈들을 이용하여 보다 명확하고 분별력 있는 햅틱 이벤트들을 생성할 수 있다는 것에 주목해야 한다. 그러나, 디지털 오디오 파일은 필요한 클립핑 및 효과 맵핑 정보를 이미 포함하는 것을 조건으로 하여 이들 모듈들을 옵션이 되게 할 수 있다.

<72> 또한, 시스템은 MIDI 파일들을 포함하되 이에 한정되지는 않는 표상적으로(representationally) 구성된 인코딩 음악 파일들을 수신하고 처리하여, 인코딩 음악 파일들에 부합하는 햅틱 이벤트들을 생성할 수 있다. 본 명세서에서는 표상적으로 구성된 파일들을 음악 파일로서 기술하였지만, 표상적으로 구성된 파일들은 사운드 효과, 또는 오디오와 관련된 다른 이벤트를 갖는 파일들일 수 있다는 것에 주목해야 한다.

<73> 도 9는 일 실시예에 따른 악곡을 구성하는 일련의 MIDI 이벤트들의 시간-음표 표현이다. MIDI 파일 포맷은 동시에 일어나는 악기의 16 채널을 포함할 수 있으며, 각 채널은 플레이될 수 있는 127개의 고유 피치(키(key)라고 명명됨)까지의 이벤트를 가질 수 있다. 또한, MIDI 파일 포맷은 일반적으로 피치가 없는 타악기를 위해 보유되는 채널을 포함한다. 타악기들은 타악기 채널 내의 키들의 범위에 걸쳐 분포된다.

<74> 도 10은 일 실시예에 따른 악곡을 구성하는 일련의 MIDI 이벤트들의 시간-악기-음표 표현을 나타낸다. MIDI 음악 파일에서, 모든 악기의 모든 음표는 시스템에 의해 햅틱 이벤트로 지정될 수 있는 이벤트이다. 그러나, 액추에이터에 대한 물리적 한계 및 성능 한계를 고려하여, 시스템은 MIDI 파일들을 분석하고 처리하여 사용자에게 기분좋은 햅틱 경험을 생성하는 개별 햅틱 이벤트들을 생성한다.

<75> 도 11a, 도 11b, 및 도 11c는 모두, MIDI 파일들을 자동으로 처리하고, 엔드-플레이어 디바이스 내의 개별 햅틱 이벤트들을 생성하기 위해 시스템에 의해 사용되는 처리 흐름도를 나타낸다. 일반적으로, 시스템은 MIDI 데이터를 분석하고, 구조화된 표현(structured representation)으로부터 멜로디 및/또는 타악기 이벤트 시퀀스를 추출한다. 그 후 시스템은 각 악기에 대응하는 각 채널에 대한 스코어를 생성하고, 타악기가 아닌(non-percussive) 채널의 스코어는 해당 채널의 음표들의 가변성에 관련된다. 이는 경험적으로 달리 알려지지 않은 입력 채널들의 세트로부터 멜로디를 선택하는데 효과적인 것으로 발견되어 왔다. 타악기에 있어서, 스코어링 전략은 각종 타악기의 우세(dominance)에 대한 선행 지식에 의존한다. 예를 들어, 베이스 드럼은 가장 두드러진 타악기 채널일 것이기 때문에, 타악기 채널 내의 베이스 드럼에 관련된 음표들은 일반적으로 높은 스코어를 받는다.

<76> MIDI 시퀀스로부터 입력 채널들을 스코어링하는 것 외에, 시스템 및 방법은 이벤트들에 대해 하나 이상의 클립핑 오퍼레이션을 수행하여, 개별의 햅틱 이벤트들이 엔드 디바이스에서 생성되는 것을 가능하게 한다. 일 실시예에서, 시스템은 하나 이상의 음표들을 원하는 지속기간으로 클립핑하고, 채널들을 처리하여 화음 등의 중복되는 음표들을 제거한다. 일 실시예에서, 시스템은 다른 채널들에 대해 하나 이상의 채널(예컨대, 타악기 대 멜로디)의 상대 우선순위를 제어하도록 조작할 수 있는 다수의 파라미터들에 기초하여 구성될 수 있다.

<77> 도 11a에 도시된 바와 같이, MIDI 시퀀스의 선택된 하나, 일부, 또는 모든 채널들은 시스템(1102)에서 수신된다. 시퀀스는 엔드 디바이스, 엔드 디바이스와 통신하는 서버, 및/또는 다른 중간 컴포넌트, 컴퓨터 또는 프로세서 내의 메모리로 импорт(import)될 수 있다. 일 실시예에서, 햅틱 이벤트들을 생성하는데 실제로 사용되지 않는 импорт된 MIDI 파일로부터의 정보는 메모리로부터 실제 삭제된다. 일 실시예에서, 햅틱 이벤트들을 생성하는 데 실제로 사용되지 않는 정보가 시스템에 의해 보유되어, 개발자 또는 엔드 유저에 의해 햅틱 이벤트들의 재구성 또는 미세 튜닝이 가능하게 한다.

<78> 일 실시예에서, 시스템은 파일에 대하여 그것이 수신된 후에 각종 전처리 절차를 실행한다. 일 실시예에서, 시스템은 이벤트(도 11a의 블록(1104))의 타임라인을 분석함으로써 이벤트에 대한 이전-음표(pre-note) 식별 기법 및 다음-음표(next-note) 식별 기법을 실행한다. 블록(1104)에서, 일 실시예에서, 하나의 이벤트로부터 동일한 채널 내의 이전 이벤트에 이르는 링크들을 셋업하도록 각 채널이 처리된다. 또한, 하나의 이벤트로부터 동일한 채널 내의 다음 이벤트에 이르는 링크들을 셋업하도록 각 채널이 처리된다. 일 실시예에서, 시스템은 구성된 오디오 표현 파일 전체의 타임라인을 일시에 분석한다. 다른 실시예에서, 시스템은 오디오 표현의 타임라인의 부분들을 일시에 분석한다. 이는, 시스템으로 하여금 긴 타임라인을 갖는 큰 오디오 파일들을 신속하게 처리 가능하게 한다(예컨대, 영화에서 사운드 효과).

<79> 도 11a에 도시된 바와 같이, 그 후 시스템은 멜로디 및 타악기 이벤트들이 발생하는 채널들을 식별함으로써 블록(1106)에 나타난 바와 같이, 관심 대상의 멜로디 이벤트 및 타악기 이벤트를 식별한다. 멜로디 이벤트 및 타악기 이벤트를 식별 시에, 시스템은 식별된 이벤트들을 효과적으로 이용하여 도 11b의 멜로디 이벤트들(블록

1116) 및 타악기 이벤트(블록 1132)를 스코어링할 수 있다.

<80> 도 11에 도시된 바와 같이, 시스템은 일 실시예에서, 각 채널을 선택적으로 처리하여 화음들을 제거한다(1108). 일 실시예에서, 시스템은 MIDI 파일로부터의 화음들을 동일한 채널에 대해 서로 소정 지속기간 내에서 시작하고 타악기가 아닌 채널들 상에 있는 이벤트들로서 식별한다. 일 실시예에서, 이 지속기간은 15ms로 설정되지만, 다른 지속기간도 고려된다. 일 실시예에서, 일단 시스템이 MIDI 이벤트 채널들 내의 화음들을 식별하면, 시스템은 최상위 피치를 갖는 이벤트만을 보유하고, 타임라인으로부터 화음을 구성하는 다른 이벤트들을 폐기한다. 다른 실시예에서, 시스템은 최상위 피치 이벤트들을 제거하고, 하위 또는 최하위 피치 이벤트들을 보유하고. 이하 상술하는 바와 같이, 타임라인으로부터 화음들의 제거는 다수의 화음을 갖는 채널에 부여된 스코어를 감소시켜, 타임라인의 멜로디들을 식별하는 데에 있어 시스템을 돕는다.

<81> 참조번호 1110에서, 시스템은 동시 또는 거의 동시에 일치하는 또는 거의 일치하는 음표를 갖는 채널 쌍들을 갖는 채널 상들을 식별한다. 이는 주로 2개 이상의 악기들이 하모니로 연주되는 경우이다(예컨대, 2개의 기타가 동일한 리프(riff)로 동시에 연주되는 경우). 각 채널에 있어서, 온타임즈(on-times)의 타임 샘플링 벡터가 생성되는데, 상기 온타임즈는 이벤트들이 발생하는 포인트들이다. 시스템은 타임라인에 대해 각 채널의 이벤트들을 조사하고 이벤트가 발생하는 시간 간격을 값 1로 설정함으로써 이를 달성한다. 그 외에는 벡터에 값 0이 설정된다. 이는 길이 n인 벡터들의 세트를 생성한다.

수학식 2

$$n = \text{floor}\left(\frac{T}{t_s}\right)$$

<82> 여기서, T는 MIDI 입력 파일의 총 지속기간이며, t_s 는 샘플링 간격이다.

<84> 중복되는 음표들을 갖는 채널들을 식별하기 위하여, 채널들의 각각에 대해 상관이 계산된다.

수학식 3

$$\text{cor}(v_i, v_j) = \frac{\text{cov}(v_i, v_j)}{\sigma_i \sigma_j} = \frac{\langle v_i v_j \rangle - \langle v_i \rangle \langle v_j \rangle}{\sigma_i \sigma_j}$$

<85> 여기서, v_i, v_j 는 타임 샘플링 벡터들이고, σ_i 는 시계열로 처리되는 벡터 v_i 의 표준 편차이다. 2개 이상의 채널의 상관의 상관 임계값을 초과하는 경우, 타임라인으로부터 일부 채널 또는 전체 채널이 제거될 수 있다.

<87> 예를 들어, 도 9를 다시 참조하면, 라인 900과 라인 908 사이에는 채널들(902, 904)이 실질적으로 동일하게 도시되어 있다. 시스템이 채널들(902, 904)에 대한 상기 계산을 수행하면, 시스템은 이들이 동일한 멜로디를 동시에 연주하는 것으로 상정하여 2개의 채널 중 하나를 제거할 것이다. 2개의 채널 중 어느 채널이 제거될 것인지 사용자 및/또는 시스템에 의해 설정된 파라미터에 의존할 수 있다.

<88> 도 11a에 도시된 바와 같이, 시스템은 클리핑 처리(1112)를 이용하여 최대 및 최소 길이를 설정함으로써 이벤트들을 원하는 길이로 제한한다. 시스템은 지정된 최소 지속기간보다 짧은 이벤트들을 적어도 최소 지속기간의 길이가 되도록 클리핑한다. 또한, 지정된 최대 지속기간보다 긴 이벤트들이 최대 지속기간보다 길지 않도록 변경되거나 또는 클리핑된다. 또한, 이들 사이에 미리 지정된 최소값보다 작은 스페이싱된 지속기간을 갖는 2개 이상의 이벤트들이 적어도 최소의 스페이싱된 지속기간을 갖도록 변경된다. 이는 시퀀스 내의 이벤트들 중 하나 또는 양자 모두를 짧게 하여 그들 사이에 최소의 스페이싱을 생성함으로써 행해질 수 있다.

<89> 이제 도 11b를 참조하면, 일단 시스템이 각 채널의 임포팅 및 전 처리를 완료하면, 시스템은 수신 파일 내의 각 채널의 스코어링을 개시한다(1116). 일 실시예에서, 시스템은 수신 파일의 임포팅/전처리가 실행되고 있을 때 스코어링 처리를 시작할 수 있다는 것에 주목해야 한다. 일 실시예에서, 시스템은 파일의 임포팅/전처리가 수행된 후에 스코어링 처리를 시작한다.

<90> 일 실시예에서, 시스템은 두 세트의 규칙을 이용하여 채널들을 스코어링한다. 제1 규칙 세트는 멜로디를 운반할 가능성이 있는 채널들로서 소정의 가변성을 갖는 채널들에 높은 스코어를 부여하도록 설계된다. 제2 규칙 세트는 노래의 리듬을 정의함에 있어서 종래의 중요도에 의해 타악기 채널들에 순위를 부여한다.

- <91> 타임라인 내의 이벤트 모두를 검사함으로써 스코어링이 완성된다. 각 이벤트는 스코어를 개별 이벤트에 할당하는 스코어링 필터를 통과한다. 일단 모든 이벤트가 처리되면, 타임라인을 통해 제2 루프가 각 채널의 모든 이벤트에 대한 스코어를 유도하고, 채널당 총 스코어를 산출한다. 일 실시예에서, 이는 이벤트들의 스코어들을 간단히 누적함으로써 행해진다. 다른 실시예에서, 이는 적절한 알고리즘을 이용하여 행해진다. 도 11b에 도시한 바와 같이, 일 실시예에서 시스템은 멜로디 이벤트 및 타악기 이벤트를 별개로 스코어링하지만, 스코어링은 동시에 또는 차례대로 행해질 수도 있다. 일 실시예에서, 시스템은 멜로디 이벤트 및 타악기 이벤트를 모두 스코어링할 수 있어, 한 종류의 이벤트(예컨대, 멜로디 또는 타악기)에 대해 결정된 정보가 다른 타입의 이벤트(예컨대, 타악기 또는 멜로디)를 분석하거나 평가하는 데 이용될 수 있다.
- <92> 타임라인 내의 멜로디 이벤트에 있어서, 시스템은 도 11b의 참조번호 1116에 도시된 바와 같이, 이벤트들을 스코어링하는 데에 서브세트의 규칙들을 이용한다. 일 실시예에서, 서브세트 규칙들의 각각은 시스템에 의해 동시에 실행된다. 일 실시예에서, 하나 이상의 서브세트 규칙들은 시스템에 의해 시퀀스로 실행된다.
- <93> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1118)에 따라 이벤트들 간의 피치 변화를 스코어링한다. 블록(1118)에서와 같이, 시스템은 현재 이벤트의 피치와 하나 이상의 이전의 이벤트들의 피치 사이의 차이의 절대값을 계산함으로써, 피치들 각각은 결정된 값을 갖는다. 시스템은 오디오 표현 파일(audio representation file) 내에 고유하게 인코딩된 채널 데이터로부터 피치에 값을 할당할 수 있다. 실시예에 있어서, 차이는 피치 스코어로서 출력된다. 실시예에 있어서, 차이값은 미리 결정된 인수(예를 들어, 500)에 의해 스케일링되어 피치 스코어를 얻는다.
- <94> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1120)에 따라 이벤트들 간의 시작 시간을 스코어링한다. 블록(1120)의 시스템은 리듬이 채널에서 플레이되는지의 여부를 결정하기 위해 이벤트들의 지속기간을 분석한다. 일 실시예에서, 블록(1120)에서는 이벤트와 직전의 이벤트 사이에서 시작 시간들의 차이가 계산된다. 일 실시예에서, 블록(1120)에서는 선행하는 이벤트와 선행하는 이벤트 이전의 이벤트 사이에서 시작 시간들의 차이가 계산된다. 일 실시예에서, 시스템은 시작 시간 스코어를 출력하기 전에 차이들에 가중치를 부여한다. 일 실시예에서, 가중된량은 인수 5이다. 그러나, 가중값은 5 이외의 인수일 수 있다는 것이 고찰될 수 있다. 파일 내의 다른 리듬들 또는 노래의 장르에 따라, 시스템은 그 시스템이 적절하게 협화음으로된 헵틱 이벤트들을 출력하도록 지속기간을 보다 높거나 낮게 스코어링할 수 있다.
- <95> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1122)에 따라 이벤트의 크기 또는 강도를 스코어링한다. 블록(1122)은 이벤트의 크기의 함수에 기초하여 동일한 스코어를 지정한다. 시스템은 구조화된 표현 파일 내에 고유하게 인코딩된 채널 데이터로부터 각각의 이벤트의 크기를 결정할 수 있다.
- <96> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1124)에 따라 이벤트의 지속기간을 스코어링한다. 블록(1124)은 이벤트의 지속기간의 함수에 기초하여 스코어를 지정한다. 실시예에 있어서, 시스템은 보다 짧은 지속기간을 갖는 이벤트보다 더 긴 지속기간을 갖는 이벤트들에 대해 보다 높은 스코어를 할당한다. 다른 실시예에 있어서, 시스템은 보다 짧은 지속기간을 갖는 이벤트보다 더 긴 지속기간을 갖는 이벤트들에 대해 보다 낮은 스코어를 할당한다.
- <97> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1126)에 따라 이벤트의 지속기간의 변화를 스코어링한다. 특히, 시스템은 블록(1126)에서 현재 이벤트와 이전의 이벤트 사이의 지속기간의 변화에 기초하여 스코어를 지정한다.
- <98> 도 11b에 도시된 바와 같이, 시스템은 실시예의 블록(1128)에 따라 이벤트의 지속기간의 변화를 스코어링한다. 특히, 시스템은 블록(1128)에서 이벤트 내의 각각의 음표 전의 침묵(silence)의 지속기간에 기초하여 스코어를 지정한다. 예를 들면, 만약 시스템이 특정 채널에서 긴 침묵 지속기간 후에 발생하는 이벤트를 식별하는 경우에는, 시스템은 헵틱 이벤트로서 가장 출력할 가능성이 많은 이벤트에 대해 높은 스코어를 할당할 수 있다. 이것의 예로는 드럼 롤의 끝에서의 심벌 소리일 수 있다.
- <99> 일 실시예에 있어서, 블록(1130)에 의해 도시된 바와 같이, 멜로디 이벤트들을 시스템에 의해 그들의 엔트로피(entropy)와 관련지어 선택적으로 스코어링할 수 있다. 규칙들을 평가하는 처리에 있어서, 시스템은, 채널에 할당된 스코어들과 해당 채널과 연관된 음향 엔트로피 사이의 관계를 이용할 수 있다. 각각의 멜로디 채널의 엔트로피를 평가하기 위하여, 시스템에 의해 음표들의 분포를 구축할 수 있다. 전통적인 12 톤의 스케일을 이용하여 음표 피치들의 등가 분류들 $\{p(k), k=1..12\}$ 을 정의한다. 각각의 이벤트는 그 12 톤 스케일 중 이벤트의 키와 관련된 톤에 따라 분류된다. 분포는 각각의 이벤트 분포를 지각적인 악센트 규칙을 이용하여 스케일링

함으로써 구축되어, 가중계수 w 는 수학식 4에 의해 결정된다.

수학식 4

$$w = \left(1 - e^{-\frac{d}{\tau}}\right)^{\alpha}$$

<100>

<101> 여기서 변수 d 는 현재 이벤트와 이전의 이벤트 사이의 침묵의 지속기간(초 단위)이다. 변수 τ 는 포화 휴지 길이(0.5s로 고려함)이고, α 는 최소 판별 기간(2s로 고려함)이다. 포화 휴지 길이 및 최소 판별 기간에 대한 그 외의 값들을 고려할 수 있으며, 여기에 언급된 값들로 제한되는 것은 아님을 유의해야 한다.

<102> 이 분포의 엔트로피는 수학식 5의 공식을 이용하여 계산된다.

수학식 5

$$e = -\sum_{k=1}^{12} p(k) \log(p(k))$$

<103>

<104> 여기서 $p(k)$ 는 (12 톤 스케일에 대해) 1부터 12까지의 톤 피치들의 분류들이다.

<105> 멜로디 이벤트들을 스코어링하는 것에 추가하여, 시스템은 도 11b의 블록(1132)에 도시된 타악기 이벤트들을 스코어링한다. 일 실시예에서, 시스템은 타악기 이벤트들을 스코어링하는 것에 있어서 2개의 규칙들을 활용한다. 특히, 시스템은 각각의 타악기 음표들 전의 침묵의 지속기간을 스코어링한다. 침묵의 지속기간은 밀리초로 고려되며, 일 실시예에서 스코어는 가중된다. 일 실시예에서, 스코어링된 침묵 값은, 다른 값을 고찰할 수도 있지만, 값 4로 나누어진다.

<106> 도 11b에 도시된 바와 같이, 시스템은 타악기의 유형에 따라 각각의 타악기에 값을 할당한다. 일 실시예에서, 더욱 유력한 악기가 더 높은 값을 갖도록 각각의 타악기에 대해 시스템에 의해 1 내지 5의 가중값이 할당된다. 예를 들면, 베이스 드럼 악기에 기여하는 이벤트들에는 값 5가 주어지는 한편, 하이햇(high-hat) 심벌에 기여하는 이벤트들에는 값 1이 주어진다. 일 실시예에서는, 이벤트 당 스코어를 발생함에 있어서 값은 인수(예를 들어, 100)에 의해 가중된다. 타악기 이벤트에 대한 총 이벤트 스코어는 이 규칙들의 서브세트에 기초한다.

<107> 일 실시예에 있어서, 채널들의 절대 랭킹이 생성될 때 타악기에 대한 멜로디 악기를 강조시키기 위하여 타악기 및 멜로디 규칙들 모두에 대한 스코어링 증배기(scoring multipliers)는 가중된다. 이 처리는 채널 스코어가 생성되는 블록(1138)에서 발생한다. 가중치 부여는 음악의 장르, 오디오 파일의 유형(예를 들어, 비디오의 음악 대 사운드 효과), 사용자 선호도, 엔드 디바이스의 유형 등에 기초할 수 있다.

<108> 이제 도 11c를 참조하면, 채널 당 스코어의 생성(블록(1138))에 후속하여, 시스템은 블록(1142)에 의해 도시된 바와 같이, 스코어 값에 기초하여 채널들을 정렬할 수 있다. 일 실시예에서, 시스템은, 시스템이 멜로디 채널과 타악기 채널의 두가지 유형을 함께 정렬할 수 있지만, 멜로디 채널과 타악기 채널을 별도로 정렬한다. 일 실시예에서, 스코어링된 채널들은 내림차순으로 정렬되어, 최고 스코어를 갖는 채널들이 "하이(high)"로서 랭킹된다. 일 실시예에서, 스코어링된 채널들은 오름차순으로 정렬된다. 일 실시예에서, 시스템은 원하는 스코어의 범위 내에 있는 채널들을 정렬하거나 또는 원하는 스코어를 갖는 채널만을 선택한다. 일 실시예에서, 시스템은 최고부터 최저까지의 스코어를 랭킹하여 최고 랭킹의 이벤트들은 햅틱 이벤트로서 출력되며, 최저 랭킹의 이벤트들은 햅틱 이벤트로서 출력되지 않을 것이다.

<109> 시스템이 스코어에 기초하여 채널들을 랭킹 또는 선택 시에, 시스템은 어느 채널이 햅틱 효과가 기초할 이벤트들을 제공할지를 결정할 수 있다. 일 실시예에서, 시스템은 이하의 의사 코드를 사용하여 블록(1144)에서 햅틱 효과들을 생성한다.

While channels remain

For each empty duration > 400ms in the Vibe channel

Fill the empty duration, with a 100ms cushion at the beginning and

end of the gap with the highest scoring channel in that interval.

end

<110>

<111>

즉, 시스템은 최고 랭킹의 햅틱 이벤트들을 노래의 시작부터 끝까지 자동으로 맵핑한다. 그 후에, 시스템은 "블랭크를 채우고", 노래의 빈 부분 또는 햅틱 이벤트가 없는 부분들 내에 배치될 수 있는 그 다음 순위의 최고 랭킹의 햅틱 이벤트들을 다시 맵핑한다. 이 처리는, 액추에이터 해상도, 지속기간 쿠션(duration cushions), 사용자 선호도, 햅틱 효과 스타일 및 노래의 유형을 고려하여, 노래 내에서 허용된 햅틱 이벤트들의 최대 수가 맵핑될 때까지 반복된다. 상기된 것들은 단지 의사 코드의 예일 뿐이며, 시스템은 상이한 지속기간 쿠션들을 활용할 수 있을 뿐만 아니라 최저 스코어링 채널 또는 그외의 조건을 활용할 수 있다는 것을 유의해야 한다.

<112>

이벤트들이 햅틱 이벤트가 되도록 할당됨에 따라, 시스템은 원하는 범위 내에서 지속기간을 갖도록 햅틱 이벤트를 클립핑한다. 일 실시예에서, 클립핑(clipping)의 범위는 200nm와 300nm 사이이지만, 더 길거나 더 짧은 지속기간을 고려할 수 있으며 햅틱 액추에이터의 속성에 의존할 수 있다. 또한, 이벤트의 시작 시간은 원하는 양만큼 오프셋되어 상이한 유형의 액추에이터들에 대한 오디오와 햅틱 이벤트들 간의 지각적인 결합(perceptual coupling)을 보장할 수 있다. 일 실시예에서, 햅틱 이벤트의 강도는 시스템에서 조정될 수 있다. 예를 들면, 최소 지속기간은 낮은 값으로 설정될 수 있으며, 쿠셔닝(cushioning) 시간은 증가될 수 있다.

<113>

도 11c에 도시된 바와 같이, 시스템은 일 실시예에서 타악기 중복(percussive overlap)을 제거한다. 시스템은 선택적으로 '블랭크 채우기' 알고리즘을 이용하여 멜로디 및 타악기 채널을 독립적으로 구축하고, 그 후 그들을 함께 병합하여 최종 맵핑된 햅틱 이벤트들을 생성한다. 이것은, 중복하는 날카로운(sharp) 효과들이 예측불가능한 바이브(vibe) 거동을 내기 때문에, 오직 "강한" 효과들만이 사용될 때 가장 효과적일 가능성이 있다. 대안적으로 시스템이 중복 타악기 이벤트들을 갖는 햅틱 이벤트들의 크기를 감쇄시키는 것이 가능하며, 감쇄된 햅틱 이벤트들은 그 후 최대 크기로 출력된다. 이것은 엔드 디바이스에 멜로디는 유지하면서 리듬에 대한 지각적인 강조는 표시하지 않는 특성을 제공한다.

<114>

일 실시예에서, 시스템은 구조화된 표현 신호의 타임라인에서의 임의의 포인트에서 커스텀 페이드들(Custom Fades)(블록(1150))을 추가할 수 있다. 설계자는 긴 지속기간의 음표를 페이드아웃(fade out)하는 짧은 효과(예를 들어, 500ms)로 대체하는 것을 선택할 수 있다. 또한, 유사한 페이드 효과를 추가함으로써 위상(phase) 또는 멜로디 라인의 끝을 더욱 흥미롭게 만들 수 있다.

<115>

일 실시예에서, 시스템은 전술한 규칙들에 따라 이벤트들이 타임라인에 추가되어짐에 따라 (약 500밀리초보다 큰) 긴 지속기간을 갖는 이벤트들을 찾는다. 또한 특정 이벤트와 그 다음 이벤트 사이에 큰 시간 공백(약 1000밀리초)이 있으면, 시스템은 특정 이벤트를 자동으로 페이드 인(fade in)하거나 또는 페이드 아웃할 수 있다. 이 접근을 구현하기 위해, 시스템은 페이드 인 및/또는 아웃하는 햅틱 효과 특성 또는 프로파일을 갖는 그러한 햅틱 이벤트들의 리스트를 식별하고 구축한다. 전체적인 타임라인이 구성된 후에, 시스템은 기존의 햅틱 이벤트들을 리스트로부터 페이드된 햅틱 이벤트들로 대체한다. 페이드 효과는 그 전체적인 지속기간 동안 감쇠하는 주기적인 또는 크기 의존적인 효과로서 구현될 수 있다.

<116>

전술한 디지털 샘플링된 오디오 및/또는 구조화된 표현 파일들을 변환하기 위한 방법은 음표들의 햅틱 이벤트들의 자동 변환을 허용한다. 일단 변환되면, 이들 햅틱 이벤트 시퀀스들은 적절한 방식으로 다른 엔드 디바이스들에 저장되고 재분배될 수 있다. 예를 들면, 햅틱 이벤트 시퀀스들은 네트워크를 통해 하나 이상의 클라이언트와 통신하고 있는 서버의 출력 파일로서 저장될 수 있다. 클라이언트와의 통신 링크를 수립하면, 서버는 출력 파일을 클라이언트에서 전자적으로 재생되도록 전달할 수 있다. 마찬가지로, 햅틱 시퀀스들은 당분야에 알려진 다양한 설정들에 이용될 수 있다. 예를 들면, 사용자에게 전화 호출, 또는 약속, 벨소리 톤, 알람, 기상 보고나 텍스트 메시지와 같은 수신 메시지 등의 이벤트를 통지하기 위한 것일 수 있다. 일 실시예에서, 구조화된 표현 파일들은, 파일의 자동 변환 후에 엔드 디바이스(예를 들어, 가라오케 마이크로폰)가 햅틱 효과들을 출력하여 가수에게 보컬을 신호보내는 가라오케 노래에 적용될 수 있다. 일 실시예에서, 엔드 디바이스는 비디오, 비디오 게임, 음악 비디오, 또는 엔드 디바이스에 다운로드가능한 영화 파일 내의 사운드 효과 또는 그외의 오디오에 대한 햅틱 이벤트 시퀀스들을 자동으로 변환하고, 맵핑하고, 출력할 수 있다.

<117> 본 발명의 실시예들 및 애플리케이션들이 도시되고 기술되었으나, 본 개시의 이익을 갖는 본 기술분야의 통상의 기술자들에게는 본 발명의 요지를 벗어나지 않는 범위에서 전술한 것보다 더욱 많은 변경이 가능하다는 것이 명백하다. 따라서, 본 발명은 첨부된 특허청구범위의 사상에 의해서만 제한된다.

도면의 간단한 설명

- <118> 도 1은 음표(notes), 화음(chords) 및 쉼표(rests)의 시퀀스를 포함하는 전통 낱장 악보 스타일을 이용하여 시각적으로 인지가능하게 나타낸 악곡의 일부를 나타낸 도면.
- <119> 도 2는 샘플링된 오디오 신호의 일부의 시간-진폭 그래프.
- <120> 도 3은 일 실시예에 따라 햅틱 효과 맵핑 방법을 도시하는 프로세스 흐름도.
- <121> 도 4는 일 실시예에 따른 시스템 피스(system piece)의 전체 구조도.
- <122> 도 5는 일 실시예에 따라 도 4의 악곡에 대한 고 주파수 및 중간 주파수 서브-대역 신호들을 도시하는 필터링된 시간-진폭 그래프.
- <123> 도 6은 일 실시예에 따라 중간 주파수 서브-대역 신호를 분할하기 위해 사용되는 평균 전력 및 순간 전력의 표시를 도시하는 정규화된 시간-신호 그래프.
- <124> 도 7은 일 실시예에 따라 저 주파수 서브-대역 신호를 분할하기 위해 사용되는 주파수-전력 그래프.
- <125> 도 8은 일 실시예에 따라 도 4의 악곡예에 대한 햅틱 효과들의 시퀀스의 시간-심볼 표현을 나타낸 도면.
- <126> 도 9는 일 실시예에 따라 악곡을 구성하는 일련의 MIDI 이벤트들의 시간-음표 표현을 나타낸 도면.
- <127> 도 10은 일 실시예에 따라 악곡을 구성하는 일련의 MIDI 이벤트들의 다른 시간-악기-음표 표현을 나타내는 도면.
- <128> 도 11a, 도 11b 및 도 11c는 일 실시예에 따라 햅틱 효과들의 대응 시퀀스를 산출하기 위한 MIDI 시퀀스 처리 방법을 도시하는 프로세스 흐름도를 함께 구성하는 도면.
- <129> 도 12는 일 실시예에 따른 엔드 디바이스의 블록도.
- <130> <도면의 주요 부분에 대한 부호의 설명>
- <131> 310, 1204: 메모리
- <132> 316, 320: 옵티마이저
- <133> 318: 중간 주파수 세그멘터
- <134> 322: 고 주파수 세그멘터
- <135> 330: 저 주파수 옵티마이저
- <136> 332: 저 주파수 세그멘터
- <137> 340, 422, 1210: 액추에이터
- <138> 402: 서버
- <139> 406: 호스트 컴퓨터
- <140> 408: 모바일 디바이스
- <141> 410: 커머셜 키오스크
- <142> 412: 수신 모듈
- <143> 414, 1202: 프로세서
- <144> 420: 엔드 디바이스
- <145> 1206: 디코더

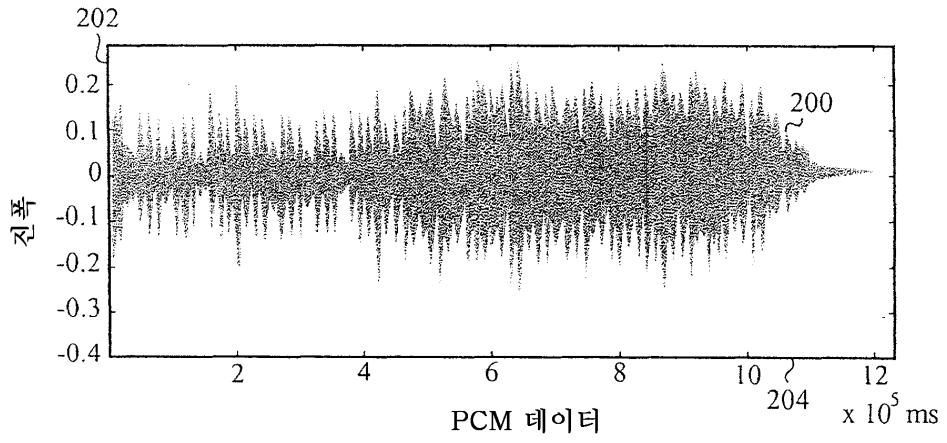
<146> 1208: 헵틱 플레이백 엔진

도면

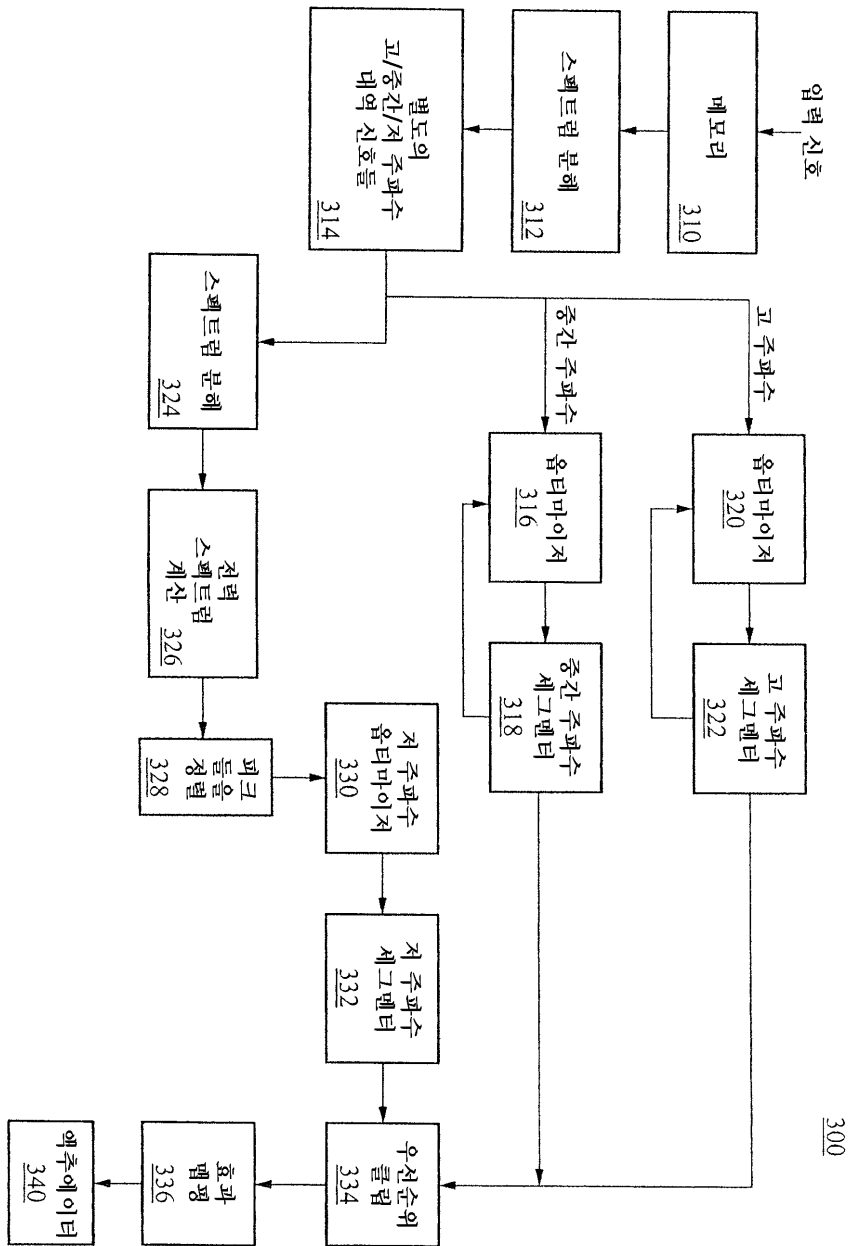
도면1

The image displays two systems of musical notation, likely representing a heptatic playback engine. Each system consists of two staves. The left system includes a treble clef staff with notes and a bass clef staff with notes and chords. The right system is similar but includes a '10φ Code' label. The notation includes various musical symbols such as notes, rests, and dynamic markings. A large arrow labeled '100' points to the right system.

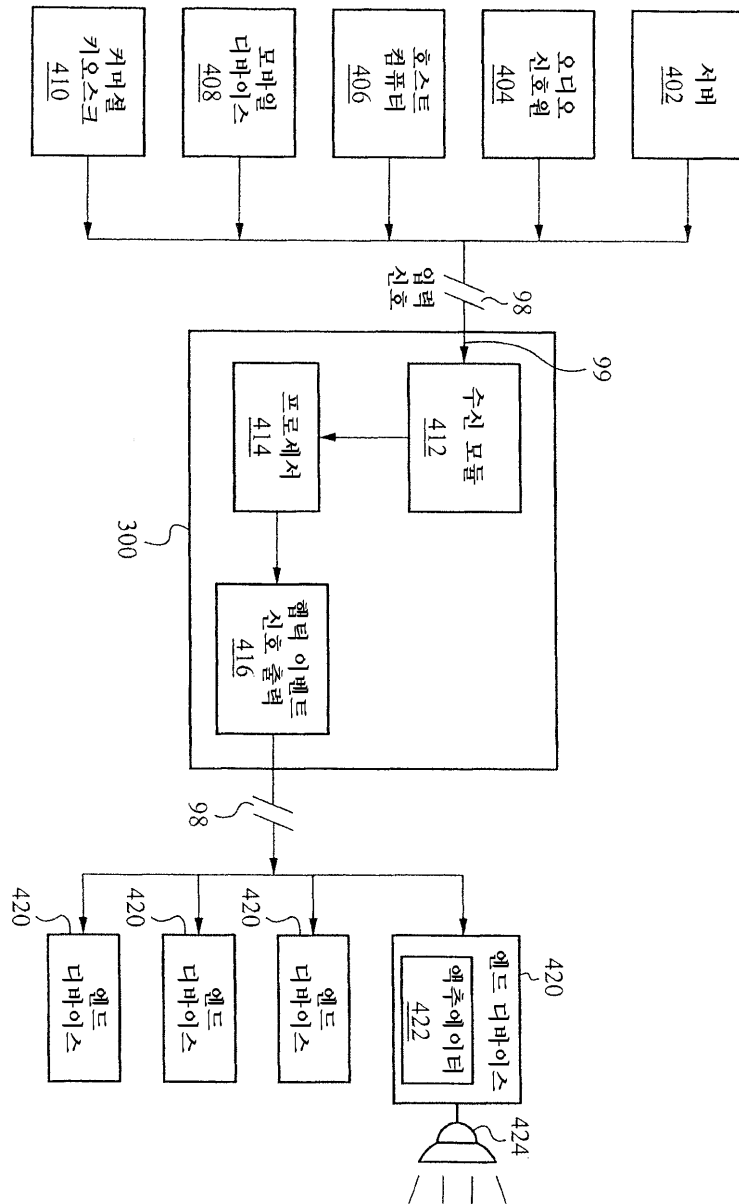
도면2



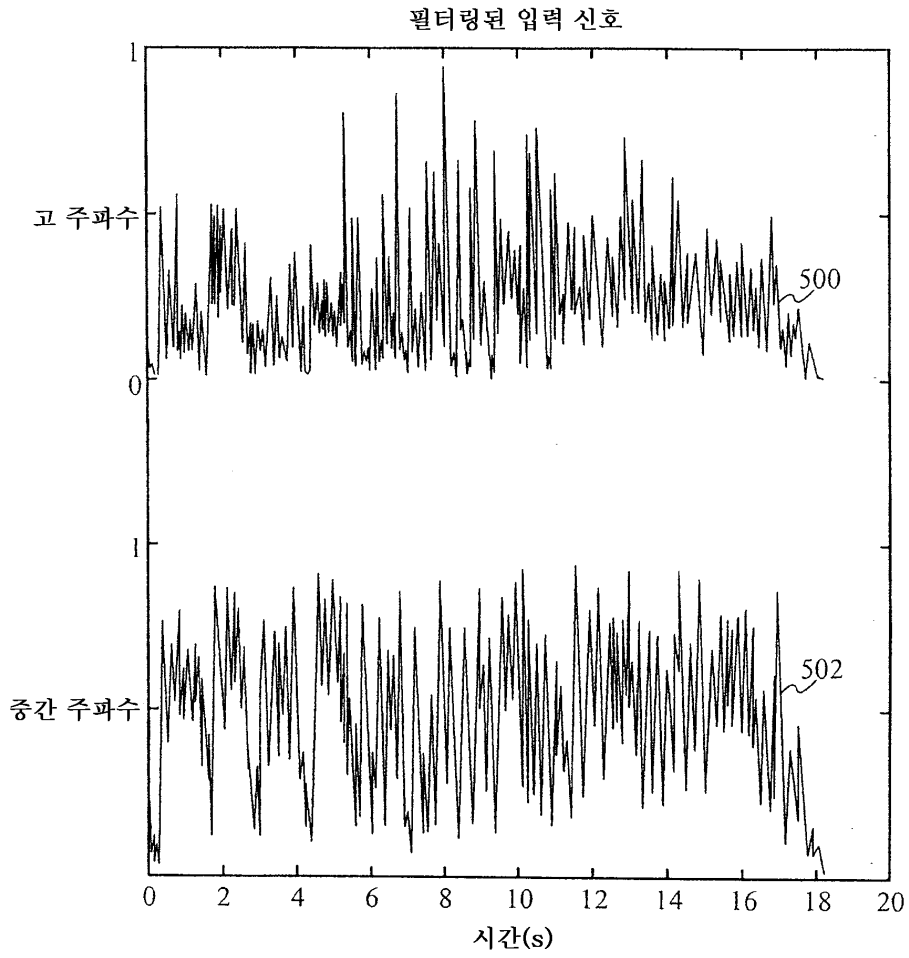
도면3



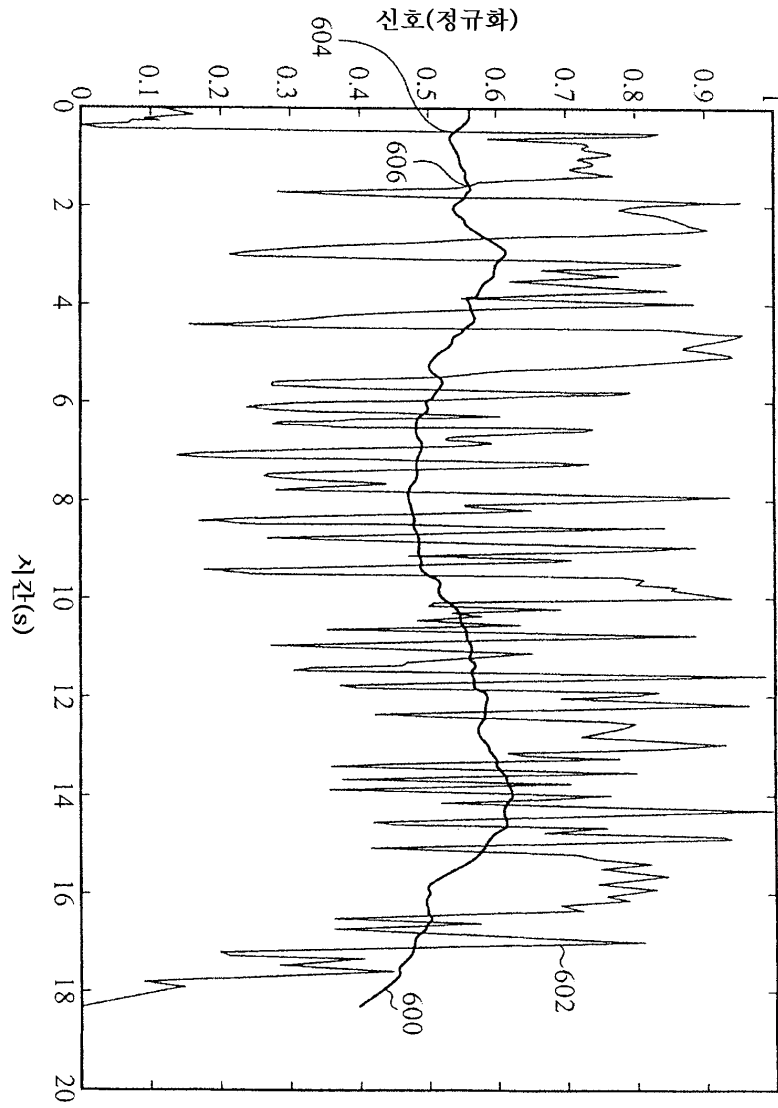
도면4



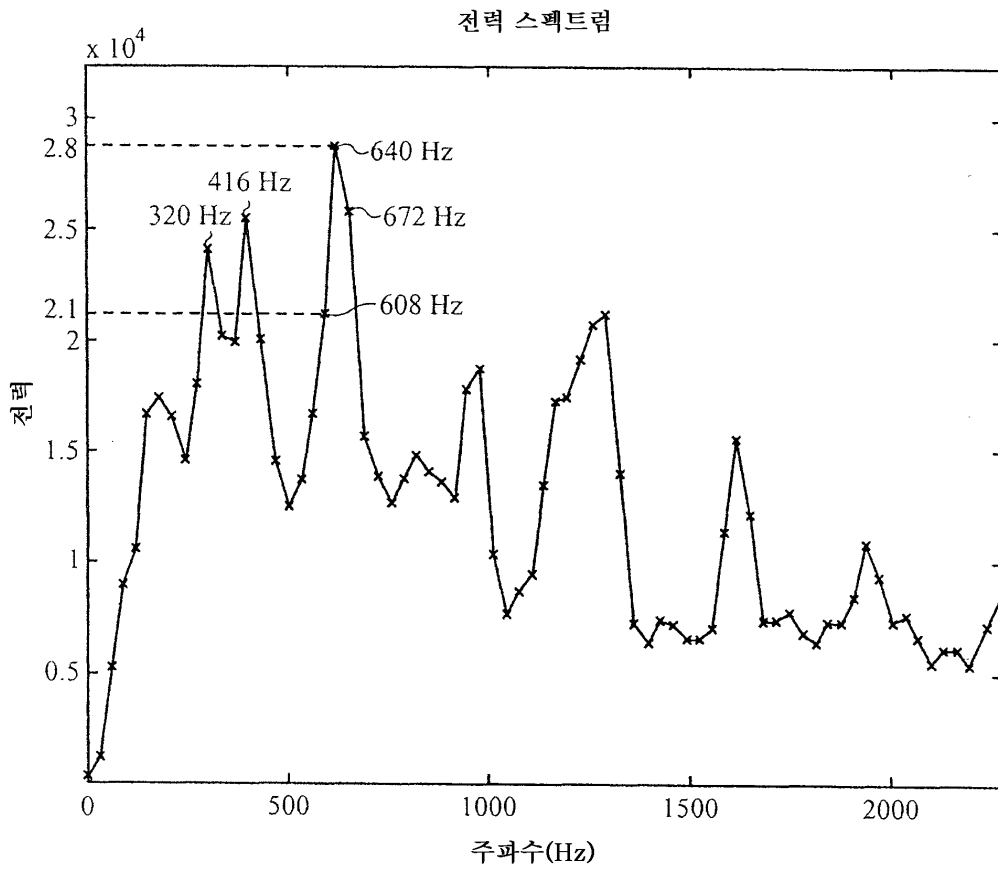
도면5



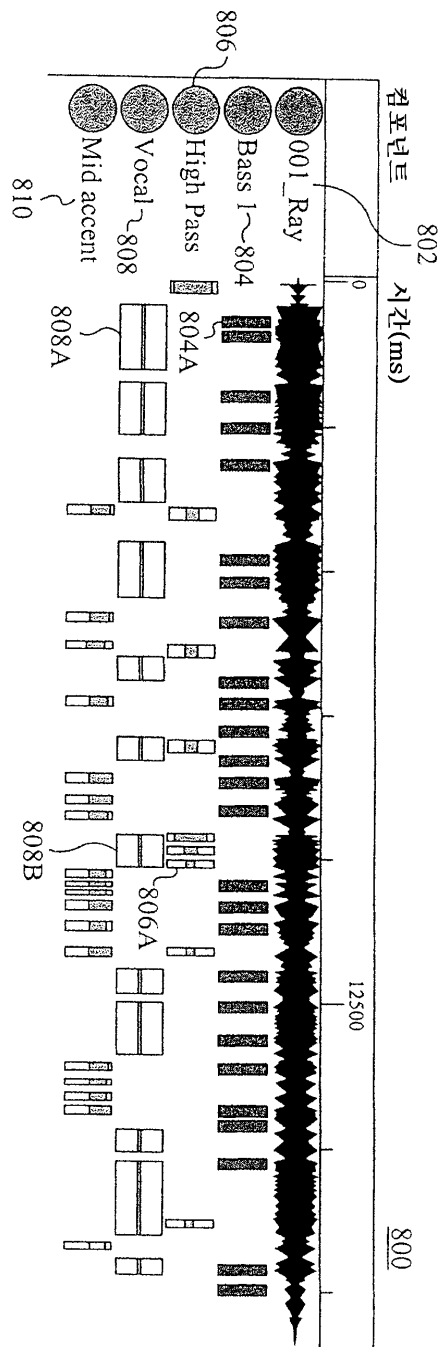
도면6



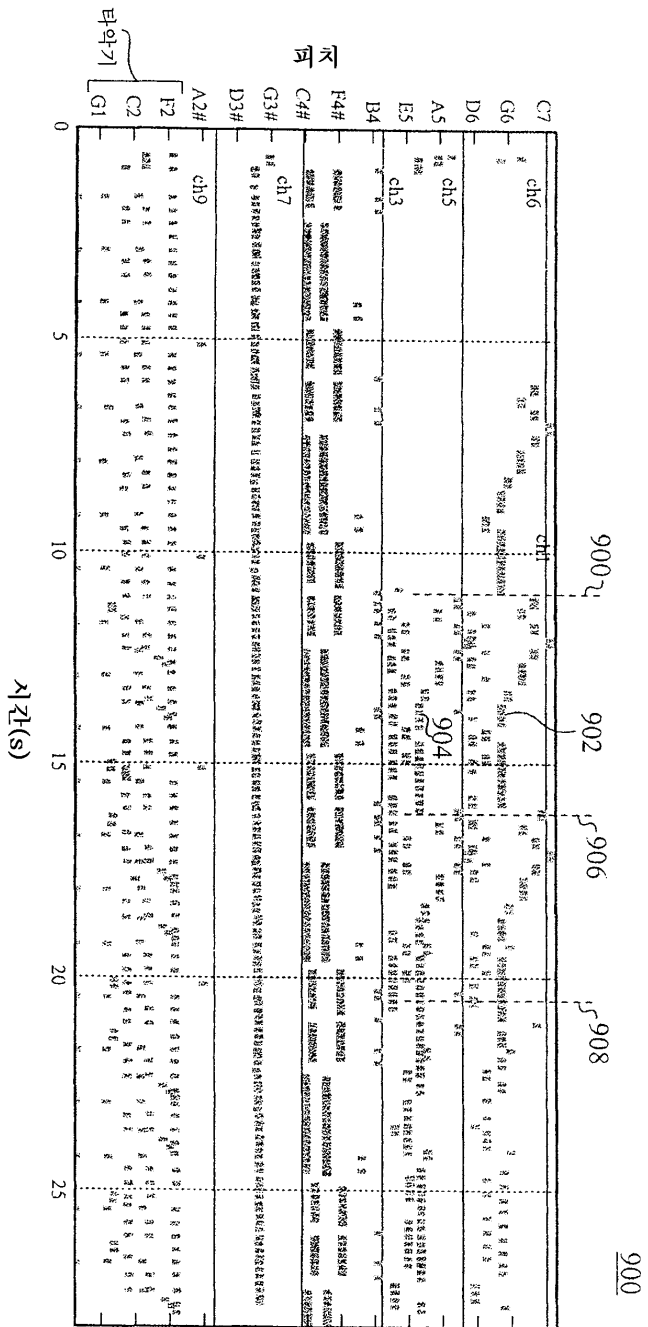
도면7



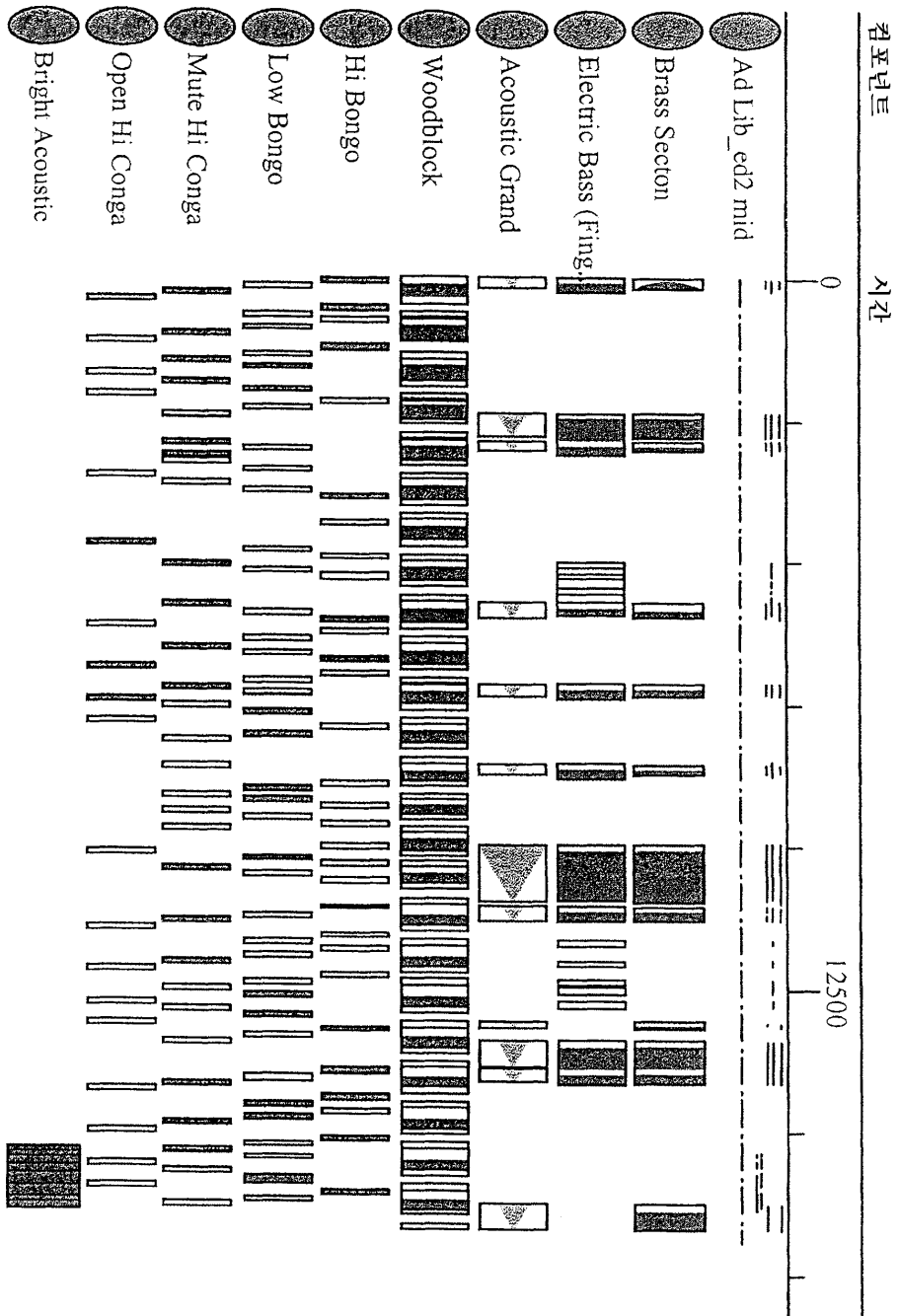
도면8



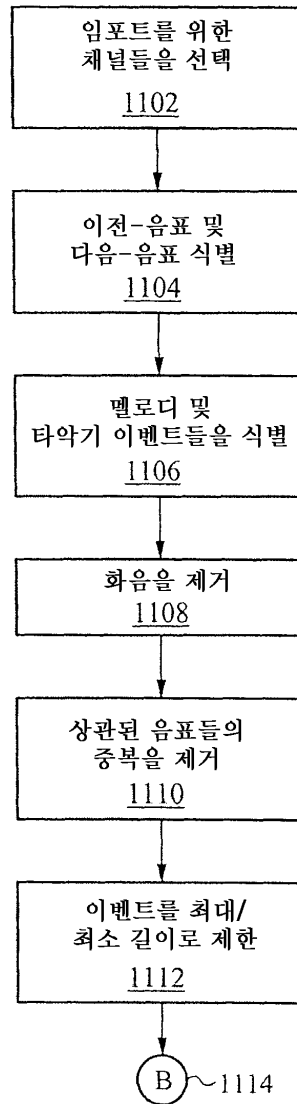
도면9



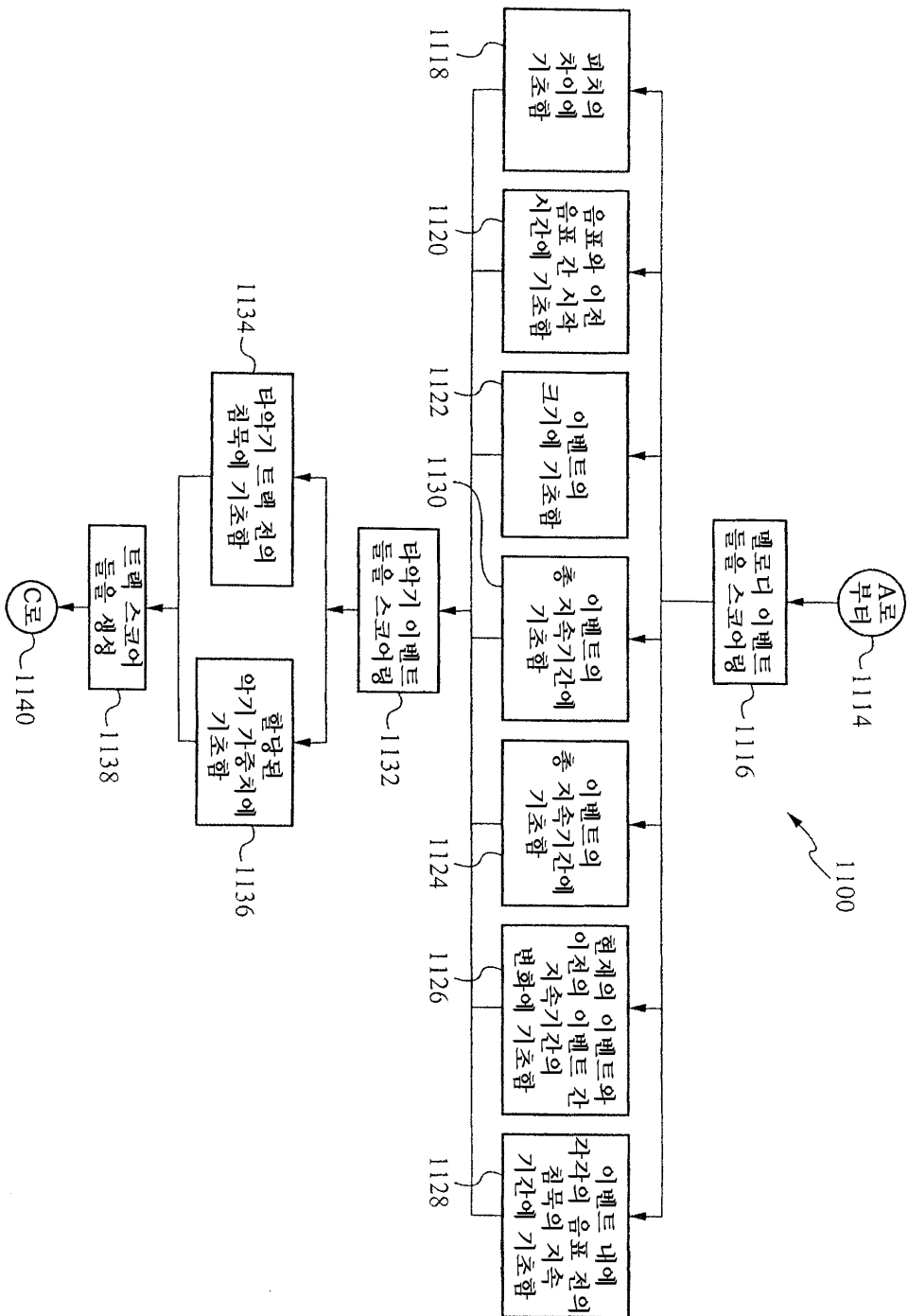
도면10



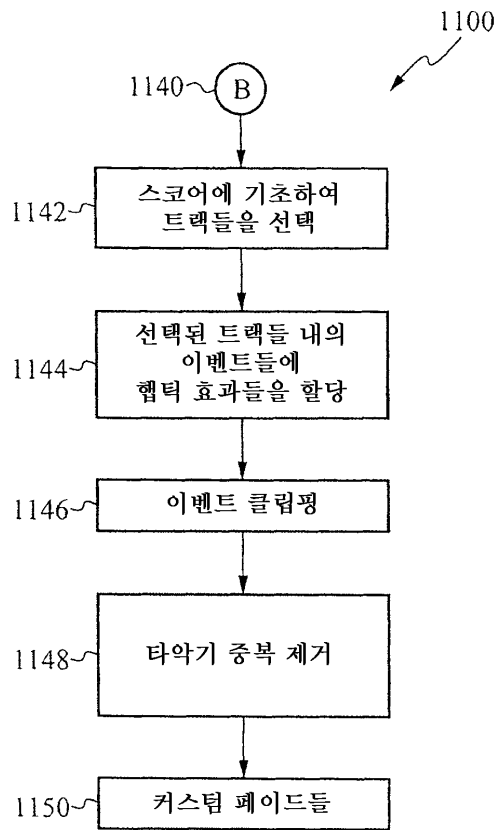
도면11a



도면11b



도면11c



도면12

