



(19)대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) 。 Int. Cl. (11) 공개번호 10-2006-0132607  
G06F 17/30 (2006.01) (43) 공개일자 2006년12월21일

(21) 출원번호	10-2006-7011219	(87) 국제공개번호	WO 2005/057429
(22) 출원일자	2006년06월08일	(88) 국제공개일자	2005년06월23일
심사청구일자	없음		
번역문 제출일자	2006년06월08일		
(86) 국제출원번호	PCT/IB2004/052499		
국제출원일자	2004년11월22일		

(30) 우선권주장 03104572.7 2003년12월08일 유럽특허청(EPO)(EP)

(71) 출원인 코닌클리케 필립스 일렉트로닉스 엔.브이.  
네덜란드왕국, 아인트호펜, 그로네보르스베그 1

(72) 발명자 파우스, 스테펜, 씨.  
네덜란드, 아아 아인트호펜 엔엘-5656, 홀스트란 6 내

(74) 대리인 정상구  
홍동오

전체 청구항 수 : 총 12 항

(54) 멜로디 데이터베이스에서 검색하는 방법

(57) 요약

멜로디 데이터베이스(114)에서, 오디오 프래그먼트를 나타내는 질의 스트링을 검색하는 시스템은 사용자로부터 질의 스트링을 수신하는 입력(122, 132)을 포함한다. 멜로디 데이터베이스(114)는 복수의 오디오 프래그먼트들의 각각의 표현들을 저장한다. 처리기(116)는 질의 스트링을 복수의 질의 서브-스트링들의 시퀀스로 분해한다(117). 각각의 서브-스트링은, 서브-스트링에 대한 적어도 각각의 가장 근접한 매칭이 데이터베이스에서 독립적으로 검색된다(118). 각각의 서브-스트링들에 대한 검색 결과들에 따라, 질의 스트링에 대한 적어도 가장 근접한 매칭이 결정된다(119).

대표도

도 1

특허청구의 범위

청구항 1.

멜로디 데이터베이스에서, 오디오 프래그먼트(audio fragment)를 나타내는 질의 스트링(query string)에 대한 매칭을 검색하는 방법에 있어서:

상기 질의 스트링을 복수의 질의 서브-스트링들의 시퀀스로 분해하는 단계;

각각의 서브-스트링에 대해, 상기 서브-스트링에 대한 적어도 각각의 가장 근접한 매칭을 상기 데이터베이스에서 독립적으로 검색하는 단계; 및

상기 각각의 서브-스트링들에 대한 검색 결과들에 따라, 상기 질의 스트링에 대한 적어도 가장 근접한 매칭을 결정하는 단계를 포함하는, 매칭 검색 방법.

## 청구항 2.

제 1 항에 있어서,

상기 질의 스트링을 분해하는 상기 단계는, 각각이 구문(phrase)에 실질적으로 대응하는 서브-스트링들로 상기 질의 스트링을 분해하는 단계를 포함하는, 매칭 검색 방법.

## 청구항 3.

제 1 항에 있어서,

사용자로 하여금 복수의 질의 입력 양식들(query input modalities)을 혼합한 상기 질의 스트링을 입력할 수 있게 하는 단계를 포함하는, 매칭 검색 방법.

## 청구항 4.

제 3 항에 있어서,

상기 질의 입력 양식들 중 적어도 하나는 허밍, 노래, 휘파람, 탭핑, 클랩핑, 타악 보컬음들(percussive vocal sounds) 중 하나인, 매칭 검색 방법.

## 청구항 5.

제 3 항에 있어서,

질의 입력 양식의 변경은 서브-스트링 경계와 실질적으로 일치하는, 매칭 검색 방법.

## 청구항 6.

제 1 항에 있어서, 상기 질의 스트링을 분해하는 상기 단계는:

얼마나 많은 ( $N_s$ ) 서브-스트링들이 상기 질의 스트링에 존재하는지를 추정하는 단계;

상기 질의 스트링을  $N_s$ 개의 순차적 서브-스트링들로 분할하는 단계로서, 각각의 서브-스트링은 상기 서브-스트링을 나타내는 각각의 센트로이드(centroid)와 연관되는, 상기 질의 스트링 분할 단계; 및

미리 결정된 수렴 기준(convergence criterion)이 충족될 때까지 반복적으로, 각각의 센트로이드에 대해, 상기 대응하는 서브-스트링에 따라 각각의 센트로이드 값을 결정하는 단계, 및 상기 서브-스트링 각각에 대해, 상기 센트로이드들 각각과 그 대응하는 서브-스트링 사이의 총 거리 측정을 최소화함으로써, 대응하는 서브-스트링 경계들을 결정하는 단계를 포함하는, 매칭 검색 방법.

### 청구항 7.

제 2 항 또는 제 6 항에 있어서,

얼마나 많은 ( $N_s$ ) 서브-스트링들이 상기 질의 스트링에 존재하는지를 추정하는 상기 단계는, 상기 오디오 프래그먼트의 지속시간을 구문의 평균 지속시간으로 나누는 단계를 포함하는, 매칭 검색 방법.

### 청구항 8.

제 5 항에 있어서,

상기 질의 스트링을 분해하는 상기 단계는, 상기 입력 양식들 각각에 대해 각각의 분류 기준을 검색하고, 상기 분류 기준들에 기초하여 질의 입력 양식의 변경을 검출하기 위한 분류 알고리즘을 사용하는 단계를 포함하는, 매칭 검색 방법.

### 청구항 9.

제 3 항 또는 제 8 항에 있어서,

서브-스트링이 질의 입력 양식의 2개의 연속하는 변경들 내에 있도록 하는 단계를 포함하는, 매칭 검색 방법.

### 청구항 10.

제 1 항에 있어서,

상기 데이터베이스에서 각각의 서브-스트링을 검색하는 상기 단계는, 상기 서브-스트링에 대해, 대응하는 유사 측정과 상기 데이터베이스에서 N개의 가장 근접한 대응하는 부분들의 N-최상 목록( $N \geq 2$ )을 발생시키고; 상기 서브-스트링들의 상기 N-최상 목록들의 상기 유사 측정들에 기초하여 상기 질의 스트링에 대한 상기 적어도 가장 근접한 매칭의 결정을 수행하는 단계를 포함하는, 매칭 검색 방법.

### 청구항 11.

처리기로 하여금 제 1 항에 청구된 방법의 단계들을 실행하게 하도록 동작 가능한 컴퓨터 프로그램 제품.

### 청구항 12.

멜로디 데이터베이스에서 오디오 프래그먼트를 나타내는 질의 스트링을 검색하는 시스템에 있어서:

사용자로부터 상기 질의 스트링을 수신하는 입력(122, 132);

복수의 오디오 프래그먼트들의 각각의 표현들을 저장하는 멜로디 데이터베이스(114); 및

프로그램의 제어하에,

- 상기 질의 스트링을 복수의 질의 서브-스트링들의 시퀀스로 분해하고(117);
- 각각의 서브-스트링에 대해, 상기 서브-스트링에 대한 적어도 각각의 가장 근접한 매칭을 상기 데이터베이스에서 독립적으로 검색하고(118);
- 상기 각각의 서브-스트링들에 대한 검색 결과들에 따라, 상기 질의 스트링에 대한 적어도 가장 근접한 매칭을 결정하기(119) 위한 적어도 하나의 처리기(116)를 포함하는, 검색 시스템.

## 명세서

### 기술분야

본 발명은 멜로디 데이터베이스에서 오디오 프래그먼트(audio fragment)를 나타내는 질의 스트링(query string)에 대해 검색하는 방법에 관한 것이다. 또한, 본 발명은 멜로디 데이터베이스에서 오디오 프래그먼트를 나타내는 질의 스트링을 검색하는 시스템과 이러한 시스템에 사용하기 위한 서버에 관한 것이다.

### 배경기술

인터넷을 통한 오디오 배급의 증가로 인해, 특정한 오디오 트랙/타이틀의 검색도 더욱 중요하게 되었다. 통상적으로 사용자는 예술가 이름, 작곡가, 레코드 회사 등과 같은 메타데이터 상의 오디오 타이틀들/트랙들을 검색할 수 있다. 오디오 트랙들의 매칭에 대하여 데이터베이스를 통해 검색이 수행되었다. 사용자는 재생/다운로드를 위해, 히트들 중 하나, 가능하면 여러 개를 선택할 수 있었다. 사용자가 항상 임의의 적절한 메타데이터를 지정할 수 있는 것은 아니기 때문에, 다른 형태들의 질의 스트링 지정도 또한 이용 가능하게 되었다. US 5,963,957호에는 소위 '허밍에 의한 질의' 방식이 개시되어 있다. 사용자는 간단히 오디오 트랙의 일부를 허밍할 수 있다. 사용자에게 의해 허밍된 오디오 프래그먼트는 질의 스트링으로 변환된다(예를 들면, 허밍된 프래그먼트를 톤들(tones) 또는 톤의 차들의 시퀀스로 변환함으로써). 그 다음, 데이터베이스는 트랙들(또는, 허밍된 프래그먼트를 포함하는 일반적으로 더 긴 오디오 프래그먼트들)을 매칭하기 위해 검색된다. 매칭은 거리 측정(distance measure)에 기초한다. 통계 기준이 사용될 수 있다. 노래, 휘파람 또는 탭핑과 같은 다른 오디오 입력 양식들(audio input modalities)도 또한 알려져 있다.

### 발명의 상세한 설명

본 발명의 목적은 데이터베이스에서 오디오 프래그먼트를 찾는데 있어서 증대된 정확도를 제공하는 상술한 종류의 개선된 방법, 시스템 및 서버를 제공하는 것이다.

본 발명의 목적을 충족시키기 위하여, 멜로디 데이터베이스에서 오디오 프래그먼트를 나타내는 질의 스트링에 대한 매칭을 검색하는 방법은:

상기 질의 스트링을 복수의 질의 서브-스트링들의 시퀀스로 분해하는 단계;

각각의 서브-스트링에 대해, 상기 서브-스트링에 대한 적어도 각각의 가장 근접한 매칭을 상기 데이터베이스에서 독립적으로 검색하는 단계; 및

상기 각각의 서브-스트링들에 대한 검색 결과들에 따라, 상기 질의 스트링에 대한 적어도 가장 근접한 매칭을 결정하는 단계를 포함한다.

본 발명자는 사실상, 사용자에게 의해 오디오 입력을 나타내는 질의 스트링이 데이터베이스에서 나타난 더 긴 오디오 프래그먼트의 하나의 일관하는 순차적인 부분이 실제로 될 수 없음을 알았다. 예를 들면, 사용자는 2개의 구문들을 가진 오디오 프래그먼트를 나타내는 질의 스트링을 제공할 수 있다: 사용자는 주요 가사들(main lyrics)의 구문을 노래하고 코러스의 구문이 후속되고, 첫 번째 구문과 코러스 구문 사이에 놓인 구문들을 건너뛰으로써 시작했다. 사용자가 구문들 중 하나만 제공하게 하여, '완전한(perfect)' 매칭이 데이터베이스에서 발견될 수 있었다. 종래의 검색 방법은 데이터베이스에 대한 2

가지 구문들의 전체 시퀀스를 매칭하도록 시도한다. 많은 경우에, 이것은 매우 근접한 매칭을 제공하지 않으며(어느 것이 라도 적어도 신뢰할 수 있게 검출될 수 있다면), 시스템의 정확도를 적어도 감소시킬 것이다. 본 발명에 따라서, 질의 스트링은 복수의 질의 서브-스트링들의 시퀀스로 분해된다. 서브-스트링들은 데이터베이스 내에 저장된 오디오 표현들에 대해 독립적으로 매칭된다. 개개의 매칭 동작들의 결과는 전체 질의 스트링에 대한 매칭을 결정하는데 사용된다. 사용자가 2 개의 비순차적인 구문들을 질의 스트링으로서 제공한 예에서, 두 구문들은 훨씬 더 신뢰할 수 있게 위치될 수 있다. 둘 모두 동일한 오디오 트랙에 대해 양호한 매칭을 보인다면, 그 트랙은 전체 질의에 대한 매칭으로서 매우 신뢰할 수 있게 식별 될 수 있다.

최근, 오디오를 저장할 수 있는 고용량의 로컬 시스템들이 더욱 인기 있게 되었다. 이러한 시스템들은 오디오 주크-박스, 내장된 튜너 및 하드디스크를 가진 셋-탑 박스, 하드디스크 레코더 등을 구비한 PC와 같은 임의의 형태를 취할 수 있다. 또한, Apple iPod 및 Philips HDD100과 같은 휴대용 고용량 오디오 저장 시스템들이 이용 가능하게 되었다. 이들 로컬 저장 시스템은 수천 개의 오디오 트랙들을 쉽게 저장할 수 있다. 통상적으로, 이러한 시스템들은 예술가, 타이틀, 앨범 등과 같은 하나 이상의 메타데이터 항목들을 지정함으로써 사용자가 특정한 트랙을 검색할 수 있게 한다. 본 발명에 따른 방법은 또한, 특히 사용자가 관련 메타데이터를 잊어버린 경우에도, 이러한 시스템에서 오디오 트랙을 신속히 선택하기 위해 이용될 수 있다.

중속 청구항 제 2 항의 방법에 따라, 분해는 질의를 서브-스트링들로 나누며, 서브-스트링들 각각은 구문에 대응한다. 구문 경계는 임의의 적절한 방식으로 검출될 수 있으며, 예를 들면, 구문은 일반적으로 8 내지 20 노트(notes)의 길이로서 중심 톤에 따라 정해진다. 구문들 사이에, 숨을 쉴 수 있게 하기 위해 잠깐 멈춤(pause)들이 일어날 수 있고, 중심 톤이 변경 될 수 있다. 구문들은 종종 허밍의 감속에 의해 종료된다. 또는, 구문들은 큰 톤 차들(즉, 간격들)과 큰 톤 지속기간들에 의해 구별된다. 질의 스트링에 나타난 순차적 구문들을 개별적으로 인식함으로써, 정확도가 증가한다.

중속 청구항 제 3 항의 방법에 따라, 사용자는 질의 스트링은 상이한 입력 양식들을 사용하여 입력된 복수의 오디오 부분 들의 혼합인 오디오 프래그먼트를 나타내는 질의 스트링을 제공할 수 있다. 종래의 멜로디 데이터베이스들은 한 형태의 입력 양식만 지원한다. 그래서, 사용자는 데이터베이스의 입력 형태를 사용해야 했다. 본 발명에 따라, 데이터베이스는 다수의 양식들을 사용하여 입력한 오디오 프래그먼트들에 대해 검색될 수 있었다. 중속 청구항 제 4 항의 방법에 따라, 질의 입력 양식들 중 적어도 하나는 허밍, 노래, 휘파람, 탭핑, 클랩핑, 타악 보컬음들(percussive vocal sounds) 중 하나이다. 대체로, 데이터베이스가 그 양식 형태를 지원하는 한, 임의의 적절한 양식이 사용될 수 있다.

중속 청구항 제 5 항의 방법에 따라, 입력 양식의 변경이 검출될 때마다, 새로운 서브-스트링이 시작된다. 상술한 바와 같이, 종래의 멜로디 데이터베이스들은 전체 질의 스트링에 대해서만 검색될 수 있다. 발명자는 사용자들이 질의 스트링에 의해 나타내는 오디오 프래그먼트의 입력 동안, 입력 양식을 변경할 수 있음을 알았다. 예를 들면, 사용자가 코러스의 구문을 노래할 수 있고, 주요 가사의 구문을 허밍할 수 있다. 질의 스트링을 나눔으로써, 상이한 입력 양식들에 대응하는 부분 들은, 예를 들면, 각각의 입력 양식들에 최적화된 데이터베이스들을 이용하거나 또는 각각의 양식에 개별적으로 데이터베이스에서 동일한 구문을 나타냄으로써, 개별적으로 검색될 수 있다.

중속 청구항 제 6 항의 방법에 따라, 서브-스트링들의 위치 및 크기를 최적화하는 반복적인 자동 처리가 사용된다. 이러한 방식으로, 자동적으로 분해가 발견될 수 있다. 초기의 추정치는 서브-스트링들의 수로 이루어질 수 있다. 각각의 서브-스트링은 각각의 센트로이드(centroid)(서브-스트링의 오디오 특성들과 함께)에 의해 나타낸다. 따라서 초기 추정치는 센트로이드들의 초기 수를 결정한다. 센트로이드들의 초기 위치들은 오디오 프래그먼트를 따라 등거리로 분배되어 선택될 수 있다. 서브-스트링들은 초기에 같은 크기일 수 있다. 그 후에 절차는, 서브-스트링과 그 센트로이드 사이의 거리를 최소화한다. 한 입력 양식으로부터 다른 입력 양식으로서의 점프는 일반적으로 거리에 악영향을 미칠 것이다. 그래서, 서브-스트링이 초기에 오디오 프래그먼트 내에서 2개의 연속하는 입력 양식들을 오버랩핑시키면, 상기 최소화는 서브-스트링의 경계가 그 센트로이드와 동일한 입력 양식 내에 주로 있을 때까지 서브-스트링의 경계를 이동시키려 한다. 유사하게, 다음 서브-스트링의 경계가 이동될 것이다.

중속 청구항 제 7 항의 방법에 따라, 서브-스트링들의 수(이에 따른 센트로이드들의 수)의 초기 추정은 구문의 평균 지속 기간에 비한 오디오 프래그먼트의 지속기간에 기초한다. 예를 들어, 40개의 톤들을 가진 오디오 프래그먼트는 최대 5개 구분들(8개 톤들의 최소 구분 길이에 기초하여)을 포함하는 것으로 가정할 수 있다. 그래서, 오디오 프래그먼트를 따라 등거리로 분배된 5개의 센트로이드들로 반복이 시작할 수 있다. 이러한 수의 센트로이드들은 최대 수의 센트로이드들로서 사용되는 것이 바람직하다. 프래그먼트가 매우 일관된 상황(예를 들면, 사용자가 정확한 시퀀스의 구문들을 노래함)을 포함 하기 위해 더 소수의 센트로이드들에 대해서도 동일한 최적화가 또한 수행될 수 있다.

종속 청구항 제 8 항의 방법에 따라, 질의 시퀀스를 더욱 일치된 서브-스트링들로 암시적으로 분리하는 자동 최소화 절차(거리 측정이 암시적인 분류 기준으로서 작용함)를 사용하는 대신에, 또는 그에 더하여, 또한 명시적 분류 기준이 분리를 위해 사용될 수 있다. 동일한 서브-스트링에 할당되는 질의 스트링의 각 부분은 동일한 미리 결정된 분류 기준을 충족시키고 각각의 2개의 순차적 서브-스트링들은 상이한 미리 결정된 분류 기준을 충족시킨다. 상이한 분류 기준은 각각의 입력 양식들의 오디오 특성들을 나타낸다. 예를 들면, 노래 및 허밍과 같은 어떤 입력 양식들은 분명한 피치(clear pitch)를 가지고, 반면, 타악기 모조품들(percussion-imitations)과 같은 다른 입력 양식들은 분명한 피치를 가지지 않는다(즉, 시끄럽다). 일부 특성들이 모든 사용자들에게 적용된다는 점에서 절대적일 수 있지만, 어떤 특성들은 상대적일 수 있고(예를 들면, 노래/허밍 피치에 대한 휘파람의 피치 레벨), 전체 오디오 프래그먼트를 분석한 후에, 또는 사용자에 의해 처음 훈련한 후에만 설정될 수 있다는 것이 인식될 것이다.

종속 청구항 제 9 항의 방법에 따라, 분류는 입력 양식의 변경을 나타내는 입력 질의 스트링에서 경계들의 검출을 유발한다. 그 다음, 검출된 경계(또는 경계들)는 서브-스트링이 2개의 연속하는 경계들 사이에 있어야 하는(즉, 서브-스트링은 경계를 오버랩할 수 없다) 자동 분리에 대한 제약으로서 사용된다. 하나 이상의 서브-스트링(예를 들면, 2개의 노래 구문들)이 2개의 경계들 사이에 위치될 수 있음을 알 것이다. 여기서, 오디오 프래그먼트의 시작과 끝도 또한 경계들로 계산한다.

종속 청구항 제 10 항의 방법에 따라, 데이터베이스 내에서 서브-스트링들 각각을 검색하는 단계는 각각의 서브-스트링에 대해, 대응하는 유사 측정과 데이터베이스에서 N개의 가장 근접한 대응하는 부분들의 N-최상 목록( $N \geq 2$ )을 제공한다. 얻어진 N-최상 목록들에 기초하여, 전체 질의 스트링에 대한 최적의 매칭이 결정된다(또는 전체 질의 스트링에 대한 N-최상 목록이 생성된다).

본 발명의 목적을 충족시키기 위하여, 멜로디 데이터베이스에서 오디오 프래그먼트를 나타내는 질의 스트링을 검색하는 시스템은:

사용자로부터 상기 질의 스트링을 수신하는 입력;

복수의 오디오 프래그먼트들의 각각의 표현들을 저장하는 멜로디 데이터베이스;

프로그램의 제어하에,

상기 질의 스트링을 복수의 질의 서브-스트링들의 시퀀스로 분해하고;

각각의 서브-스트링에 대해, 상기 서브-스트링에 대한 적어도 각각의 가장 근접한 매칭을 상기 데이터베이스에서 독립적으로 검색하고; 및

상기 각각의 서브-스트링들에 대한 검색 결과들에 따라, 상기 질의 스트링에 대한 적어도 가장 근접한 매칭을 결정하기 위한 적어도 하나의 처리기를 포함한다.

본 발명의 이들 및 다른 양태들은 이후에 기술된 실시예들을 참조하여 비제한적인 예의 방식으로 더욱 명백해질 것이다.

## 실시예

본 발명에 따라, 질의 스트링은 서브-스트링들로 나누어지며, 서브-스트링들은 데이터베이스에서 개별적으로 검색되고, 결과들에 기초하여 매칭이 결정된다. 서브-분할은 입력 양식의 변경들을 반영하는 것이 바람직하다. 이러한 서브-분할은 여러 방식으로 달성될 수 있다. 하기에, 동적 프로그래밍을 사용한 최소화 알고리즘이 기술되고, 분류 방식이 기술된다. 또한, 예를 들면 분류가 최소화를 위한 사전 분석으로 사용되는 경우에는, 조합된 방법들이 이용될 수 있다. 입력 양식의 변경을 위한 서브-분할을 수행하기 위한 대안으로서, 서브-분할은 구문의 변경에 기초할 수 있다. 임의의 적절한 구문 검출 알고리즘이 사용될 수 있다. 바람직하게, 구문들 및 입력 양식의 변경을 위한 서브-분할이 조합된다. 예를 들면, 먼저 입력 양식의 변경이 발생될 때마다 서브-스트링들을 생성하기 위해 서브-분할이 행해진다. 이들 서브-스트링들은 구문의 변경이 검출될 때마다 더 서브-분할된다.

도 1은 본 발명에 따른 방법이 이용될 수 있는 예시적 시스템(100)의 블록도fmf 도시한다. 이 시스템(100)에서, 서버(110)와 클라이언트(2개의 클라이언트들(120 및 130)이 도시됨)를 통해 기능이 분배된다. 서버(110) 및 클라이언트들(120/

130)은 네트워크(140)를 통해 통신할 수 있다. 이것은 인터넷, WiFi, 블루투스, IEEE 1394 등과 같은 근거리 네트워크가 될 수 있다. 네트워크(140)는 인터넷과 같이 광역 네트워크(wide area network)인 것이 바람직하다. 네트워크(140)를 통해 통신하기 위해 장치들은 적절한 하드웨어/소프트웨어(서버(110)에서 항목(112)으로서, 클라이언트들에서 각각의 항목들(126, 136)로서 도시됨)를 포함한다. 이러한 통신 하드웨어/소프트웨어는 알려져 있고 더 이상 기재하지 않을 것이다.

본 발명에 따른 시스템에서, 사용자는 오디오 프래그먼트를 나타내는 질의 스트링을 직접 또는 간접적으로 지정한다. 도 1의 기능의 서브분할을 사용하여, 사용자는 각각의 사용자 인터페이스(122, 132)를 통해 클라이언트들(120 또는 130) 중 하나를 사용하여 질의 스트링을 지정한다. 클라이언트는 PC와 같은 종래의 컴퓨터, 또는 PDA와 같은 유사 컴퓨터 장치 상에서 구현될 수 있다. 특히, 클라이언트는, 사용자가 라이브러리로부터 재생되거나 라이브러리에 다운로드된 오디오 트랙을 지정할 수 있도록 하기 위하여, 음악 라이브러리(Real One, 윈도우즈 미디어 플레이어, 애플 iTunes 등으로부터 알려진 라이브러리들과 유사함)를 포함하는 장치 상에서 구현될 수 있다. 마우스, 키보드, 마이크로폰 등과 같은 임의의 적절한 사용자 인터페이스가 사용될 수 있다. 특히, 사용자는 음성 입력과 같은 오디오 또는 유사 오디오 입력을 사용하여 오디오 프래그먼트를 지정할 수 있다. 예를 들면, 사용자는 오디오 프래그먼트를 노래, 허밍, 휘파람, 탭핑할 수 있다. 오디오 프래그먼트는 마이크로폰을 통하여 클라이언트에 의해 수신될 수 있다. 마이크로폰은 종래의 아날로그 마이크로폰이 될 수 있으며, 이 경우, 클라이언트는 보통 PC의 오디오 카드 상에 존재하는 A/D 변환기를 포함할 수 있다. 마이크로폰은 또한 A/D 변환기를 이미 포함하는 디지털 마이크로폰이 될 수 있다. 이러한 디지털 마이크로폰은, 예를 들면 USB, 블루투스 등을 사용하여 임의의 적절한 형태로 클라이언트(120/130)에 접속될 수 있다. 오디오 프래그먼트는 또한, 예를 들면 마우스 또는 표준 PC 텍스트 키보드를 사용하는 종래의 입력 장치들을 사용하거나, 또는 PC에 부착된 음악 키보드를 사용하여 노트들을 지정하는 바와 같은 다른 형태들로 입력될 수 있다.

클라이언트는 오디오 프래그먼트를 질의 스트링으로 변환하기 위한 어떤 형태의 전처리(preprocessing)를 수행하는 것이 바람직하다. 이러한 전처리는 적절한 프로그램의 제어하에 처리기(124/134)에 의해 수행될 수 있다. 프로그램은 하드디스크, ROM, 또는 플래시 메모리와 같은 비휘발성 메모리로부터 처리기(124/134)로 로딩된다. 전처리는, 예를 들면 MP3 압축을 사용하여 오디오 프래그먼트를 압축하는 것으로 제한될 수 있다. 오디오 프래그먼트가 Midi 포맷과 같은 적절하게 압축된 형태로 이미 존재한다면, 클라이언트(120/130)에서 다른 전처리는 요구되지 않을 수 있다. 전처리는 또한 멜로디 데이터베이스(114)를 통한 검색에 적절한 포맷으로의 변환을 포함할 수 있다. 사실상, 데이터베이스에서 오디오 프래그먼트의 실제 오디오 콘텐츠를 나타내기 위한 임의의 적절한 방법이 사용될 수 있다. 이렇게 하기 위한 다양한 방법들은, 선택적으로 노트 지속기간을 가진 톤들의 시퀀스로서 프래그먼트를 기술하는 것과 같이, 이에 대해 알려져 있다. 또한, 절대 톤 시퀀스가 주어지지 않지만 톤 값들의 변경들만이 주어지는(톤 증가, 동일 톤, 톤 감소) 경우에 형태들이 알려져 있다. 원한다면, 멜로디 데이터베이스는 또한 오디오 프래그먼트들의 스펙트럼 정보를 포함할 수 있다. 오디오 처리 분야, 특히, 다른 분석을 위한, 특히 매칭을 위해 데이터베이스를 통해 검색하기 위한 적절한 형태로 오디오 및/또는 음성 입력을 나타내기 위한 음성 처리 분야로부터의 기술들이 일반적으로 알려져 있다. 예를 들면, 피치 검출 기술들(pitch detection techniques)이 일반적으로 알려져 있고, 톤 값들 및 톤 지속기간들을 확립하기 위해 사용될 수 있다. 이러한 기술들은 본 발명의 부분이 아니다.

본 발명에 따른 시스템에 대해, 데이터베이스(114)에 액세스하기 위한 질의 스트링을 지정하기 위한 임의의 적절한 형태는, 데이터베이스(114)가 질의 스트링 포맷들을 지원하는 한, 사용될 수 있다. 데이터베이스는 질의의 매칭을 위해 데이터베이스의 기록들을 검색하도록 동작된다. 이러한 질의들을 지원하는 멜로디 데이터베이스들이 알려져 있다. 매칭은 '완전한(full)' 매칭이 되어야 할 필요가 없지만, '통계적(statistical)' 매칭이 되어야 하는 것이 바람직하며, 즉 데이터베이스에서 하나 이상의 기록들이 질의와 유사한 필드와 동일시된다. 유사성은, 예를 들면 질의 항목과 데이터베이스의 대응하는 필드 사이의 거리 측정에 기초한 통계적 가능성(statistical likelihood)일 수 있다. 데이터베이스는 매칭의 더욱 신속한 검색을 가능하게 하기 위해 인덱싱되는 것이 바람직하다. 대리인 문서 번호 제PHNL030182호의 미공개 특허 출원은 정확하지 않은 매칭들을 지원하는 데이터베이스를 인덱싱하는 방법을 개시한다. 식별된 기록을 위한 데이터베이스가 시스템의 사용자에게 유용할 수 있는 정보를 저장하는 것이 이해될 것이다. 이러한 정보는, 작곡가, 공연 예술가, 레코드 회사, 레코드 연도, 스튜디오 등과 같은 식별된 프래그먼트에 대한 도서 목록 정보를 포함할 수 있다. 데이터베이스를 통한 검색은 하나 이상의 '매칭(matching)' 기록들(바람직하게는, 예를 들면, 데이터베이스에서 10개의 가장 가능성 있는 히트들을 가진 N-최상 목록의 형태)을 식별할 수 있고, 저장된 도서 목록 데이터의 일부 또는 전부와 함께 이들 기록들을 제공할 수 있다. 도 1의 장치에서, 정보는 네트워크를 통해 서버로부터 질의를 지정하는 클라이언트로 공급된다. 클라이언트의 사용자 인터페이스는 정보를 사용자에게 제공(예를 들면, 디스플레이 또는 음성-합성을 이용)하기 위해 사용되거나, 또는 식별된 오디오 트랙 또는 인터넷 서버로부터 전체 앨범을 다운로드하는 것과 같은 다른 자동 동작을 수행하기 위해 사용된다. 데이터베이스가 검색의 견고성(robustness)을 증가시키기 위하여, 구문 또는 심지어 구문의 절반과 같은 더 작은 프래그먼트들을 검색할 수 있는 것이 바람직하다.

본 발명에 따라 질의 스트링은 복수의 질의 서브-스트링들의 시퀀스로 분해된다. 각각의 서브-스트링에 대해, 데이터베이스는 서브-스트링에 대한 각각의 가장 근접한 매칭을 위해 독립적으로 검색된다. 상술한 바와 같이, 이것은 바람직하게, 대응하는 유사 측정과 데이터베이스에서 N개의 가장 근접한 대응하는 부분들의 N-최상 목록( $N \geq 22$ )을 유발한다. 유사 측정은 거리 또는 가능성일 수 있다. 적절한 거리 측정들/가능성들은 당업자에게 알려져 있고, 더 이상 기술되지 않는다. 각각의 서브-스트링들에 대한 검색 결과들에 따라, 시스템은 전체 질의 스트링에 대한 적어도 가장 근접한 매칭을 결정한다. 바람직하게는, 시스템은 전체 스트링에 대한 N-최상 목록( $N \geq 22$ )을 생성하여 사용자가 가능성 있는 후보들의 제한된 목록으로부터 최종 선택을 하게 할 수 있다. 데이터베이스가 서브-스트링들에 대한 N-최상 목록들을 공급할 수 있는 시스템들에 대해, 전체 질의 스트링에 대한 매칭은 서브-스트링들의 N-최상 목록들의 유사 측정들에 기초하는 것이 바람직하다. 서브-매칭들에 대한 결과들로부터 전체 매칭에 대한 결과가 예를 들면, 서브-스트링들에 대한 N-최상 목록들을 하나의 N-최상 목록으로 병합함으로써 생성될 수 있는 방법이 잘 알려져 있다. 이것은 정규화된 거리들에 대한 목록들에서 모든 항목들을 서브-스트링에 대해 순서화함으로써 행해질 수 있다. 대안적으로, N-최상 목록들에서의 등가의 항목들의 평균 정규화된 거리들이 계산될 수 있다. 서브-스트링들이 상이한 길이들을 갖기 때문에 정규화가 요구된다. 문자가 모든 멜로디들의 순서를 나타내도록 하기 위해, 항목이 각각의 N-최상 목록에 발생함을 상기한다. 이것은 항목들을 순서화하기 위하여 사용될 수 있다. 두 경우, 상부 항목은 주어진 분해에 대한 최상의 후보를 나타낸다.

도 1은, 서버(110)의 처리기(116)가 질의 스트링을 분해(117)하고, 각각의 서브-스트링을 매칭하기 위해 데이터베이스를 검색(118)하고, 서브-스트링에 대한 매칭들에 기초하여 결과를 결정(119)하는 본 발명에 따른 방법을 수행하기 위해 사용되는 것을 도시한다. 서버는 인터넷 서버들로부터 알려진 대로, 임의의 적절한 서버 플랫폼 상에서 구현될 수 있다. 처리기는, 예를 들면 인텔의 서버 처리기들과 같은 임의의 적절한 처리기일 수 있다. 프로그램은 하드디스크(도시되지 않음)와 같은 배경의 저장 장치로부터 로딩될 수 있다. 데이터베이스는, 오라클(Oracle), SQL-서버 등과 같은 임의의 적절한 데이터베이스 관리 시스템을 사용하여 구현될 수 있다.

도 2는 본 발명이 독립형 장치(200)에 이용될 수 있는 대안적인 장치를 도시한다. 이러한 장치는, 예를 들면, PC 또는 AppleIPod와 같은 오디오 플레이어라 될 수 있다. 도 2에서, 동일한 참조 번호들은 도 1에서 이미 기술된 특징들에 이용된다. 유리하게, 데이터베이스는 또한, 저장된 오디오 프래그먼트 표현들에 대해, 프래그먼트를 포함하는 오디오 타이틀에 대한 링크를 포함한다. 실제 오디오 타이틀은 데이터베이스 내에 저장될 필요는 없다. 타이틀은 장치 내에 자체로 저장되는 것이 바람직하다. 대안적으로, 그것은 네트워크를 통해 액세스 가능할 수 있다. 그러한 경우, 링크는 URL일 수 있다. 오디오 트랙 또는 오디오 앨범과 같은 실제 타이틀에 대한 매칭을 링크함으로써, 타이틀의 신속한 선택이 가능하다. 오디오 트랙의 일부를 허밍함으로써, 그 부분을 가진 트랙이 식별되고 재생이 완전히 자동으로 시작되는 것도 가능하다.

도 3은 질의 스트링을 분해하는 양호한 방법을 도시한다. 분해는 단계(310)에서 시작하여, 질의 스트링에 얼마나 많은( $N_S$ ) 서브-스트링들이 존재하는지를 추정한다. 양호한 실시예에서, 이것은 시스템을 구문 당 하나의 서브-스트링에 바이어싱(biasing)함으로써 행해진다. 이것은 질의 스트링에서 나타난 노트들의 수  $N_{notes}$  을 계산함으로써 달성된다. 구문은 통상적으로 8 내지 20 노트들로 구성되기 때문에, 구문들의 수는  $N_{notes}/8$ 과  $N_{notes}/20$  사이에 있다. 제 1 분해는  $N_S$ 로서  $N_{notes}/8$ 을 사용하여 기초할 수 있다(적절한 버림(round) 후에). 단계(320)에서, 질의 스트링은  $N_S$ 개의 순차적 서브-스트링들로 나누어진다. 적절한 초기 분할은 등거리의 분배를 사용하여 얻어진다. 이것은 도 4a에 도시되어 있다. 도 4a에서, 질의 스트링(410)은 초기에 (420, 430, 440)으로 표시된 3개의 서브-스트링들로 나누어진다. 초기에 이들 서브-스트링들은 동일한 크기이며, 즉 질의 스트링(410)에 의해 나타난 오디오 프래그먼트의 동일한 지속기간을 나타낸다. 서브-스트링들은 순차적이며, 전체 질의 스트링(410)을 함께 커버한다. 각각의 서브-스트링(420, 430, 440)은 각각의 센트로이드(425, 435, 445)에 의해 나타낸다. X로 표시된 센트로이드는 대응하는 서브-스트링의 중심에 위치한 것으로 도 4a 및 도 4b에서 볼 수 있다. 이러한 서브-스트링을 나타내는 센트로이드가 어떻게 계산되는지는 잘 알려져 있다. 예를 들면, 사용자에 의해 입력된 오디오 프래그먼트는 동일하게 규격화된 짧은 길이(즉, 20ms)의 프레임들을 사용하여 분석된다. 종래의 신호 처리는 이들 프레임들로부터 낮은 레벨의 스펙트럼 특징 벡터들(low-level spectral feature vectors)을 추출하기 위해 사용되며, 특히, 상이한 입력 양식들(즉, 노래하는 스타일들)을 구별하는데 적절하다. 이러한 특징 벡터들은 본 기술 분야에 잘 알려져 있다. 스펙트럼 계수들을 사용하여, 센트로이드는 오디오 서브-스트링 내에서 벡터들의 산술적 평균이다. 이러한 방식으로, 센트로이드들의 초기값이 얻어진다. 실제로, 모든 서브-스트링들이 동일한 크기인 것은 아니다(일반적으로 하나의 양식을 가진 구문들 및 세그먼트들의 입력은 동일한 지속기간을 가지지 않는다). 이것은 현재 서브-스트링들의 최적의 위치 및 크기를 찾는 것이 바람직하다는 것을 암시한다. 바람직하게는, 문학에서 레벨-빌딩(level-building)으로서도 또한 알려진 동적 프로그래밍은 최적조건을 찾는 데 이용된다. 동적 프로그래밍은 오디오 처리 분야 및 특히 음성 처리 분야에서 잘 알려져 있다. 센트로이드들이 주어지면, 동적 프로그래밍은 단계(330)에서 센트로이드 값들을 고정 상태로 유지하면서 서브-스트링들의 길이 및 위치를 변경하는 것을 포함할 수 있다. 이러한 방식으로, 서브-스트링들의 경



계들의 제 1 추정이 이루어진다. 이것은 각각의 센트로이드들 및 대응하는 서브-스트링 사이의 총 거리 측정을 최소화함으로써 행해진다. 당업자는 적절한 거리 측정을 선택할 수 있을 것이다. 예를 들면, 적절한 스펙트럼 계수들을 이용한 (가중된) 유클리드 거리(Euclidean distance)가 적당한 거리 측정이다. 가중은 특정한 계수들을 강조/덜 강조하기 위하여 사용될 수 있다. 도 4a의 예에서, 2개의 후속하는 부분들 사이의 주요한 중단(major break)(예를 들면, 입력 양식의 변경)은 위치(450)에 표시된다. 도 4b는 제 1 최소화 버림 후에 서브-스트링들의 경계들이 어떻게 될 수 있는지를 도시한다. 이 예에서, 서브-스트링(420)이 줄어든다(shrunk). 서브-스트링(420)의 왼쪽 경계는 질의 스트링(410)의 시작에서 고정 상태로 유지된다. 서브-스트링(430)은 약간 성장했고, 왼쪽 경계는 왼쪽으로 이동된다. 이제 센트로이드 값들이 대응하는 서브-스트링을 더 이상 정확하게 나타내지 않음을 이해할 것이다. 단계(340)에서, 센트로이드들에 대한 새로운 값들이 현재 서브-스트링 경계들에 기초하여 계산된다. 미리 결정된 수렴 기준(convergence criterion)이 충족될 때까지 처리는 되풀이하여 반복된다. 수렴 기준은 센트로이드들과 각각의 서브-스트링 사이의 거리의 합이 더 이상 감소되지 않는 것이 될 수 있다. 그 기준은 단계(350)에서 테스트된다. 선택적으로, 노트 온셋들(note onsets)은 질의 스트링에서 검출된다(예를 들면, 에너지 레벨에 기초하여). 노트 온셋들은 구문 경계들의 표시기들로서 사용될 수 있다(노트의 중간에서 잘리는 것은 바람직하지 않다). 따라서 실제 서브-스트링 경계들은 노트들 사이에 있도록 조정될 수 있다.

실시예에서, 사용자는 허밍, 노래, 휘파람, 탭핑, 클랩핑, 또는 타악 보컬음들과 같은 복수의 질의 입력 양식들을 혼합함으로써 질의 스트링을 입력할 수 있다. 도 3의 방법은, 상이한 입력 양식들에 대한 오디오에서 근원적인 차들을 보여주는 적절한 센트로이드 파라미터들이 선택되는 경우에, 입력 양식들 사이의 변경이 거리 측정을 달성할 것이므로, 일반적으로 입력 양식들 사이의 변경들을 정확하게 결정할 수 있다. 상이한 입력 양식들의 오디오 특성들은 다음과 같이 요약될 수 있다:

- 노래는 분명한 피치를 가지며, 고조파 성분들이 노래 파형의 스펙트럼 표현에서 쉽게 검출될 수 있음을 의미한다. 달리 말하면, 스펙트럼 피크들은 하나의 단일 스펙트럼 피크(즉 노래의 피치라 흔히 칭해지는 제 1 고조파 또는 기본 주파수)의 다중들이다. 상이한 음성 레지스터들('가슴(chest)', '중모음(mid)', '머리, 가성(head, falsetto)' 노래)은 별개의 주파수 범위들을 가진다.
- 타악음들(클랩핑, 표면 상의 탭핑)은 기껏해야 명확하지 않은 피치를 가지며, 제 1 고조파로서 해석될 수 있는 다중 피크들이 있음을 의미한다. 더욱이, 타악음들은 과도 현상들(transients) 또는 클릭들(clicks)이다; 모든 주파수들에 걸쳐 번진 전력 및 진폭의 급속한 변화들이며, 이들은 쉽게 식별될 수 있다.
- 허밍은 임의의 두드러진 스펙트럼 피크들을 가지지 않고, 어떤 중간 범위의 주파수들을 가진 저-주파수 대역을 포함한다.
- 휘파람은 700Hz 내지 2800Hz의 피크(제 1 고조파들) 범위를 갖는다. 그것은 일부 약한 고조파들을 가진 거의 순수한 톤이다. 사람의 가장 낮은 휘파람 톤은 사람의 가장 높은 도달 가능한 노래 노트에 거의 가깝다(그래서, 휘파람은 노래보다는 높은 1과 1/2 내지 2 옥타브를 발생한다).
- 잡음은 사실상 확률적이다. 이것은 주파수들의 대역에 걸쳐(핑크 노이즈) 또는 전체 주파수 범위에 걸쳐(화이트 노이즈) 평평한 스펙트럼(하나의 에너지 레벨)을 유발한다.

당업자는 원한다면, 더 많은 입력 양식들을 구별할 수 있을 것이다.

상술된 자동 최소화 방법을 사용하여 서브-분할하기 위한 대안에 따라, 질의 스트링은 질의 스트링을 서브-스트링들의 시퀀스로 분해함으로써 서브-스트링들로 서브-분할될 수 있으며, 시퀀스의 각각의 서브-스트링은 미리 결정된 분류 기준을 충족시키고, 각각의 2개의 순차적인 서브-스트링들은 상이한 미리 결정된 분류 기준을 충족시킨다. 그래서, 오디오 프래그먼트의 부분이 규정된 일치성(예를 들어, 노래하기 위해 사용될 수 있는 규정된 범위 내에서 명확히 구별 가능한 노트들(피치))을 보이고, 다음 부분이 다른 일치성(예를 들면, 통상적으로 휘파람을 불기 위해 사용되는 범위에서 명확하게 구별 가능한 노트들, 평균적으로 1.5 옥타브 더 높은 피치)을 보인다면, 이것은 그 부분들의 상이한 분류를 유발하고, 분류의 변경이 새로운 서브-스트링의 시작으로 해석된다. 특정 분류 기준들이 전체 프래그먼트의 사전-분석(pre-analysis) 후에, 또는 사용자에게 의한 훈련 후에 완전히 결정될 수 있음을 이해할 것이다. 이러한 사전-분석은 예를 들면, 사용자가 남자 또는 여자이고, 노래, 휘파람 등에 사용되는 평균 피치에 대한 정보를 제공하는 것을 밝힐 수 있다. 다른 기준은 각각의 사람에 대해 동일할 수 있으며, 예를 들면, 음성 타악들이 주로 톤이 없다(예를 들면, 분명하게 식별 가능한 피치를 가지지 않은 잡음). 디폴트 및/또는 사람-특정 기준들을 확립하여, 질의 스트링(또는 질의 스트링에 의해 나타난 오디오 프래그먼트)이 더 분석된다. 분류에 사용되는 오디오 피쳐들은 스트링/프래그먼트들의 부분들에 대해 결정되고, 상이한 분류 기준들에 비교된다. 따라서, 시스템은 분류 기준들의 상이한 세트들을 포함하는 것이 바람직하며, 각각의 세트는 입력 양식들의 각각을 나타낸다. 분석되는 프래그먼트의 오디오 피쳐들은 각각의 기준 세트와 비교된다. 피쳐들이 세트들 중 하나와 (완전히

또는 거의) 매칭되면, 그 오디오 부분이 그 세트에 대응하는 입력 양식을 통해 가장 적절하게 지정되는 것이 확립된다. 분류 기술들은 잘 알려져 있다. 임의의 적절한 기술이 사용될 수 있다. 예시적인 분류 방식은 다음과 같다. 프래그먼트의 상대적으로 작은 부분들은 매 시간마다 분석된다(예를 들면, 구문의 1/3 또는 1/2). 분석 동안, 그러한 폭을 가진 분석 윈도우(analysis window)는 전체 오디오 프래그먼트를 통해 슬라이딩될 수 있다. 윈도우가 전체 오디오 프래그먼트의 일치된 부분 내에 완전히 있는 한, 대응하는 분류 기준 세트와의 상대적으로 근접한 매칭이 얻어질 것이다. 입력 양식 사이의 변경이 발생하는 경계에 걸쳐 윈도우가 이동할 때, 매칭은 윈도우가 더 이동함에 따라 덜 근접하고 더 감소하게 될 것이다. 윈도우가 다음 일치된 부분으로 충분히 멀리 이동되었을 때, 입력 양식에 대한 분류 기준 세트와의 더 근접한 매칭이 발견될 것이다. 윈도우가 그 부분으로 더 이동함에 따라 매칭이 개선될 것이다. 이러한 방식으로, 상대적으로 정확하게 경계들이 검출될 수 있다. 분석 윈도우는 예를 들면 10 내지 30ms의 프레임-단계들에서 이동될 수 있다. 전체 오디오 프래그먼트의 분석이 완료되었으면, 적어도 하나의 경계(전체 오디오 프래그먼트의 시작 및 종료 경계 외에도)가 검출되었고, 서브-스트링들이 경계들 내에서 형성될 수 있다.

상술된 분류 기술은 상술한 바와 같이 서브-스트링들로의 서브 분할들을 수행하기 위해 이용될 수 있다. 양호한 실시예에서, 분류는 서브-스트링의 일부가 분류를 사용하여 검출된 2개의 연속하는 경계들 내에 있도록 제약함으로써, 도 3의 자동 절차에 대한 전처리로서 이용된다. 제약된 동적 프로그래밍 기술들은 잘 알려져 있고, 본 명세서에 더 이상 기술하지 않는다.

상술된 분류 정보가 서브-스트링들의 위치 및 크기의 찾기를 최적화하는데 이용될 뿐만 아니라, 데이터베이스를 통한 검색을 개선시키는 데에도 이용될 수 있음을 알 것이다. 오디오 프래그먼트의 부분에 대한 최상의 매칭 일치 기준을 확립하여, 대부분에 경우에 또한 대응하는 입력 양식이 알려져 있다. 이러한 정보는 찾은 부분에 대응하는 서브-스트링에 대한 검색을 개선시키기 위해 이용될 수 있다. 예를 들면, 최적화된 데이터베이스는 각각의 입력 양식에 이용될 수 있다. 대안적으로, 데이터베이스는 상이한 입력 양식들을 사용하여 동일한 프래그먼트에 대한 검색을 지원할 수 있다. 그 후, 입력 양식은 하나의 부가의 질의 항목이고, 데이터베이스는 각각의 오디오 프래그먼트(예를 들면, 구문)에 대해, 프래그먼트를 지정하는데 사용된 입력 양식을 저장한다.

도 2에 도시된 방법에서, 서브-스트링들의 수의 초기 추정치는 더 이상 변경되지 않는다. 초기 추정치는 전체 프래그먼트에 존재하는 것으로 예상되는 서브-스트링들의 최대 수를 기술하는 것이 바람직하다. 프래그먼트가 '최악의 경우(worst case)' 가정보다 더 일치될 수 있기 때문에, 동일한 처리가 더 적은 서브-스트링들에 대해서도 또한 반복되는 것이 바람직하다. 도 2의 예에서, 또한, 2개의 서브-스트링들로의 분해가 행해질 수 있고, 데이터베이스를 통해 검색이 수행될 수 있다. 데이터베이스는 또한 전체 스트링에 대해 검색될 수 있다. 이러한 방식으로, 전체 스트링의 매칭이 3개의 서브-스트링들, 2개의 서브-스트링들 및 1개의 서브-스트링(즉, 전체 스트링)에 대해 얻어질 수 있다. 3개의 결과들이 비교될 수 있고, 가장 가능성 있는 것이 클라이언트에게 제공된다. 따라서, 실제로, 질의 스트링은 많은 방식으로 분해될 수 있으며, 각각의 분해 결과들은 데이터베이스에서 독립적으로 검색될 수 있는 다수의 서브-스트링들을 유발한다. 그래서 전체로서 질의 스트링은, 질의 스트링의 2개로의 분해를 유발하는 서브-스트링들로부터 독립적으로, 질의 스트링의 3개로의 분해를 유발하는 서브-스트링들 등으로부터 독립적으로 검색될 수 있다. 서브-스트링에 대한 각각의 검색은 가능성 있는 후보들의 N-최상 목록을 유발할 수 있다. 이 N-최상 목록은, 그들 거리 상에서 서브-스트링으로 순서화된 데이터베이스 내의 모든 멜로디들의 목록일 수 있다. 예를 들면, 모든 가능한 분해들에 대한 목록들을 사용자에게 제공될 한 목록으로 조합함으로써, 전체 결과들이 생성될 수 있다. 조합은 모든 목록들을 병합하고, 그들 서브-스트링으로부터 정규화된 거리들에 대해 정렬시킴으로써 달성될 수 있다.

상술한 바와 같이, 질의 스트링을 분해하는 단계는, 각각이 구문에 실질적으로 대응하는 서브 스트링들로 질의 스트링을 분해하는 단계를 포함한다. 이것은 단지 분해 단계일 수 있거나, 입력 양식의 변경들을 위한 서브-분할에 목표된 분해를 수행한 후에 여분의 분해와 같은 다른 분해 단계들/기준과 조합하여 사용될 수 있다. 구문들은 임의의 적절한 방식을 사용하여 검출될 수 있다. 구문들은 흔히, 허밍의 감속으로 종료된다. 또는, 구문들은 큰 톤 차들(즉, 간격들)과 큰 톤 지속기간들에 의해 구별된다. 구문 검출 알고리즘들은 예를 들면, "ICMC 2001 회보에서 Cambouropoulos, E.(2001)에 의한 The local boundary detection model(Ibdlm) and its application in the study of expressive timing" 및 "2003년 5월 8-9-10일, 이탈리아 피렌체에서 뮤지컬 정보 과학에 관한 XIV 세미나의 회보(XIV CIM 2003)에서 Ferrand, M., Nelson, P와 Wiggins, G.(2003)에 의한 Memory and melodic density: A model for melody segmentation"로부터 알려져 있다.

본 발명이 컴퓨터 프로그램들, 특히 본 발명을 실시하기에 적합한 캐리어 상 또는 캐리어 내의 컴퓨터 프로그램들로 확장될 수도 있음을 인식할 것이다. 프로그램은 부분적으로 컴파일된 형태와 같은 소스 코드, 오브젝트 코드, 코드 매개 소스 및 오브젝트 코드의 형태로 될 수 있거나, 또는 본 발명에 따른 방법의 구현에 사용하기에 적합한 임의의 다른 형태로 될 수 있다. 캐리어는 프로그램을 담을 수 있는 임의의 엔티티 또는 장치이다. 예를 들면, 캐리어는 ROM, 예를 들면 CD ROM 또는 반도체 ROM과 같은 저장 매체, 또는 예를 들면 플로피 디스크 또는 하드 디스크와 같은 자기 기록 매체를 포함할 수

있다. 또한, 캐리어는 전기 또는 광 케이블을 통하거나 무선 또는 다른 수단에 의해 전달될 수 있는 전기 또는 광 신호와 같은 전송 가능한 캐리어일 수 있다. 프로그램이 그러한 신호로 구현될 때, 캐리어는 그러한 케이블 또는 다른 장치나 수단에 의해 구성될 수 있다. 대안적으로, 캐리어는 프로그램이 구현되는 집적 회로일 수 있으며, 집적 회로는 관련 방법을 수행하거나 또는 수행에 사용되기에 적합하다.

상기 언급한 실시예들이 본 발명을 제한하기보다는 설명하기 위함이며, 당업자들은 첨부된 청구항들의 범위를 벗어나지 않고 대안적인 실시예들을 설계할 수 있음을 주지한다. 청구항들에서, 괄호 안에 놓인 임의의 참조 부호들은 청구항을 제한하는 것으로 구성되어서는 안 된다. 동사 '포함하다(comprise)' 및 그 동사 변화는 청구항에 언급된 구성 요소들 또는 단계들이 외의 구성 요소들 또는 단계들의 존재를 배제하지 않는다. 구성 요소 앞에 있는 단어 "하나의"는 복수의 구성 요소들의 존재를 배제하지 않는다. 본 발명은 여러 개의 개별 구성 요소들을 포함하는 하드웨어, 및 적절히 프로그래밍된 컴퓨터에 의해 구현될 수 있다. 여러 개의 수단을 열거하는 장치 청구항들에서, 이들 여러 개의 수단은 하나 또는 동일 항목의 하드웨어에 의해 구현될 수 있다. 특정 측정들이 상이한 종속 청구항들에서 서로 인용되는 단순한 사실은 이들 측정들의 조합이 이점으로 사용될 수 없음을 나타내지 않는다.

**도면의 간단한 설명**

도 1은 본 발명에 따른 방법을 수행하는 분배 시스템의 블록도.

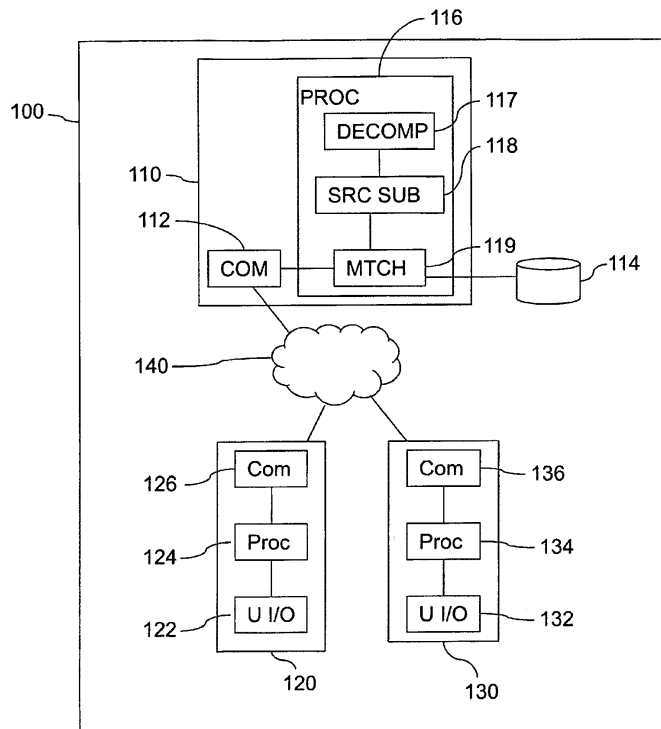
도 2는 본 발명에 따른 방법을 수행하는 독립형 장치의 블록도.

도 3은 본 발명의 실시예의 흐름도.

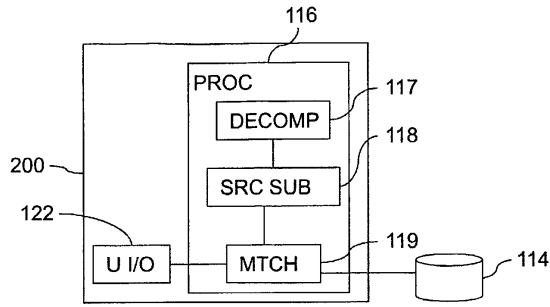
도 4a 및 도 4b는 예시적 서브-분할들을 도시한 도면.

**도면**

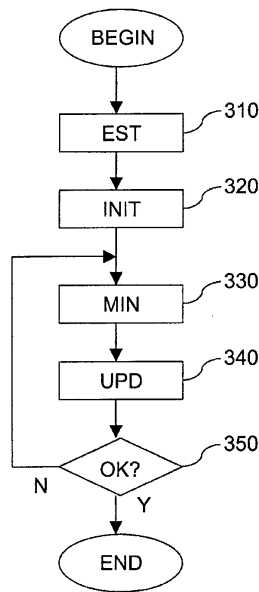
**도면1**



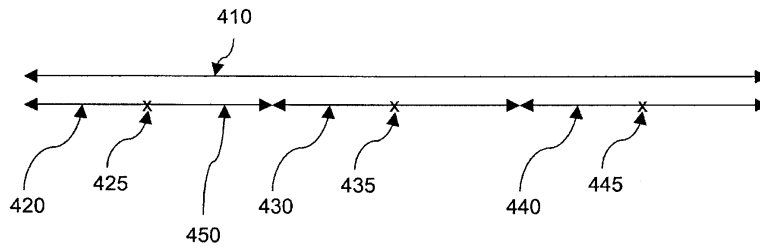
도면2



도면3



도면4a



도면4b

