

Index-Based Similarity Searching with Partial Sequence Comparison for Query-by-Content Audio Retrieval

Yi Yu, Masami Takata, Kazuki Joe
{yuyi,takata,joe}@ics.nara-wu.ac.jp

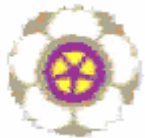
To First Workshop on Semantics of Audio Signals Athens
Greece Dec.03 2006



Nara Women's
University

Outline

- Background and motivation
- Review of ANN based on LSH
- Proposed approach
- Experiments and results
- Conclusions and future work



Audio Retrieval

- Explosion -- audio materials appear on Internet/personal devices
- Access -- music download/upload conveniently
- Search -- retrieval mode is often limited to the text tags
- Necessity -- query-by-content retrieval capabilities is exploited



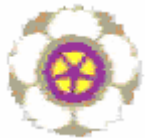
Query-by-Content Audio Retrieval

- Based on spectral similarity retrieval mode is difficult
 - High dimensionality of features
 - Complex computation
 - Large database size
- Scalable searchability need to be exploited
 - The features in service of scalable matching
 - Audio indexing structures



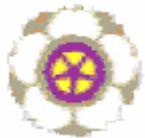
Motivation

- Depend on appropriate metrics and algorithms to reduce pairwise comparisons
- The main challenges:
 - Characterize acoustic objects with relevant spectral features.
 - Represent audio features so that they can be indexed.
 - Locate the desired music segments with a given query within the acceptable time.



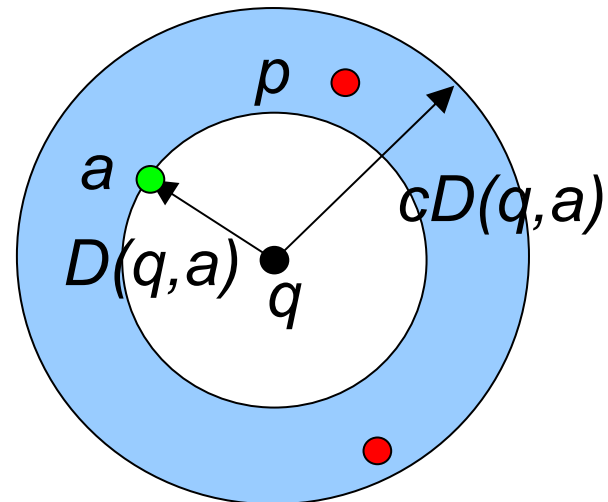
Outline

- Background and motivation
- Review of ANN based on LSH
- Proposed approach
- Experiments and results
- Conclusions and future work



Approximate Nearest Neighbor(ANN)

- **Given** -- a set P of n points in \mathbb{R}^d (d - dimension) and a slackness parameter $\epsilon > 0$
- **Goal** -- with a query point q whose nearest neighbor in P is a , find one/all points p in P , satisfying
$$D(p, q) \leq c D(q, a), \quad c = 1 + \epsilon$$
- Points in the shadowed ring are desired.



Locality-Sensitive Hashing(LSH)

- Purpose -- divide the point set P into subsets
- Assigns a hash value to each subset
 - All points falling into the same subset have the same hash value
- Locality sensitivity
 - The points with a short distance fall into the same subset with a high probability
 - The points with a big distance fall into the same subset with a low probability

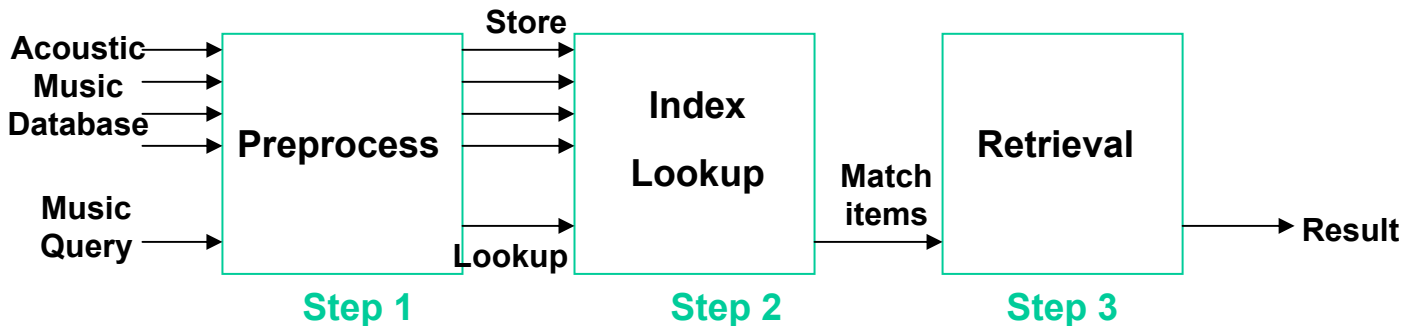


Outline

- Background and motivation
- Review of LSH-based ANN
- **Proposed approach**
- Experiments and results
- Conclusions and future work



Problem Definition

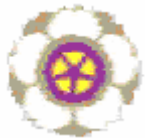


- Step1-- organize database
- Step2 -- lookup the hash table
- Step3 -- compare feature sequences



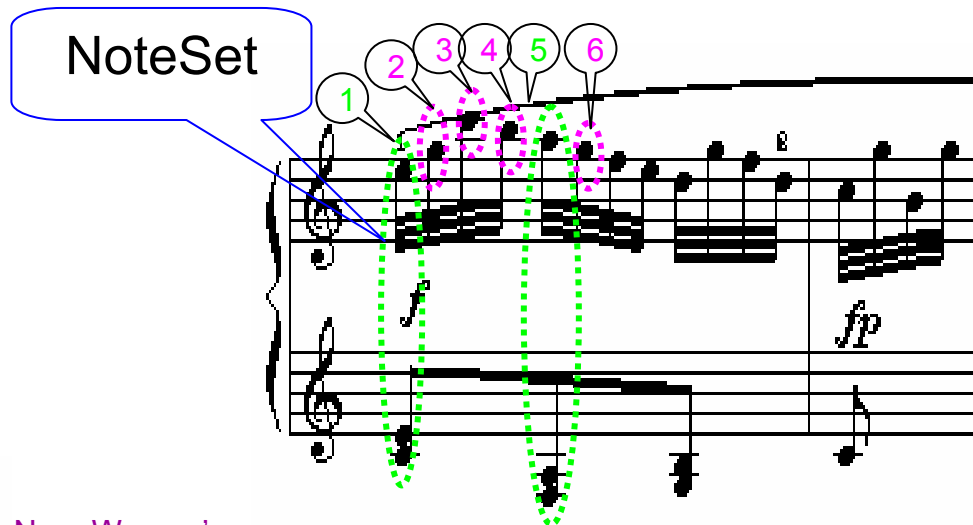
Main Idea

- Match acoustic sequences without comparing a query to each object from the database.
 - a corpus of n musical reference pieces are represented by frames $R = \{r_{i,j} : r_{i,j} \in R_i, 1 \leq i \leq n, 1 \leq j \leq |R_i|\}$
 - $r_{i,j}$ -- the j^{th} spectral feature of the i^{th} reference pieces in a high-dimension space
 - A query sequence q_1, q_2, \dots, q_Q match some resemblances by LSH-based ANN.
 - The resemble features are reorganized and compared with PSC.



Music Score

- Music has strong descriptive composition
 - Music score implies the music theme.
 - Music score is composed of notes.
- For the simplicity of description
 - Define NoteSet as simultaneously initiated notes.



Spectral Analysis I

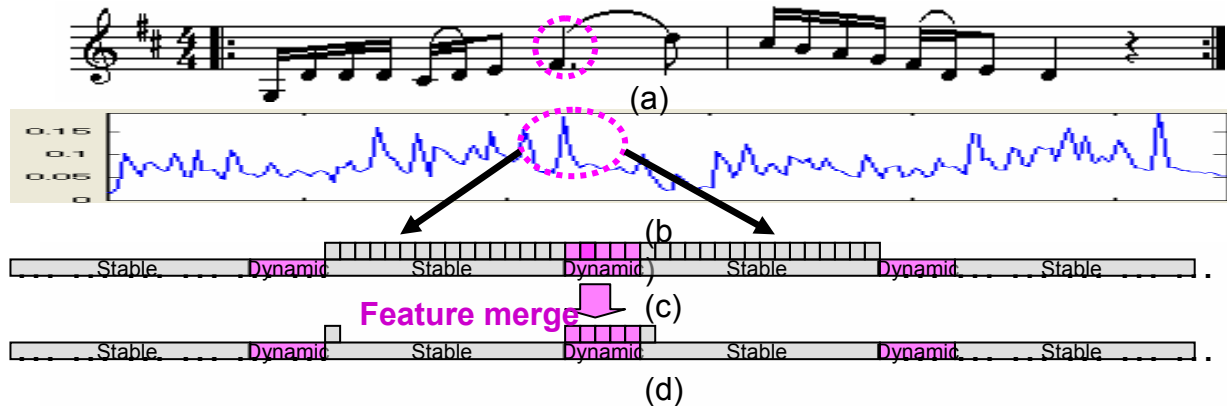
- Adjacent frames can be merged.
- No matter what a NoteSet plays
 - its sound lasts from tens to hundreds of milliseconds
 - it has relatively stable spectral structure
 - spectrum of the adjacent frames are correlated
- A transition
 - happens between adjacent NoteSets
 - is very short



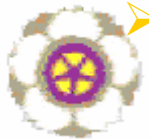
Spectral Analysis II

Example

score
↓
spectrum

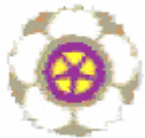


- (a) is the score of Alamuhan (Chinese folk).
- (b) is the corresponding energy profile.
- (c) shows the frames for feature analysis. Within a NoteSet
 - the music has a long stable state & a fixed spectral structure
 - adjacent frames tend to have the same spectral structure
- (c)→(d) is to merge the features
 - the redundant frames are removed
- ideal case, only one frame is necessary to represent a NoteSet



Feature Selection

- Usually Pitch is extracted as the feature
- Here MFCC is adopted as feature because:
 - STFT can stand for spectrum and is associated with music score.
 - STFT can be grouped into spectrum bins according to Mel-frequency scaling filter bank.
 - Spectrum bins can be regarded as rough spectral structure.
 - MFCC is calculated from spectrum bins. So it can represent spectrum structure.



Feature Organization in Database

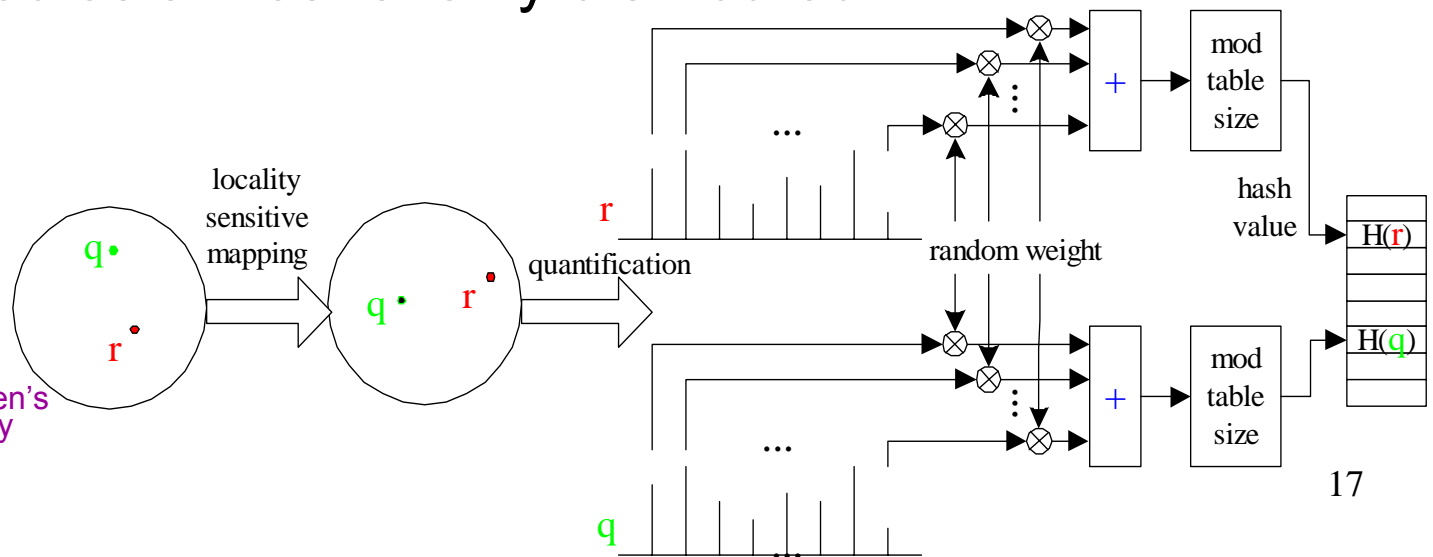
- Aim -- facilitate the access to the database and minimize the search time.
- The basic procedures:
 - Audio sequences are divided into small frames
 - MFCC is calculated and regarded as the feature
 - The feature is quantified and its hash value is calculated.
 - The feature is stored in the bucket according to the hash value.



Single Hash Instance

➤ Hash calculation

- Per-dimensional quantification of the feature
- Two features with a short distance are quantified to the same integer sequences with a high probability
 - generate the same hash value
 - fall into the same bucket
- The random weight makes the features of reference melodies almost evenly distributed.



Parallel Hash Instances

- The increase of hash instances improves hit rate.
- Each hash instance contains all the features.
- Two similar frames collide in at least one of the hash instances.
- Locality sensitive mapping generate different features in each hash instance.
- Different Locality sensitive mapping functions amplify the difference.



Query Stage I

➤ Feature Extraction

- Divide the query into overlapped frames
- Calculate the spectral correlation between adjacent frames
- Merge the adjacent similar frames
- Calculate MFCC for the remaining frames



Query Stage II

➤ Indexing and Post Filtering

- LSH only enables single feature indexing
- Melody retrieval is to find the target with a sequence of features
- With each feature of the query
 - Calculates its hash value in each hash instance
 - The features indexed by the hash values are obtained.
 - The unsimilar frames are removed so as to reduce the post computation.
 - The remaining frames from all hash tables are reorganized.



Query Stage III

- **Our scheme -- Partial Sequence Comparison**
 - Directly utilize the distance calculated in the filtering stage to a DTW table.
- **Why partial sequence comparison ?**
 - The conventional Dynamic Programming is not efficient since most of matching pairs are missing.
 - The remaining matching pairs are sparsely distributed over the Dynamic Time Warping (DTW) table.



Outline

- Background and motivation
- Problem description
- Proposed approach
- **Experiments and results**
- Conclusions and future work



Experiment Setup

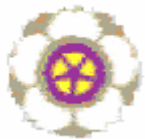
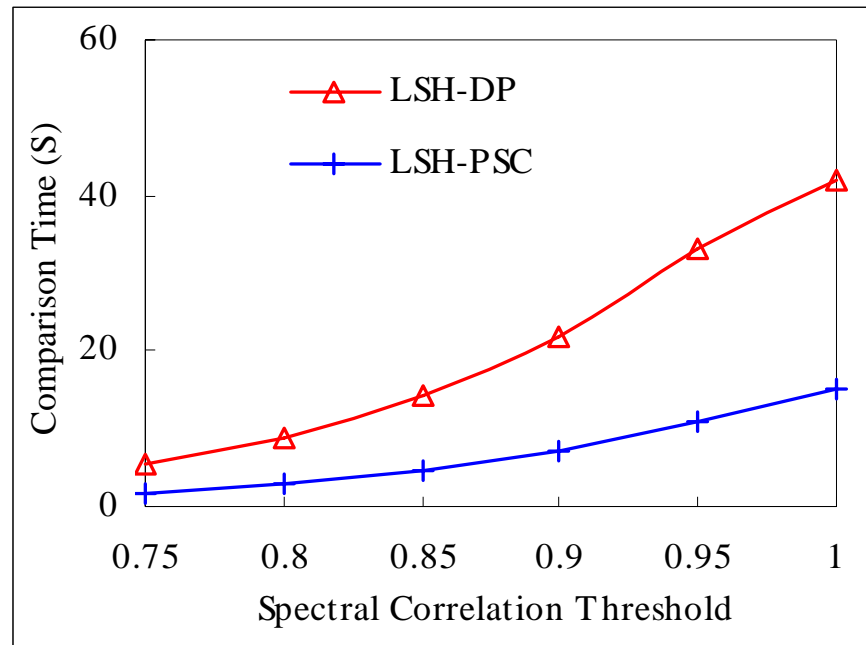
- All polyphonic melodies played by a popular Chinese 12-girl-band
 - Database size: 111 melodies
 - Reference pieces: 60s
 - Query pieces: 8s
 - Sampling rate: 22.05KHz
 - Frame length: 1024
 - Frame overlap: 50%
 - STFT resolution: 11Hz (22050/2048)
- **Experiments goal :**
 - evaluate LSH-DP and LSH-PSC
- **Evaluation metric:**
 - Computation time
 - Retrieval ratio



Sequence Comparison Time

- Under different Spectral Correlation threshold
 - Consider time computation from three aspects:
 - Building the hash table (ahead of the actual retrieval).
 - Filtering the features with LSH
 - Comparing feature sequences
 - Frame merge
 - Reduce both filtering time and sequence comparison time
 - PSC outperforms DP

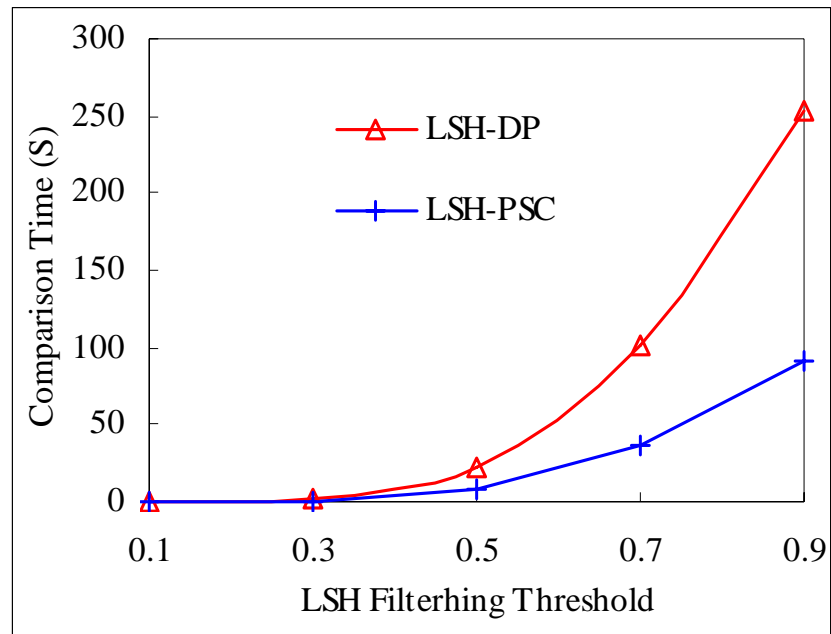
Fig.1 Sequence comparison time in DP and PSC under different SC threshold ρ . $\rho=1.0$ means no frame merge. (3 hash tables, LSH filtering threshold $\delta=0.5$)



Sequence Comparison Time

- Under different LSH filtering threshold
 - Consider computation time from three aspects:
 - Building the hash table (ahead of the actual retrieval).
 - Filtering the features with LSH
 - Comparing feature sequences
 - Filtering in LSH
 - Reduce sequence comparison time
 - PSC outperforms DP

Fig.2 Sequence comparison time in DP and PSC under different filtering threshold δ (3 hash tables, SC threshold=0.9)



Total Retrieval Time

- Computation is involved in three aspects:
 - Building the hash table (ahead of the actual retrieval).
 - Filtering the features with LSH
 - Comparing feature sequences
- Total retrieval time involves filtering and comparison time
 - Most of the time is spent in filtering stage
- Effectiveness of LSH-PSC
 - LSH effectively reduces the number of pairwise comparison
 - PSC further reduces the sequence sequence comparison

Table 1. The total retrieval time consumed for 111 queries under different schemes. For LSH, the filtering time is about 65.4s at $\rho=0.9$ and about 100s at $\rho=1.0$. (3 hash instances, LSH filtering threshold $\delta=0.5$)

Scheme	LSH-DP ($\rho=0.9$)	LSH-DP ($\rho = 1.0$)	LSH-PSC ($\rho = 0.9$)	LSH-PSC ($\rho = 1.0$)	DP ($\rho = 1.0$)
Time(s)	87.6	139.2	72.8	118.7	1519.7



Retrieval Ratio

- Under different Spectral Correlation threshold
 - Frame merge reduces the retrieval time at the cost of retrieval ratio
 - A small SC threshold results in less computation time
 - While the retrieval ratio is also affected.
 - The effect is little when the SC threshold is bigger enough

Table 2. Top-4 retrieval ratio under different SC threshold ρ (3 hash tables, $\delta = 0.5$).

ρ	0.75	0.8	0.85	0.9	0.95	1.0
LSH-DP	0.928	0.955	0.982	0.982	0.991	0.991
LSH-PSC	0.856	0.892	0.937	0.946	0.964	0.982



Retrieval Ratio

- Under different filtering threshold
 - LSH filtering removes the unsimilar features
 - A small filtering threshold reduces the sequence comparison time.
 - At the cost of reduced retrieval ratio.
 - The effect is little when the filtering is bigger enough

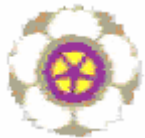
Table 3 Top-4 retrieval ratio under different filtering threshold (3 hash tables, $\rho=0.9$)

δ	0.1	0.3	0.5	0.7	0.9
LSH-DP	0.5945	0.973	0.982	1.0	1.0
LSH-PSC	0.604	0.914	0.946	0.977	0.98



Outline

- Background and motivation
- Problem description
- Proposed approach
- Experiments and results
- **Conclusions and future work**



Conclusions and Future Work

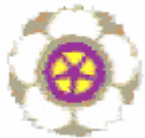
➤ Our contribution

- Index-based query-by-content audio retrieval
 - Efficient organization of audio features
 - Partial sequence comparison
- Effectiveness of proposed algorithms
 - Computation time
 - Retrieval ratio

➤ Future work

- Evaluation of scalability of the proposed scheme with a larger database
- Application of query-by-content audio retrieval in ubiquitous environment.





Nara Women's
University