



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0064683
(43) 공개일자 2017년06월12일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G10L 19/02 (2006.01)
G10L 25/90 (2013.01)

(52) CPC특허분류
G06F 17/30758 (2013.01)
G06F 17/30769 (2013.01)

(21) 출원번호 10-2015-0170444

(22) 출원일자 2015년12월02일

심사청구일자 2015년12월02일

(71) 출원인

충남대학교산학협력단

대전광역시 유성구 대학로 99 (궁동, 충남대학교)

(72) 발명자

박정희

대전광역시 유성구 은구비남로 56 908동 1803호
(노은동, 열매마을아파트9단지)

(74) 대리인

김건수

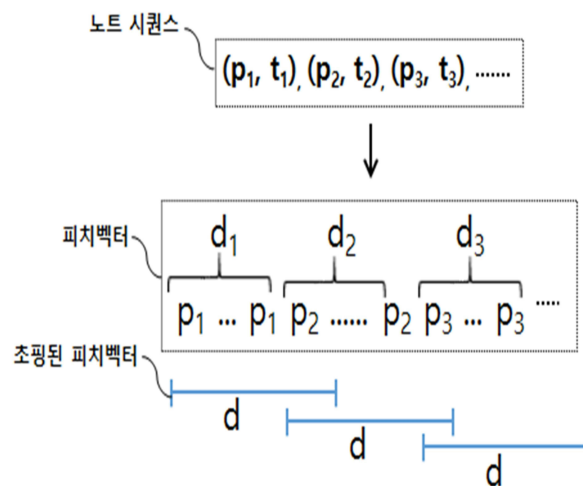
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 다중 스펙트럴 해싱 및 스케일된 오픈-엔드 동적 타임 워핑에 기반한 허밍 질의 방법 및 시스템

(57) 요약

본 발명은 다중 스펙트럼 해싱 및 스케일된 오픈-엔드 동적 타임 워핑에 기반한 허밍 질의 방법 및 시스템에 관한 것으로, 입력되는 허밍 질의와 기 저장된 복수의 음원에 대해 다중 스펙트럴 해싱 및 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 템포 변동 및 키 변동을 보완하고 상기 허밍 질의에 대한 음원을 검색하기 위한 시간을 현저하게 줄일 수 있는 방법 및 시스템에 관한 것이다.

대표도 - 도2



(52) CPC특허분류

G10L 19/02 (2013.01)

G10L 25/90 (2013.01)

명세서

청구범위

청구항 1

미디 파일과 허밍 질의에 대해서 다중 스펙트럴 해싱을 수행하여 상기 허밍 질의에 대한 후보 집단을 생성하는 단계; 및

스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 후보 집단 중에서 상기 허밍 질의와 유사도 매칭을 수행하는 단계;를 포함하며,

상기 허밍 질의로 음악검색을 수행하는 것을 특징으로 하는 허밍 질의 방법.

청구항 2

청구항 1에 있어서,

상기 허밍 질의 방법에 있어서,

미디 파일을 적어도 하나 이상의 노트 시퀀스로 변환하고, 상기 노트 시퀀스를 피치벡터로 변환한 다음, 상기 피치벡터를 초핑하는 것을 포함하는 데이터 프로세싱 단계;를 더 포함하는 것을 특징으로 하는 허밍 질의 방법.

청구항 3

청구항 2에 있어서,

상기 다중 스펙트럴 해싱은,

상기 초핑된 피치벡터를 사용하여 복수의 고유함수를 찾아내고, 상기 복수의 고유함수에 대해서 다시 복수회에 걸쳐서 복수개의 고유함수를 랜덤하게 선택하여 복수의 비트를 가진 복수의 바이너리 임베딩 스페이스를 생성하며, 상기 허밍 질의의 길이를 선형적으로 스케일하여 생성한 복수의 상기 허밍 질의에 대해서, 상기 생성된 각 바이너리 임베딩 스페이스에 대해서 스케일된 허밍 질의로 미리 정해진 수의 인접한 이웃을 탐색하고, 각 탐색한 미리 정해진 수의 인접한 이웃으로부터 미디 파일번호와 상기 노트의 시작점을 포함하는 후보군 리스트를 생성하는 것을 포함하는 것을 특징으로 하는 허밍 질의 방법.

청구항 4

청구항 3에 있어서,

상기 스케일된 오픈-엔드 동적 타임 워핑은,

상기 허밍 질의를 상기 생성한 후보군 리스트와 매칭을 수행하여 유사도를 측정하여, 후보 미디 파일의 랭킹을 리턴하는 것을 특징으로 하는 허밍 질의 방법.

청구항 5

미디 파일과 허밍 질의에 대해서 다중 스펙트럴 해싱을 수행하여 상기 허밍 질의에 대한 후보 집단을 생성하는 다중 스펙트럴 해싱부; 및

스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 후보 집단 중에서 상기 허밍 질의와 유사도 매칭을 수행하는 유사도 매칭부;를 포함하며,

상기 허밍 질의로 음악검색을 수행하는 것을 특징으로 하는 허밍 질의 시스템.

청구항 6

청구항 5에 있어서,

상기 허밍 질의 시스템에 있어서,

미디 파일을 적어도 하나 이상의 노트 시퀀스로 변환하고, 상기 노트 시퀀스를 피치벡터로 변환한 다음, 상기 피치벡터를 초핑하는 것을 포함하는 데이터 프로세싱부;를 더 포함하는 것을 특징으로 하는 허밍 질의 시스템.

청구항 7

청구항 6에 있어서,

상기 다중 스펙트럴 해싱은,

상기 초핑된 피치벡터를 사용하여 복수의 고유함수를 찾아내고, 상기 복수의 고유함수에 대해서 다시 복수회에 걸쳐서 복수개의 고유함수를 랜덤하게 선택하여 복수의 비트를 가진 복수의 바이너리 임베딩 스페이스를 생성하며, 상기 허밍 질의의 길이를 선형적으로 스케일하여 생성한 복수의 상기 허밍 질의에 대해서, 상기 생성된 각 바이너리 임베딩 스페이스에 대해서 스케일된 허밍 질의로 미리 정해진 수의 인접한 이웃을 탐색하고, 각 탐색한 미리 정해진 수의 인접한 이웃으로부터 미디 파일 번호와 상기 노트의 시작점을 포함하는 후보군 리스트를 생성하는 것을 포함하는 것을 특징으로 하는 허밍 질의 시스템.

청구항 8

청구항 7에 있어서,

상기 스케일된 오픈-엔드 동적 타임 워핑은,

상기 허밍 질의를 상기 생성한 후보군 리스트와 매칭을 수행하여 유사도를 측정하여, 후보 미디 파일의 랭킹을 리턴하는 것을 특징으로 하는 허밍 질의 시스템.

발명의 설명

기술 분야

- [0001] 본 발명은 다중 스펙트럴 해싱 및 스케일된 오픈-엔드 동적 타임 워핑에 기반한 허밍 질의 방법 및 시스템에 관한 것으로, 더욱 상세하게는 입력되는 허밍 질의와 기 저장된 복수의 미디파일에 대해 다중 스펙트럴 해싱을 수행하여 상기 허밍 질의와 유사도가 높은 상기 미디파일의 후보 집단군을 생성하고, 상기 후보 집단군에 대해 스케일된 오픈-엔드 동적 타임 워핑 기법을 적용하여 상기 허밍 질의에 대해 유사도를 매칭하여 미디파일 검색을 수행하는 방법 및 시스템에 관한 것이다.

배경 기술

- [0002] 최근 디지털 음원의 시장이 방대해지고 인터넷이 급속도로 확대됨에 따라 언제 어디서나 용이하게 음원 정보를 검색하고자 하는 사용자의 요구도 증대되고 있다.
- [0003] 음원 정보 검색 기술에는 오디오 키 추출(Audio Key Detection), 오디오 멜로디 추출(Audio Melodic Extraction), 오디오 템포 추출(Audio Temp Estimation), 음악 장르 분류(Music Genre Classification) 및 허밍 질의 검색(Query By Humming)시스템 등 다양한 분야가 있다.
- [0004] 특히 상기 허밍 질의 검색 시스템은 사용자가 특정 음원을 검색하기 위해 허밍으로 상기 시스템에 질의하여 해당 음원을 검색할 수 있도록 한다.
- [0005] 또한 상기 허밍 질의 검색 시스템은 다른 검색 시스템보다 질의 입력이 용이하고, 특정 음원에 대한 정보 없이 단순한 허밍만으로도 상기 사용자가 원하는 음원을 검색할 수 있는 장점 때문에 사용자의 관심이 증대되고 있다.
- [0006] 그러나 상기 사용자가 입력하는 허밍은 원 음원에 비해 상대적으로 키(음의 높낮이)와 템포(음의 빠르기)가 부정확하고, 상기 허밍을 입력하는 환경에 따라 상기 허밍에 노이즈가 혼합될 수 있어 음원 검색 시 상당한 오류를 초래할 수 있다.
- [0007] 이러한 문제점을 해결하기 위해 입력된 사용자의 허밍 질의로부터 정확한 특징을 추출하기 위한 방법과 상기 추출된 특징을 기반으로 상기 입력된 허밍 질의와 음원에 대하여 최적화된 유사도 매칭 방법이 개발되고 있다 ("J.-S. Jang, H.-R. Lee, A general framework of progressive filtering and its application to query by singing/humming, IEEE Trans. Audio, Speech Lang. Process.16(2008)350??358.", "X. Wu, M. Li, A top-down

approach to melody match in pitch contour for query by humming, in: Proceedings of International Conference of Chinese Spoken Language Processing, 2006." 및 "J. Jang, J. Chen, M. Kao, A query-by-singing system based on dynamic programming, in: Proceedings of International Workshop on Intelligent Systems Resolutions, 2000." 참고).

[0008] 우선 상기 허밍 질의와 원곡 간의 멜로디 유사성을 측정하는 방법에 있어서, 상기 허밍 질의의 템포 변동과 키 변동에 의해 야기되는 상기의 문제점을 해결하기 위해서 스케일링(Scaling) 방법과 평균 이동(Mean shifting)방법이 광범위하게 사용되어 왔으며, 특히 상기 허밍 질의의 글로벌 템포 변동을 극복하기 위한 가장 기본적인 방법으로는 선형 스케일링(Linear Scaling, LS, 이하 LS라 칭함)방법이 있다.

[0009] 또한 상기 LS 방법은 사용자가 허밍을 원곡보다 빠르게 혹은 느리게 부를 수 있다는 것을 고려하여 상기 허밍 질의의 음을 전역적으로 줄이거나 늘려서 스케일링하는 방법으로 상기 허밍 질의는 스케일링 팩터에 의해 스케일링되어 균등하게 늘이거나 줄인 후, 원곡과 매칭 된다.

[0010] 이는 계산량이 적으면서도 비교적 높은 성능을 나타내지만 상기 허밍 질의와 원곡의 빠르기 차이를 극복하기 위해 상기 허밍 질의의 시퀀스에 대한 전체 길이를 균등하게 스케일링하기 때문에 부분적인 템포 변동 문제(Local Temp Variation)를 해결할 수 없는 한계가 있다.

[0011] 또한 상기 LS방법의 한계를 해결하기 위해 탑-다운(Top-Down)방식으로 상기 허밍 질의와 음원 간의 전역적인 매칭을 시도한 후, 부분적인 영역에서 얼라이먼트(Alignment)를 반복적으로 수행함으로써, 상기 부분적인 템포 변동 문제를 해결하는 RA(Recursive Alignment)방법이 개발되었으나 키(음의 높낮이) 변동(Key Variation)에 대한 문제점을 해결하지 못하고 있다.

[0012] 또한 타임 시퀀스 간의 유사도 매칭 방법으로 DTW(Dynamic Time Warping, DTW) 또는 OEDTW(Open-End DTW) 방법이 있다. 상기 DTW 또는 OEDTW 방법은 상기 허밍 질의 기반 음악 검색 시스템에 가장 일반적으로 사용되며, 비선형적 길이의 차이가 있는 타임 시퀀스에 대한 유사도 매칭에 효과적인 방법이다("P.Tormene, T.Giorgino, S.Quanlini, M.Stefanelli, Matchingincomplete time series with dynamic time warping: an algorithm and anapplication to post-stroke rehabilitation, Artif.Intell.Med.45(2009) 11??34." 참고).

[0013] 예를 들어 $Q = q_1, \dots, q_m$ 와 $C = c_1, \dots, c_n$ 를 각각 상기 허밍 질의와 타겟 음원(Target Song)이라고 가정하면, 일반적으로 상기 n는 m보다 큰 수를 갖는다, 또한 상기 OEDTW는 키 변동에 대한 문제점을 극복하기 위해 상기 쿼리 Q와 타겟 음원 C를 평균이 0이 되도록 쉬프트 한다. 이때, 상기 Q로부터 쉬프트 된 시퀀스는

$\hat{Q} = \hat{q}_1, \dots, \hat{q}_m$ 로 나타내며, 상기 C로부터 쉬프트 된 시퀀스는 $\hat{C} = \hat{c}_1, \dots, \hat{c}_n$ 이다. 한편 d(i, j)를 상기 $\hat{Q}$ 의 i-번째와 상기 $\hat{C}$ 의 j-번째 특징 간의 거리(distance)라면 상기 $\hat{Q} = \hat{q}_1, \dots, \hat{q}_m$ 와 상기 $\hat{C} = \hat{c}_1, \dots, \hat{c}_n$ 사이의 거리 D(i, j)는 다음의 [수학식 1]에 의해서 파생될 수 있다.

[0014] [수학식 1]

$$\begin{aligned} D(1, 1) &= d(1, 1) \\ D(i, 1) &= \infty, \quad i = 2, \dots, m \\ D(1, j) &= \infty, \quad j = 2, \dots, n \\ D(i, j) &= d(i, j) - \eta + \min \left\{ \begin{array}{l} D(i-2, j-1) \\ D(i-1, j-1) \\ D(i-1, j-2) \end{array} \right\}, \\ 2 \leq i \leq m, \quad 2 \leq j \leq n \end{aligned}$$

[0015]

[0016] 여기서 상기 파라미터 η 는 긴 매칭 경로의 탐색 영역을 보장하기 위해 사용된다. 따라서 상기 OEDTW방법에 의해 상기 쿼리와 상기 타겟 음원 사이의 거리 $dis_{OEDTW}(Q, C)$ 를 구할 수 있으며, 상기 $dis_{OEDTW}(Q,$

C)는 상기 D행렬의 마지막 열에서 최소값을 취함으로써 구해진다.

[0017] 상기 DTW방법은 비선형적인 로컬 템포 변동을 고려하여 최적화된 매칭관계를 찾으며, 특히 OEDTW(Open-End DTW, OEDTW)방법은 입력된 허밍 질의와 음원의 앞부분과 매칭하여 비유사도를 계산함으로써, 글로벌 템포 변동과 로컬 템포 변동의 문제점을 동시에 해결한다.

[0018] 그러나 상기 DTW 및 OEDTW방법은 상기 음원 검색에 대한 수행을 완료하기 전에는 템포 변동을 보완하기 위해 상기 질의에 대한 최적의 전역적 스케일을 결정할 수 없으며, 키 변동을 보완하기 위해 상기 질의 시퀀스의 평균을 이동시켜 가면서 OEDTW방법을 반복적으로 수행하기 때문에 시간적으로 매우 큰 비용을 치르게 되는 문제점이 있다("P.Tormene, T.Giorgino, S.Quanlini, M.Stefanelli, Matchingincomplete time series with dynamic time warping: an algorithm and anapplication to post-stroke rehabilitation, Artif.Intell.Med.45(2009) 11??34." 참고).

[0019] 또한 최근에 들어서는 대형 데이터 셋에서 근사 최근접 이웃(Approximate Nearest Neighbor, ANN)탐색을 위한 해싱 기반 방법들이 제안되고 있다.

[0020] 특히 LSH(Local Sensitive Hashing)방법은 멀리 떨어져 있는 것들 보다 근접한 것들의 충돌(Collision) 확률이 훨씬 높게 되도록 다수의 해시함수들을 사용하여 데이터 포인트를 해시한다("A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of International Conference on Very Large Data Bases, 1999." 참고).

[0021] 또한 상기 LSH방법은 상기 데이터 포인트들이 가우시안 분포와 같은 P-stable 분포를 따르는 것으로 가정하고 상기 가우시안 분포로부터 랜덤 프로젝션(Projection)벡터를 유도한다.

[0022] 또한 상기 LSH방법에서의 해시함수 $H = [h_1, \dots, h_k]$ 는 다음의 [수학식 2]와 같이 상기 P-stable분포로부터 랜덤하게 샘플링 된 복수의 랜덤 프로젝션 벡터에 의해 구성된다.

[0023] [수학식 2]

$$h_i(x) = \left\lfloor \frac{w_i^T x + b_i}{r} \right\rfloor$$

[0024]

[0025] 여기서 상기 r은 하나의 해시 값에 대응하는 프로젝션 라인 간격의 폭을 지정하며, 상기 w_i 는 표준 정규 분포(Standard Normal Distribution)로부터 선택된 랜덤 벡터를 나타내며, 상기 b_i 는 범위 (0, r)에서 균일하게 선택된다("M. Datar, N. Immorlica, P. Indyk, V. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the Twentieth Annual Symposium on Computational Geometry, 2004." 참고).

[0026] 또한 상기 LSH방법은 먼저 해시 함수 패밀리 $\mathcal{F}$ 을 포함하는 멀티 해시 테이블을 구성하고, 상기 각 해시 테이블의 상기 $\mathcal{F}$ 로부터 랜덤하게 적어도 하나 이상의 k개의 해시 함수를 선택하며, 상기 해시된 입력 쿼리를 위한 버킷의 조합에서의 검색은 근사 최근접 이웃들을 검색하여 리턴 한다.

[0027] 또한 상기 LSH방법은 상기 허밍 질의에 대해 적은 수의 음원 후보들을 매우 효율적으로 데이터베이스로부터 검색하며, 상기 질의와 상기 후보들을 각각 매칭하여 유사도가 높은 순으로 최종 랭킹을 측정하는 절차를 수행한다.

[0028] 그러나 상기 LSH방법은 정수의 집합에 데이터 포인트를 매핑 하는 해시 함수를 생성하기 위해 임의의 프로젝션들을 사용하며, 높은 정밀도를 달성하기 위해 상기 프로젝션은 긴 이진 코드를 필요로 하기 때문에 낮은 재현율을 야기시킬 수 있다.

[0029] 또한 상기 LSH방법과 같이 데이터 독립적인 해싱 기반 방법과는 달리 SH(Spectral Hashing)방법, KMH(K-Means Hashing) 및 AGH(Anchor Graph Hashing)를 포함하여 몇몇의 데이터 의존적인 방법들도 있다. 상기 데이터 의존적인 방법들은 트레이닝 데이터로부터 해시 함수를 학습함으로써 오리지널 스페이스의 유사 데이터 포인트를 이진 임베딩 스페이스의 근접 포인트들로 매핑 시킨다.

[0030] 또한 상기 SH방법은 다음의 [수학식 3]에 따라 입력 벡터 $\{x_1, \dots, x_n\}$ 로부터 k-비트 이진 코드 $\{H(x_1), \dots, H(x_n)\}$ 를 위한 임베딩을 찾는다.

[0031] [수학식 3]

$$\begin{aligned} &\text{minimize} \quad \sum_{ij} \text{sim}(x_i, x_j) \|H(x_i) - H(x_j)\|^2 \\ &\text{subject to} \quad H(x_i) \in \{-1, 1\}^k, \\ &\quad \sum_i H(x_i) = 0, \\ &\quad \frac{1}{n} \sum_i H(x_i) H(x_i)^T = I. \end{aligned}$$

[0032]

[0033] 여기서 상기 $\text{sim}(x_i, x_j)$ 는 데이터 포인트 상기 x_i 와 x_j 사이의 유사도를 의미하며, 상기 유사도는 $\text{sim}(x_i, x_j) = e^{-\|x_i - x_j\|^2 / \epsilon^2}$ 로 설정된다. 또한 상기 두 번째 및 세 번째에 해당하는 제약 조건은 균형을 이루고 관련성이 없는 코드들을 생성하기 위한 것으로서 NP-hard 문제를 야기 한다. ("Y.Weiss, A.Torralba, R.Fergus, Spectral hashing, in: Proceedings of Advances in Neural Information Processing Systems, vol. 21, 2008, pp. 1753??1760." 참고).

[0034] 상기 NP-hard 문제는 제약 조건 $H(x_i) \in \{-1, 1\}^k$ 을 제거함으로써 완화되고 라플라스 고유 함수에 의해 해결된다.

[0035] 한편 트레이닝 데이터의 의미 관계(Semantic relationship)에 기초하여 해시 함수를 유도하는 상기 SH방법은 LSH에 비하여 우수한 성능을 보인다.

[0036] 또한 상기 LSH방법을 기반으로 하는 종래의 허밍 질의 기반 음악 검색 시스템은, 멜로디 데이터베이스로부터 피치백터를 추출하여 멜로디 단편(Melodic fragments)의 인덱스를 구성한다. 또한 상기 종래의 허밍 질의 기반 음악 시스템은 상기 허밍 질의에 대한 음원의 후보 리스트를 얻기 위해 상기 LSH방법을 사용하여 데이터베이스에 저장된 상기 멜로디 단편의 인덱스로부터 최근접 이웃들을 검색한다.

[0037] 또한 상기 LSH방법을 기반으로 하는 종래의 허밍 질의 기반 음악 검색 시스템은, 상술한 RA를 사용하여 전체 쿼리와 각 후보의 멜로디 세그먼트를 비교함으로써, 상기 얻은 음원의 후보 리스트들에 대한 우선순위(Final Ranking)를 설정하여 사용자에게 제공한다("M. Rynanen, A. Klaouri, Query by humming of midi and audio using locality sensitive hashing, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2008."참고).

[0038] 상술한 허밍 질의 기반 검색 시스템에 적용되는 다양한 방법들은 템포 변동 및 키 변동에 대한 해결점을 제시하고 있지만 검색시간이 오래 걸리는 문제점이 있으며 검색에 대한 정확도가 떨어지는 단점이 있다.

[0039] 따라서 다중 스펙트럴 해싱을 이용하여 입력된 해밍 질의와 기 저장한 미디어파일을 피치백터로 변환하고, 상기 미디어파일의 피치백터를 각 노트의 시작점에 따라 쪼핑(Chopping)하여 상기 쪼핑한 피치백터와 상기 허밍 질의의 피치백터에 대한 유사도를 측정함으로써, 상기 미디어파일의 중간에 대한 허밍 질의에 대해서도 그 적용이 가능하고 허밍 질의 기반 음악 검색의 정확도 성능을 향상할 수 있는 방법 및 시스템을 제공하고자 한다.

발명의 내용

해결하려는 과제

[0040] 본 발명은 상기와 같은 문제점을 해결하기 위해 창작 된 것으로서, 입력되는 허밍 질의와 복수의 음원에 대해 다중 스펙트럴 해싱을 수행함으로써, 상기 음원의 중간에 대한 허밍 질의에 대해서도 그 적용이 가능하도록 하

고, 해당 허밍 질의와 유사도가 높은 음원의 후보 집단을 생성하여, 상기 후보 집단에 대해 스케일된 오픈-엔드 동적 타임 워핑 기법을 적용함으로써, 허밍 질의를 기반으로 하는 음원 검색의 정확도를 향상시킬 수 있는 방법 및 시스템을 제공하는 것을 그 목적으로 한다.

과제의 해결 수단

- [0041] 본 발명의 일 실시예에 따른 허밍 질의 방법은 미디어파일과 허밍 질의에 대해서 다중 스펙트럴 해싱을 수행하여 상기 허밍 질의에 대한 후보 집단을 생성하는 단계, 및 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 후보 집단 중에서 상기 허밍 질의와 유사도 매칭을 수행하는 단계를 포함하며, 상기 허밍 질의로 음악검색을 수행하는 것을 특징으로 한다.
- [0042] 또한 상기 허밍 질의 방법에 있어서, 미디어파일을 노트 시퀀스로 변환하고, 상기 노트 시퀀스를 피치벡터로 변환한 다음, 상기 피치벡터를 초핑하는 것을 포함하는 데이터 프로세싱 단계를 더 포함하는 것을 특징으로 한다.
- [0043] 또한 상기 다중 스펙트럴 해싱은, 상기 초핑된 피치벡터를 사용하여 복수의 고유함수를 찾아내고, 상기 복수의 고유함수에 대해서 다시 복수회에 걸쳐서 복수개의 고유함수를 랜덤하게 선택하여 복수의 비트를 가진 복수의 바이너리 임베딩 스페이스를 생성하며, 상기 허밍 질의의 길이를 선형적으로 스케일하여 생성한 복수의 상기 허밍 질의에 대해서, 상기 생성된 각 바이너리 임베딩 스페이스에 대해서 스케일된 허밍 질의로 미리 정해진 수의 인접한 이웃을 탐색하고, 각 탐색한 미리 정해진 수의 인접한 이웃들로부터 미디어 파일 번호와 상기 노트의 시작점을 포함하는 후보군 리스트를 생성하는 것을 포함하는 것을 특징으로 한다.
- [0044] 또한 상기 스케일된 오픈-엔드 동적 타임 워핑은, 상기 허밍 질의를 상기 생성한 후보군 리스트와 매칭을 수행하여 유사도를 측정하여, 후보 미디어파일의 랭킹을 리턴하는 것을 특징으로 한다.
- [0045] 아울러 본 발명의 일 실시예에 따른 허밍 질의 시스템은 미디어파일과 허밍 질의에 대해서 다중 스펙트럴 해싱을 수행하여 상기 허밍 질의에 대한 후보 집단을 생성하는 다중 스펙트럴 해싱부 및 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 후보 집단 중에서 상기 허밍 질의와 유사도 매칭을 수행하는 유사도 매칭부를 포함하며, 상기 허밍 질의로 음악검색을 수행하는 것을 특징으로 한다.
- [0046] 또한 상기 허밍 질의 시스템에 있어서, 미디어파일을 노트 시퀀스로 변환하고, 상기 노트 시퀀스를 피치벡터로 변환한 다음, 상기 피치벡터를 초핑하는 것을 포함하는 데이터 프로세싱부를 더 포함하는 것을 특징으로 한다.

발명의 효과

- [0047] 본 발명은 장치 및 그 방법에 관한 것으로, 미디어파일을 피치벡터로 변환하고, 상기 각 피치벡터에 대해 각 노트의 시작점을 기준으로 각각 초핑하여 입력된 허밍 질의에 대해 유사도가 높은 미디어파일의 후보군을 생성하여 유사도 측정을 수행할 수 있도록 함으로써, 상기 허밍 질의가 상기 미디어파일의 어느 부분에 해당하더라도 검색이 가능하도록 하여, 검색에 대한 정확도를 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

- [0048] 도 1은 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템에 있어서, 미디어파일의 검색에 대한 절차를 나타낸 흐름도이다.
- 도 2는 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템에 있어서, 미디어파일에 대한 데이터 프로세싱을 설명하기 위한 예시도이다.
- 도 3은 본 발명의 일 실시예에 따른 다중 바이너리 임베딩 스페이스를 구성하는 방법을 설명하기 위한 예시도이다.
- 도 4는 본 발명의 일 실시예에 따른 입력된 허밍 질의를 기반으로 미디어파일을 검색하는 방법을 설명하기 위한 예시도이다.
- 도 5는 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템과 종래의 기술에 대한 성능을 비교한 예시도이다.
- 도 6은 본 발명의 일 실시예에 따른 허밍 질의 시스템의 구성을 나타낸 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0049] 이하, 본 발명의 바람직한 실시 예를 첨부된 도면을 참조하여 상세히 설명한다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.
- [0050] 도 1은 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템에 있어서, 미디파일의 검색에 대한 절차를 나타낸 흐름도이다.
- [0051] 도 1에 도시한 바와 같이, 미디 파일을 검색하기 위한 절차는 우선, 허밍 질의 시스템(100)을 통해 미디파일 데이터베이스(200)에 저장된 모든 미디 파일의 노트 시퀀스를 피치벡터로 변환하여 DB1(300)에 저장한다(S110).
- [0052] 상기 미디파일이 포함하는 멜로디는 (p_i, t_i) 와 같은 포맷의 노트 시퀀스로 구성되어 있으며, 상기 p_i 는 해당 노트의 피치값이고 상기 t_i 는 상기 노트가 지속되는 시간을 의미한다. 한편 상기 노트는 상기 멜로디에 대한 음의 고저를 나타낸다.
- [0053] 또한 상기 허밍 질의 시스템(100)은 상기 미디파일에 대한 노트 시퀀스를 구성하는 각각의 노트에 대해 길이 d_i 의 시퀀스로 전환함으로써, 변환한다.
- [0054] 한편 상기 d_i 는 샘플링 비율(Sampling Rate)과 프레임 사이즈로부터 계산된다. 예를 들어, 상기 샘플링 비율이 8kHz이고 상기 프레임 사이즈는 피치 트래킹에서 오버랩핑(overlapping)없이 256 포인트로 설정되었다면 초당 피치값의 수는 31.25가 된다. 따라서 상기 노트가 지속되는 시간인 t_i 에 상응하는 피치값의 수는 $31.25 \times t_i$ 로 계산될 수 있다. 즉, 상기 피치벡터는 상기 t_i 에 상응하는 피치값의 수로 구성된다.
- [0055] 다음으로 상기 허밍 질의 시스템(100)은 상기 변환한 피치벡터를 각 노트의 시작점부터 길이 d 의 크기로 각각 쪼개(Chopping)하여 DB2(400)에 저장한다(S120). 한편 상기 d 는 실험에서 110으로 하였다.
- [0056] 또한 상기 미디파일에 대한 피치벡터를 각 노트에 대한 시작점으로부터 쪼개는 이유는, 일반적으로 사용자가 허밍을 이용하여 음악을 검색하는 경우, 상기 허밍은 해당 노래의 어느 부분에서든지 시작될 수 있기 때문이다.
- [0057] 이에 따라 상기 미디파일에서 상기 허밍이 시작되는 지점을 찾기 위해서는 상기 미디파일의 어느 부분에서든지 상기 허밍 질의가 매칭될 수 있어야 된다. 즉, 일반적으로 상기 어느 노트에서 사용자의 허밍이 시작되었다고 가정하면 상기 각 노트에 상응하는 모든 시작지점에서 상기 허밍 질의가 매칭되어야 비로소 상기 허밍 질의를 기반으로 해당 미디파일을 정확하게 검색하여 사용자에게 제공할 수 있다.
- [0058] 다음으로 상기 허밍 질의 시스템(100)은 상기 쪼개한 피치벡터에 스펙트럴 해싱을 적용하여 k 개의 고유함수, $\Phi_1, \dots, \Phi_k$ 을 찾는다(S130).
- [0059] 또한 상기 스펙트럴 해싱은 총 3단계로 수행되며, 제 1단계는 PCA(Principal Component Analysis)를 이용하여 상기 쪼개된 피치벡터들의 주성분을 찾는다.
- [0060] 또한 제 2단계는 모든 PCA의 방향에 따라 직사각형 근사 $[a, b]$ 를 이용하여 다음의 [수학식 4]에 따라 가장 작은 고유값 k 에 대응하는 가중된 라플라스 연산자 L_p 의 단일 차원 분석적인 고유함수들을 추출한다.
- [0061] [수학식 4]
- $$\Phi_i(x) = \sin\left(\frac{\pi}{2} + \frac{i\pi}{b-a}x\right),$$
- $$\lambda_i = 1 - e^{-(\epsilon^2/2)|i\pi/(b-a)|^2}$$
- [0062]
- [0063] 여기서 상기 $[a, b]$ 는 상기 PCA 방향(direction)에 따라 투사된 범위를 나타내며, 상기 파라미터 ϵ 는 유사한 이웃들의 반경을 정의하고 상기 λ_i 는 고유값을 나타낸다.
- [0064] 또한 제 3단계는 상기 2단계에서 계산된 모든 고유값에 따라 상기 k 의 가장 작은 값을 찾으며, 제로에서 대응되는 고유함수들을 쓰레싱(Thresholding)함으로써, 이진 코드들을 얻는다.
- [0065] 한편 상기 스펙트럴 해싱에서는 큰 편차와 낮은 스페이스 주파수를 갖는 프로젝트이 바람직하며, 이에 따라 동일한 PCA 방향들은 다른 쓰레싱 방식으로 여러번 선택되어 질 수 있다. 어느 특정 상황에서는 각 PCA 방향은

한번만 포착되어 질 수 있으며 이 때 상기 스펙트럴 해싱은 PCA에 의한 해싱과 유사해 질 수 있다.

[0066] 다음으로 상기 허밍 질의 시스템(100)은 상기 k개의 고유함수로부터 K개의 고유함수를 랜덤하게 선택하여 (S140), K-비트 이진 임베딩 스페이스를 구성한다(S150).

[0067] 또한 상기 허밍 질의 시스템(100)은 상기 제 2단계에서 스펙트럴 해싱을 통해 찾은 k개의 고유함수에 대해서 다시 K개의 고유함수를 랜덤 선택하여 상기 K-비트 이진 임베딩 스페이스를 구성하고, 상기 허밍 질의 시스템(100)은 다중 스펙트럴 해싱을 적용하여 상기 k개의 고유함수에 대해서 다시 복수회에 걸쳐 K개의 고유함수를 연속적으로 선택하여 상기 다중 이진 임베딩 스페이스를 구성한다. 한편 상기 K개의 고유함수를 랜덤하게 선택할 때마다(즉, 상기 K개의 고유함수를 선택하는 회차 마다) 이진 임베딩 스페이스가 구성된다.

[0068] 또한 상기 다중 스펙트럴 해싱에서 최선의 선택을 하기 위해서 최소 고유값K에 대응하는 고유함수의 선택에 의한 이진 임베딩 스페이스가 포함되도록 하는 것이 좋다.

[0069] 다음으로 상기 허밍 질의 시스템(100)은 상기 구성한 각각의 임베딩 스페이스에서 상기 허밍 질의 Q의 스케일된 버전의 Q_u 에 대한 5개의 최근접 이웃(Nearest Neighbor)을 검색한다(S160). 즉, 상기 허밍 질의 시스템(100)은 상기 생성된 각 바이너리 임베딩 스페이스에 대해서 스케일된 허밍 질의로 미리 정해진 수의 인접한 이웃을 탐색한다. 한편 본 발명을 상세히 설명하기 위해 5개의 인접한 이웃을 탐색하는 것으로 도 1에 도시하고 있으나 상기 개수는 사용자의 설정에 따라 달라질 수 있음은 당연하다.

[0070] 다음으로 상기 허밍 질의 시스템(100)은 상기 미디어파일의 넘버와 스타팅 포인트를 포함하는 후보 집단의 리스트를 생성한다(S170).

[0071] 한편 상기 허밍 질의 시스템(100)은 상기 사용자의 허밍 질의가 입력되면 해당 허밍 질의를 피치벡터로 변환하고 적어도 하나 이상의 스케일된 버전의 허밍 질의에 대한 피치벡터를 생성한다.

[0072] 또한 상기 허밍 질의 시스템(100)은 d의 길이를 가지는 상기 허밍 질의의 피치벡터를 상기 각각의 이진 임베딩 스페이스에 매핑하여 각각의 이진 임베딩 스페이스로부터 인접한 해밍거리의 적어도 하나 이상의 후보들을 선택하여 후보 집단 리스트를 생성하고 리턴 한다.

[0073] 상기 적어도 하나 이상의 후보 집단 리스트는 해당 미디어파일의 넘버와 노트의 시작점 포인트를 포함한다.

[0074] 다음으로 상기 허밍 질의 시스템(100)은 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 허밍질의 Q와 상기 선택한 후보들에 대해 유사도 매칭을 수행한다(S180).

[0075] 한편 상기 OEDTW를 적용한 종래의 허밍 질의 기반 음악 검색 시스템은, 상기 사용자로부터 입력되는 허밍 질의와 대상 미디어 파일의 시작부분과 매칭될 수 있도록 함으로써, 글로벌 템포 변화와 로컬 템포 변화를 모두 해결하지만 상기 OEDTW의 수행을 완료하기 전에는 상기 템포 변화를 보완하기 위한 최적의 글로벌 스케일을 결정할 수 없으며, 또한 상기 키 변화를 보완하기 위해 허밍 질의에 대한 시퀀스의 평균을 이동시켜 가면서 상기 OEDTW를 반복적으로 수행하기 때문에 시간적으로 매우 큰 비용을 치르는 문제점이 있다.

[0076] 이에 따라 본 발명인 상기 허밍 질의 시스템(100)은 상기 허밍 질의에 대해 선형 스케일링을 수행하여 각 대상 미디어파일의 상기 허밍 질의에 대한 최적의 글로벌 스케일을 찾고 상기 스케일된 오픈-엔드 동적 타임 워핑 수행 전에 상기 얻은 최적의 스케일을 상기 허밍 질의와 대상 미디어파일의 키 이동에 적용한다.

[0077] 예를 들어, $Q = q_1, \dots, q_m$ 와 $C = c_1, \dots, c_n$ 를 허밍 질의와 미디어파일이라고 가정하면 다음의 [수학식 5]에 따라 스케일링 벡터 u에 의해 스케일된 버전의 허밍 질의 $Q_u = q_1^*, \dots, q_{m_u}^*$ 를 얻을 수 있다.

[0078] [수학식 5]

$$m_u = \lfloor m \times u \rfloor$$

$$q_j^* = q_{\lfloor 1/u \rfloor}, \quad 1 \leq j \leq m_u$$

[0079]

[0080] 또한 $C_{m_u} = c_1, \dots, c_{m_u}$ 는 상기 C로부터 상기 m_u 요소들만큼 취한 시퀀스를 나타내며 $\hat{Q}_u = \hat{q}_1^*, \dots, \hat{q}_{m_u}^*$ 와

상기 $\hat{C}_{m_u} = \hat{c}_1, \dots, \hat{c}_{m_u}$ 는 상기 Q_u 와 C_{m_u} 으로부터 제로평균을 가지기 위해 이동한 시퀀스를 나타낸다.

또한 상기 $\hat{Q}_u$ 와 $\hat{C}_{m_u}$ 사이의 거리는 다음의 [수학식 6]에 의해 계산되어 진다.

[수학식 6]

$$dis(\hat{Q}_u, \hat{C}_{m_u}) = \frac{\sum_{i=1}^{m_u} |\hat{q}_i^* - \hat{c}_i|}{m_u}$$

또한 최소 거리를 제공하는 최적의 스케일링 팩터 s는 다음의 [수학식 7]에 의해 구할 수 있다.

[수학식 7]

$$s = \underset{u}{\operatorname{argmin}} dis(\hat{Q}_u, \hat{C}_{m_u})$$

또한 상기 스케일된 오픈-엔드 동적 타임 워핑은 상기 최적의 스케일링 팩터 s가 적용된 허밍 질의 Q에 대한 Q_s 와 상기 C를 Q_s 의 길이 m_s 의 t=1.4배 절단하며, 상기 절단한 결과는 $C_{t \cdot m_s}$ 로 나타난다.

한편 일반적으로 0.5와 2사이의 스케일링 팩터에 의해 스케일링이 잘 작동된다고 알려져 있으며 본 발명에서는 1.4 스케일링 팩터를 사용한다.

또한 키 쉬프팅을 위해, 상기 $C_{t \cdot m_s}$ 의 시작점으로부터 m_s 번째 피치까지의 평균은 상기 $C_{t \cdot m_s}$ 를 이동시키는데 사용되어지고, 그 결과는 $\tilde{C}_{t \cdot m_s}$ 로 나타난다.

이에 따라 다음의 [수학식 8]과 같이 상기 Q와 C에 적용된 스케일된 오픈-엔드 동적 타임 워핑을 통해 거리를 측정하여 유사도 매칭을 수행할 수 있다.

[수학식 8]

$$dis_{SOEDTW}(Q, C) = \frac{dis_{OEDTW}(\hat{Q}_s, \tilde{C}_{t \cdot m_s})}{m_s}$$

또한 상기 m_s 로 상기 $dis_{OEDTW}(\hat{Q}_s, \tilde{C}_{t \cdot m_s})$ 을 나눔으로써 얻어지는 최종 거리는 상기 허밍 질의 압축 또는 확장을 통해 생성되는 왜곡을 제거하기 위해서이다.

다음으로 상기 허밍 질의 시스템(100)은 상기 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 허밍 질의 Q와 상기 생성한 후보들에 대한 유사도 매칭을 수행하고(S180), 상기 유사도 매칭 결과 유사도가 높은 순서대로 상기 후보들의 랭킹을 측정하고 상위 랭킹에 해당하는 적어도 하나 이상의 후보를 리턴 한다(S190).

도 2는 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템에 있어서, 미디어파일에 대한 데이터 프로세싱을 설명하기 위한 예시도이다.

도 2에 도시한 바와 같이 상기 미디어파일은 노트 시퀀스에 대한 정보를 가지고 있거나 상기 허밍 질의 시스템(100)에 의해 노트 시퀀스로 변환될 수 있다.

한편 상기 노트 시퀀스는 피치와 해당 피치의 지속시간으로 구성된 복수의 노트로 구성된다.

또한 상기 허밍 질의 시스템(100)은 상기 미디어파일 데이터베이스(200)에 저장된 모든 미디어파일의 노트 시퀀스에 대한 각각의 노트를 샘플링 비율에 따라 상기 피치의 지속시간에 대응하는 피치값의 개수로 전환함으로써, 상기

노트 시퀀스를 피치벡터로 변환하고 상기 DB1(300)에 저장한다.

- [0099] 또한 상기 허밍 질의 시스템은(100) 상기 각 노트의 피치값의 시작점을 기준으로 특정 길이만큼 상기 피치벡터를 쉼핑하여 상기 피치벡터의 서브시퀀스를 생성한다.
- [0100] 또한 상기 피치벡터를 쉼핑하는 이유는 상기 입력되는 사용자의 허밍 질의가 해당 미디어파일의 시작 또는 중간 등 어디에서든지 시작될 수 있기 때문이다. 이에 따라 상기 미디어파일의 각 노트에 대응하는 위치에서 상기 피치벡터의 모든 서브시퀀스의 시작점과 매칭하여 더욱 정확한 미디어파일을 검색하기 위해서이다.
- [0101] 도 3은 본 발명의 일 실시예에 따른 다중 바이너리 임베딩 스페이스를 구성하는 방법을 설명하기 위한 예시도이다.
- [0102] 도 3에 도시한 바와 같이 상기 허밍 질의 시스템(100)에 입력되는 사용자의 허밍 질의를 기반으로 미디어파일을 검색하기 위해 다중 바이너리 임베딩 스페이스를 구성하는 방법은 우선, 상기 미디어파일 데이터베이스(200)에 저장된 모든 미디어파일을 피치벡터로 변환하여 상기 DB1(300)에 저장한다.
- [0103] 다음으로 상기 피치벡터를 구성하는 각 노트의 시작점으로부터 길이 d만큼 쉼핑하여 DB2(400)에 저장한다.
- [0104] 다음으로 상기 쉼핑된 피치벡터에 스펙트럴 해싱을 적용하여 k개의 고유함수를 추출한다.
- [0105] 다음으로 상기 추출한 k개의 고유함수에 있어서, 최소 고유값 K에 대응하는 복수의 고유함수에 대해서 다시 복수회에 걸쳐서 복수개의 고유함수를 랜덤하게 선택하여 K-비트 임베딩 스페이스를 생성한다.
- [0106] 도 4는 본 발명의 일 실시예에 따른 입력된 허밍 질의를 기반으로 미디어파일을 검색하는 방법을 설명하기 위한 예시도이다.
- [0107] 도 4에 도시한 바와 같이 상기 허밍 질의 시스템(100)을 통해 입력되는 사용자의 허밍 질의를 기반으로 해당 미디어파일을 검색하는 방법은 우선, 상기 허밍 질의를 피치벡터 Q로 변환한다.
- [0108] 다음으로 상기 변환한 허밍 질의의 피치벡터 Q를 스케일링하여 적어도 하나 이상의 스케일링된 허밍 질의의 피치벡터 Q_s를 생성한다.
- [0109] 다음으로 상기 생성한 각각의 이진 임베딩 스페이스로부터 상기 Q_s의 미리 정해진 수의 최근접 이웃을 탐색한다. 즉, 상기 각각의 이진 임베딩 스페이스에 매핑되어 있는 것은 미디어파일로부터 쉼핑된 피치벡터들이므로 상기 최근접 이웃은 쉼핑된 피치벡터를 말하며, 상기 생성한 각 바이너리 임베딩 스페이스에 대해서 상기 Q_s로 미리 정해진 수의 인접한 쉼핑된 피치벡터들을 탐색한다.
- [0110] 다음으로 상기 각 탐색한 이웃으로부터 미디어파일 번호와 해당 미디어파일에 대한 노트의 시작점을 포함하는 후보군 리스트를 생성한다.
- [0111] 다음으로 스케일된 오픈-엔드 동적 타임 워핑을 적용하여 상기 [수학식 8]에 따라 상기 허밍 질의 Q와 상기 생성한 리스트의 후보군과 각각 유사도 매칭을 수행한다.
- [0112] 다음으로 상기 수행결과 유사도가 높은 순으로 후보군에 해당하는 미디어파일의 랭킹을 리턴 한다.
- [0113] 도 5는 본 발명의 일 실시예에 따른 허밍 질의 방법 및 시스템과 종래의 기술에 대한 성능을 비교한 예시도이다.
- [0114] 또한 본 발명인 허밍 질의 방법 및 시스템과 종래의 기술에 대한 성능은 MRR(Mean Reciprocal Rank, MRR)과 Top-X hit rate를 사용하여 평가하였다. 상기 MRR은 다음의 [수학식 9]에 의해 계산된다.
- [0115] [수학식 9]

$$MRR = \frac{1}{I} \sum_{i=1}^I \begin{cases} 1/rank_i & \text{if } rank_i \leq 20 \\ 0 & \text{if } rank_i > 20 \end{cases}$$

- [0116]
- [0117] 여기서 상기 I은 입력된 허밍 질의의 수를 나타내며, 상기 rank_i는 i-번째 의 허밍 질의를 위해 검색된 후보군 리스트에서 미디어파일의 랭크를 나타낸다.

- [0118] 또한 상기 rank_i 는 상기 미디어파일이 상위 20위 이내에 랭크될 때에만 상기 성능 비교에 포함된다.
- [0119] 또한 상기 MRR은 범위 [0, 1]을 가지며, 모든 입력 허밍 질의에 대한 미디어 파일이 첫 번째 플레이스에 랭크되었을 때 상기 MRR은 1의 값을 가지며, 상기 Top-X hit rate는 순위 X 이내에 랭크되는 미디어파일의 비율을 나타낸다.
- [0120] 본 발명에서는 두 개의 해싱 방법(LSH와 MSH)과 두 개의 유사도 매칭방법(SOEDTW와 RA)의 결합으로 구성된 4개의 허밍 질의 음악 검색 방법의 성능을 비교하였으며, 도 5는 상기 비교한 방법들의 MRR과 Top-X hit rate를 보여준다.
- [0121] 또한 스케일링은 값 0.1, 스케일링 팩터 0.6에서 1.6으로 수행되었으며, 다중 스펙트럴 해싱을 위한 임베딩 스페이스의 수는 20, 40, 60으로 세팅하였고 각 공간 당 비트 수는 4, 8, 16, 32로 세팅하였다.
- [0122] 또한 종래의 기술인 LSH를 위한 테이블의 수는 5, 10, 20, 40, 60이고, 비트 수는 4, 8, 16, 32로 세팅하였다.
- [0123] 또한 특정 인터벌(Interval) 폭(width) r 대신에 함수
$$h_i(x) = w_i^T x + b_i$$
에 의해 구획한 프로젝션 범위로부터 얻은 인터벌의 값 2, 4, 8, 16 또는 24로 셋팅 하였다.
- [0124] 또한 상기 성능 비교에 있어서, 상기 다중 스펙트럴 해싱 또는 LSH에 의해 검색된 후보군에 상기 스케일된 오픈-엔드 동적 타임 워핑을 적용할 때 $\eta = 0$ 으로 설정되어 사용하였으며, 최선의 경우를 위해 상기 $\eta = 1, 2, 3, 4$ 로 추가적으로 설정하여 상기 스케일된 오픈-엔드 동적 타임 워핑을 적용하였다.
- [0125] 또한 RA(Recursive Alignment)의 경우, 분할을 위한 스케일 페어(pair)의 셋을 $\{(1,2), (37,63), (5,7), (45,55), (1,1), (55,45), (7,5), (63,37), (2,1)\}$ 설정하고, 최대 재귀 깊이는 4로 설정하였다.
- [0126] 한편 도 5의 (a)와 (b)는 장(Jang)의 데이터 셋을 사용하여 본 발명과 종래의 기술에 대한 성능을 비교한 것이며, 장의 데이터 셋은 48개의 미디어 파일과 4431개의 허밍 질의의 파일로 구성되어 있다("http://www.music-ir.org/mirex/2010/index.php/query-by-singing/humming..." 참고).
- [0127] 도 5의 (a)는 상기 허밍 질의는 미디어파일의 시작점으로부터 시작되는 것으로 가정하여 그 성능을 비교한 것이며, 도 5의 (b)는 상기 장의 데이터 셋에서 상기 입력되는 허밍 질의가 상기 미디어파일의 시작점으로부터 시작된다는 것을 모르는 것으로 가정하여 그 성능을 비교한 것이다. 도 5의 (c)는 Think IT 코퍼스에서 제공하는 데이터 셋을 이용하여 그 성능을 비교한 것이다.
- [0128] 한편 상기 Think IT 코퍼스에서 제공하는 데이터 셋은 106개의 미디어파일과 355개의 허밍 질의 파일로 구성되어 있으며, 상기 허밍 질의 파일은 상기 미디어파일의 어느 부분에서든지 시작되며, 그 길이는 12초로 구성되어 있다.
- [0129] 도 5에 도시한 바와 같이 다중 스펙트럴 해싱과 스케일된 오픈-엔드 동적 타임 워핑을 적용한 본 발명이 종래의 기술보다 우수한 성능을 나타내고 있다.
- [0130] 도 6은 본 발명의 일 실시예에 따른 허밍 질의 시스템의 구성을 나타낸 블록도이다.
- [0131] 도 6에 도시한 바와 같이 상기 허밍 질의 시스템(100)은 상기 허밍 질의 시스템(100)에 입력되는 상기 미디어파일을 프로세싱하는 데이터 프로세싱부(110), 상기 프로세싱한 미디어파일에 다중 스펙트럴 해싱을 적용하는 다중 스펙트럴 해싱부(120) 및 상기 허밍 질의와 상기 미디어파일의 유사도를 매칭하는 유사도 매칭부(130)를 포함하여 구성된다.
- [0132] 또한 상기 허밍 질의 시스템(100)은 미디어파일 데이터베이스(200)에 저장된 모든 미디어파일을 복수의 노트로 구성된 노트 시퀀스로 변환한다.
- [0133] 또한 상기 데이터 프로세싱부(110)는 상기 변환한 노트 시퀀스를 피치벡터로 변환하는 피치벡터 변환부(111) 및 상기 변환한 피치벡터를 표정하는 피치벡터 표정부(112)를 포함하여 구성된다.
- [0134] 또한 상기 피치벡터 변환부(111)는 상기 미디어파일에 대한 노트 시퀀스를 피치벡터로 변환한다. 상기 변환한 피치벡터는 DB1에 저장된다. 다만 허밍 질의를 피치벡터로 변환한 것은 DB1에 저장할 필요가 없다.
- [0135] 한편 상기 노트 시퀀스는 복수의 노트로 구성되며 상기 각각의 노트는 음의 높낮이를 나타내는 피치와 상기 피

치가 지속되는 시간으로 구성되어 있다.

- [0136] 또한 상기 변환한 피치벡터는 특정 길이를 가지는 각 노트에 대해 상기 지속시간에 상응하는 피치값의 수로 구성되며, 상기 길이는 샘플링 비율과 프레임 사이즈로부터 계산된다.
- [0137] 또한 상기 피치벡터 초핑부(112)는 상기 변환한 피치벡터를 각 노트의 시작점을 기준으로 d의 길이로 초핑한다.
- [0138] 또한 상기 각 노트의 시작점으로부터 초핑하는 이유는 상기 허밍 질의가 상기 미디어파일의 어느 부분에서든지 시작될 수 있으며 이는 상기 허밍 질의를 통해 상기 미디어파일을 검색하기 위해서는 상기 허밍 질의가 상기 미디어파일의 어느 부분에서든지 매칭 되어야 하기 때문이다.
- [0139] 또한 상기 다중 스펙트럴 해싱부(120)는 상기 허밍 질의의 피치벡터를 스케일링하는 허밍 질의 스케일링부(121), 상기 초핑한 미디어파일의 피치벡터로부터 고유함수를 추출하는 고유함수 추출부(122), 상기 추출한 고유함수를 기반으로 다중 임베딩 스페이스를 생성하는 임베딩 스페이스 생성부(124) 및 상기 다중 임베딩 스페이스로부터 후보군 리스트를 생성하는 후보군 리스트 생성부(123)를 포함하여 구성된다.
- [0140] 또한 상기 허밍 질의 스케일링부(121)는 상기 허밍 질의의 템포 변화를 보완하기 위해 상기 허밍 질의의 피치벡터를 선형 스케일링하여 적어도 하나 이상의 스케일링된 버전의 허밍질의에 대한 피치벡터를 생성한다.
- [0141] 또한 상기 고유 함수 추출부(122)는 상기 초핑된 미디어파일의 피치벡터를 사용하여 상기 [수학식 4]에 따라 복수의 고유함수를 추출한다.
- [0142] 또한 상기 임베딩 스페이스 생성부(124)는 상기 추출한 복수의 고유함수에 대해서 다시 복수회에 걸쳐 복수의 고유함수를 랜덤하게 선택하여 복수의 비트를 가진 복수의 바이너리 임베딩 스페이스를 생성한다.
- [0143] 또한 상기 후보군 리스트 생성부(123)는 상기 생성된 각 바이너리 임베딩 스페이스에 대해서 스케일링된 허밍 질의에 대한 피치벡터로 미리 정해진 수의 인접한 이웃을 탐색하고, 각 탐색한 미리 정해진 수의 인접한 이웃으로부터 해당 미디어파일의 번호와 상기 노트의 시작점을 포함하는 후보군 리스트를 생성한다.
- [0144] 한편 상기 후보군 리스트 생성부(123)는 해밍 거리 측정 방법을 사용하여 최소 해밍 거리의 후보군을 상기 생성한 각 바이너리 임베딩 스페이스로부터 리턴 하여 후보군 리스트를 생성한다.
- [0145] 또한 상기 유사도 매칭부(130)는 상기 스케일링 오픈-엔드 동적 타임 워핑을 적용하여 상기 허밍 질의와 상기 후보군 리스트의 미디어파일에 대한 유사도를 측정하고, 해당 후보군 리스트에 대한 미디어파일의 랭킹을 리턴 한다.
- [0146] 이상에서 설명하였듯이, 본 발명인 다중 스펙트럴 해싱 및 스케일링 오픈-엔드 동적 타임 워핑에 기반한 허밍 질의 방법 및 시스템은 다중 스펙트럴 해싱과 스케일링 오픈-엔드 동적 타임 워핑을 동시에 적용하여 입력되는 사용자의 허밍 질의의 템포 변동과 키 변동을 보완하고 해당 미디어파일의 어느 부분에서 시작되는지에 상관없이 정확한 검색이 가능한 효과가 있다.
- [0147] 상기에서는 본 발명에 따른 바람직한 실시예를 위주로 상술하였으나, 본 발명의 기술적 사상은 이에 한정되는 것은 아니며 본 발명의 각 구성요소는 동일한 목적 및 효과의 달성을 위하여 본 발명의 기술적 범위 내에서 변경 또는 수정될 수 있을 것이다.
- [0148] 또한, 이상에서는 본 발명의 바람직한 실시 예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특징의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형 실시가 가능한 것은 물론이고, 이러한 변형 실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안 될 것이다.

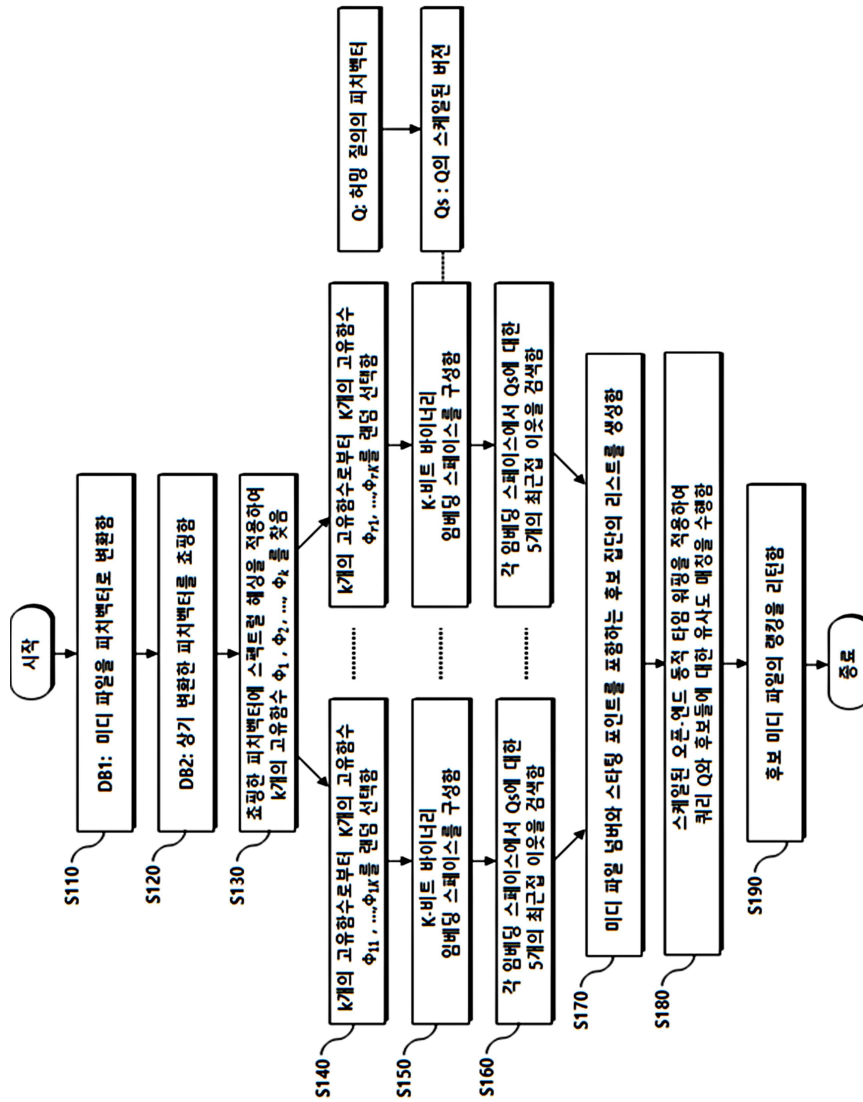
부호의 설명

- [0149]
- | | |
|--------------------|-------------------|
| 100 : 허밍 질의 시스템 | 110 : 데이터 프로세싱부 |
| 111 : 피치벡터 변환부 | 112 : 피치벡터 초핑부 |
| 120 : 다중 스펙트럴 해싱부 | 121 : 허밍 질의 스케일링부 |
| 122 : 고유함수 추출부 | 123 : 후보군 리스트 생성부 |
| 124 : 임베딩 스페이스 생성부 | 130 : 유사도 매칭부 |

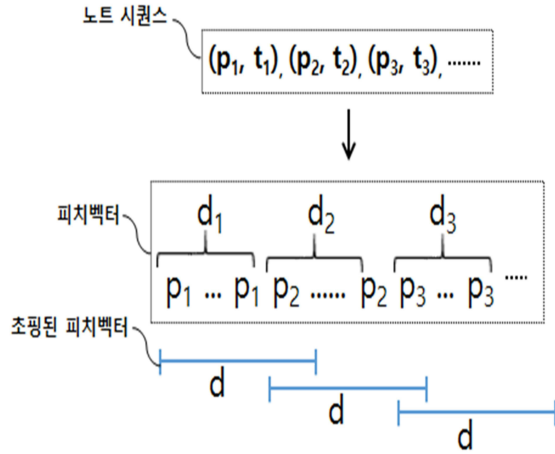
200 : 미디어파일 데이터베이스 300 : DB1
400 : DB2

도면

도면1



도면2



도면3

Phase 1: Construction of multiple binary embedding spaces

1. Construct a database DB1 which is composed of pitch vectors converted from midi files.
2. Construct a database DB2 which is composed of choppings of pitch vectors of DB1 starting at each note with the length d .
3. Find k eigenfunctions $\Phi_1, \dots, \Phi_k$ by Spectral hashing using DB2.
4. Construct K-bits binary embedding space E_1 of DB2 by $\Phi_1, \dots, \Phi_K$ corresponding to the K smallest eigenvalues.
5. for $i = 2, \dots, r$
 6. Select randomly K functions $\Phi_{i1}, \dots, \Phi_{iK}$ from $\Phi_1, \dots, \Phi_k$.
 7. Construct K-bits binary embedding space E_i of DB2 by $\Phi_{i1}, \dots, \Phi_{iK}$.
8. end for

도면4

Phase 2: Retrieval phase for a query vector

1. Let Q be a query pitch vector.
2. for each scaled version Q_s of Q
3. for $i = 1, \dots, r$
4. Search for 5-nearest neighbors of Q_s in the space E_i .
5. Add them to the list of candidates in the form of (midi file number, starting point).
6. end for
7. end for
8. Perform similarity matching of a query Q and the candidates by using SOEDTW in DB1.
9. Return ranking of candidate midi files.

도면5

Methods	top-1	top-5	top-10	top-20	MRR
MSH+SOEDTW	0.942	0.9664	0.9752	0.9828	0.9539
MSH+RA	0.9244	0.961	0.9686	0.9788	0.9412
LSH+SOEDTW	0.842	0.861	0.8675	0.8716	0.8505
LSH+RA	0.835	0.8585	0.8653	0.8702	0.846

(a)

Methods	top-1	top-5	top-10	top-20	MRR
MSH+SOEDTW	0.9025	0.9443	0.9560	0.9661	0.9215
MSH+RA	0.8743	0.9327	0.9456	0.961	0.9
LSH+SOEDTW	0.6928	0.717	0.7238	0.7346	0.7044
LSH+RA	0.6816	0.715	0.7231	0.735	0.6973

(b)

Methods	top-1	top-5	top-10	top-20	MRR
MSH+SOEDTW	0.7746	0.8338	0.8592	0.8789	0.8067
MSH+RA	0.7239	0.8113	0.8394	0.862	0.7666
LSH+SOEDTW	0.5633	0.6113	0.631	0.6451	0.5862
LSH+RA	0.5408	0.5972	0.6254	0.6394	0.5642

(c)

도면6

