

Week 10A

Query-by-Humming and Music Fingerprinting

Roger B. Dannenberg


Professor of Computer Science, Art and Music
Carnegie Mellon University



Overview

- Melody-Based Retrieval
- Audio-Score Alignment
- Music Fingerprinting

Metadata-based Retrieval

- Title
 - Artist
 - Genre
 - Year
 - Instrumentation
 - Etc.
- 
- What if we could search by content instead?

3

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Melody-Based Retrieval

- Representations:
 - Pitch sequence (not transposition invariant)
 - Intervals (chromatic or diatonic)
 - Approximate Intervals (unison, seconds, thirds, large)
 - Up/Down/Same: sududdsududdsuddddusddud
- Rhythm can be encoded too:
 - IOI = Inter-onset interval
 - Duration sequences
 - Duration ratio sequences
 - Various quantization schemes

4

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Indexing

- Easily done, given exact, discrete keys*
- Pitch-only index of *incipits***
- Manual / Printed index works if melody is transcribed without error

*here, *key* is used in the CS sense of “*Searching* involves deciding whether a *search key* is present in the *data*” (as opposed to musical keys)

** the initial notes of a musical work

5

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Computer-Based Melodic Search

- Dynamic Programming
- Typical Problem Statement: find the *best match* in a database to a *query*
 - Query is a sequence of pitches
 - “best match” means some substring of some song in the database with minimum edit distance
 - Query does not have to match beginning of song
 - Query does not have to contain entire song

6

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

What Features to Match?



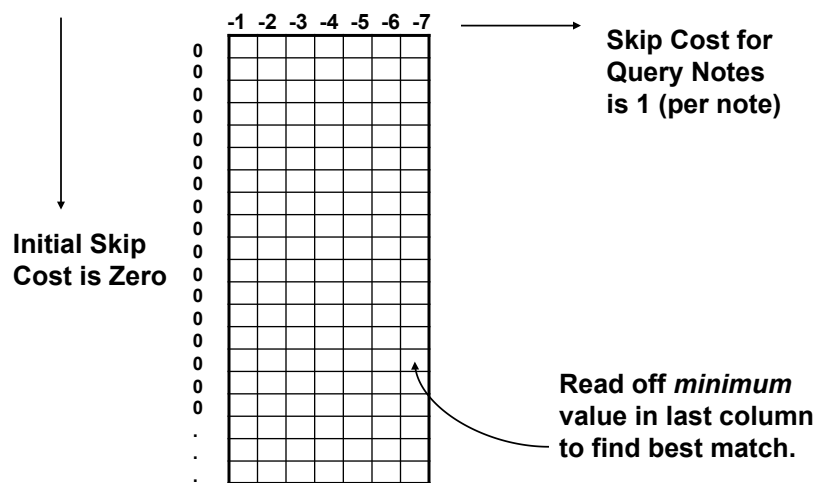
Absolute Pitch:	67	69	71	67
Relative Pitch:		2	2	-4
IOI:	1	0.5	0.5	1
IOI Ratio:		0.5	1	2
Log IOI Ratio:		-1	0	1

7

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Dynamic Programming for Music Retrieval



8

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Example

melody:

key: A G F C
-1 -2 -3 -4

C	0				
D	0				
A	0				
G	0				
E	0				
C	0				
D	0				
G	0				

9

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Example

melody:

key: A G F C
-1 -2 -3 -4

C	0	-1	-2	-3	-2
D	0	-1	-2	-3	-3
A	0	1	0	-1	-2
G	0	0	2	1	0
E	0	-1	1	1	0
C	0	-1	0	0	2
D	0	-1	-1	-1	1
G	0	-1	0	-1	0

Here, rather than classical edit distance, we are computing:
#matches -
#deletions -
#insertions -
#substitutions, so this is a measure of "similarity" rather than "distance": larger is better.

10

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Search Algorithm

- For each melody in database:
 - Compute the best match cost for the query
- Report the melody with the lowest cost
- Linear in size of database and size of query

11

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Themes

- In many projects, themes are entered by hand
- In MUSART, themes are extracted automatically from MIDI files
- Interesting research in its own right
- Colin Meek: themes are patterns that occur most often
 - Encode n-grams as bit strings and sort
 - Add some heuristics to emphasize “interesting” melodic material
 - Validated by comparing to a published thematic index

12

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

How Do We Evaluate Searching?

- Typically there is a match score for each document
- Sort the documents according to scores
- “Percent in top 10”: Count number of “relevant”/correct documents ranked in the top 10
- “Mean Reciprocal Rank”: the mean value of $1/\text{rank}$, where rank is the lowest rank of a “correct” document. 1=perfect, worst $\rightarrow 0$

13

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

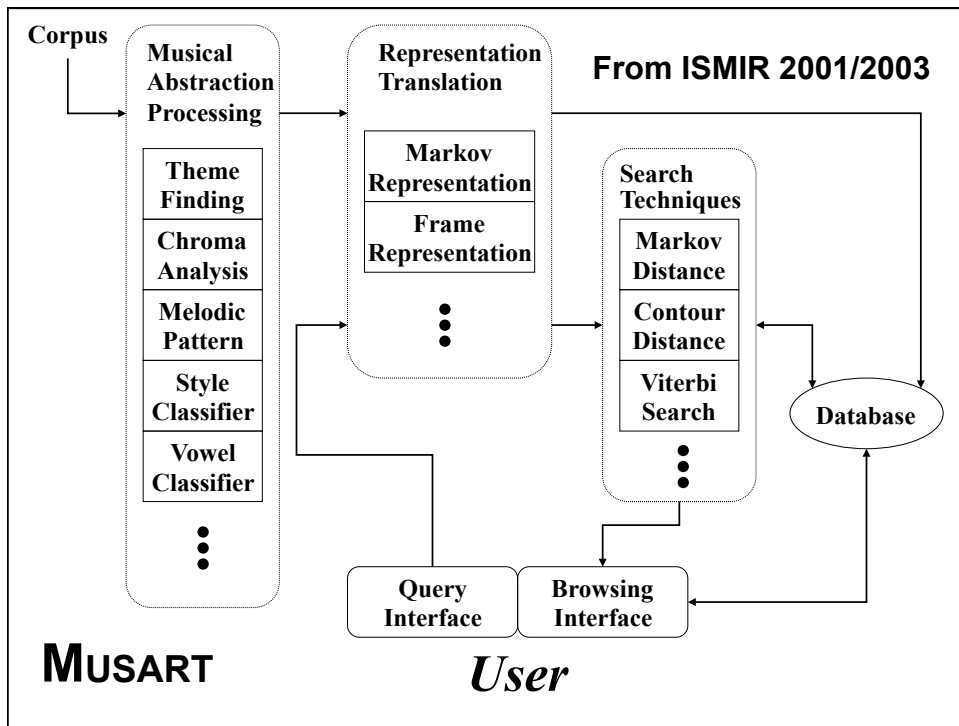
MRR Example

- Test with 5 keys (example only, you really should test with many)
- Each search returns a list of top picks.
- Let's say the correct matches rank #3, #1, #2, #20, and #10 in the lists of top picks
- Reciprocals: $1/3$, $1/1$, $1/2$, $1/20$, $1/10 = 0.33, 1.0, 0.5, 0.05, 0.1$
- Sum = 1.98, divide by 5 $\rightarrow 0.4$
- MRR = 0.4

14

Carnegie Mellon University

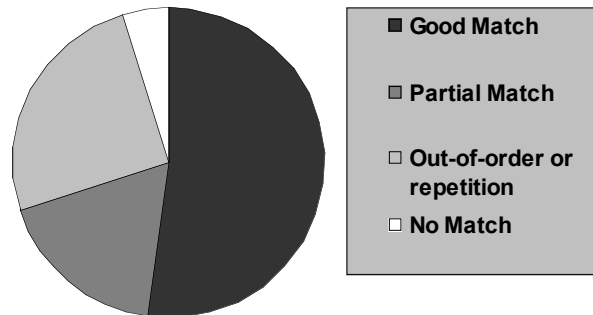
© 2019 by Roger B. Dannenberg



Queries	Databases
High Quality: 160 queries, 2 singers, 10 folk songs	10,000 Folk songs
Beatles (Set #1): 131 queries, 10 singers, 10 Beatles songs	258 Beatles songs (2844 themes)
Popular (Set #2): 165 queries, various popular songs	868 Popular songs (8926 themes)

16 Carnegie Mellon University © 2019 by Roger B. Dannenberg

How good/bad are the queries?



17

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Results

Representations	MRR
Absolute Pitch & IOI	0.0194
Absolute Pitch & IOIR	0.0452
Absolute Pitch & LogIOIR	0.0516
Relative Pitch & IOI	0.1032
Relative Pitch & IOIR	0.1355
Relative Pitch & LogIOIR	0.2323

18

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Insertion/Deletion Costs

$C_{ins} : C_{del}$	MRR	$C_{ins} : C_{del}$	MRR
0.5 : 0.5	0.1290	1.0 : 1.5	0.2000
1.0 : 1.0	0.1484	0.2 : 2.0	0.2194
2.0 : 2.0	0.1613	0.4 : 2.0	0.2323
1.0 : 0.5	0.1161	0.6 : 2.0	0.2323
1.5 : 1.0	0.1355	0.8 : 2.0	0.2258
2.0 : 1.0	0.1290	1.0 : 2.0	0.2129
0.5 : 1.0	0.1742		

Other Possibilities

- Indexing – not robust because of errors
- N-gram indexing – also not very robust
- Dynamic Time Warping
- Hidden Markov Models

N-Grams

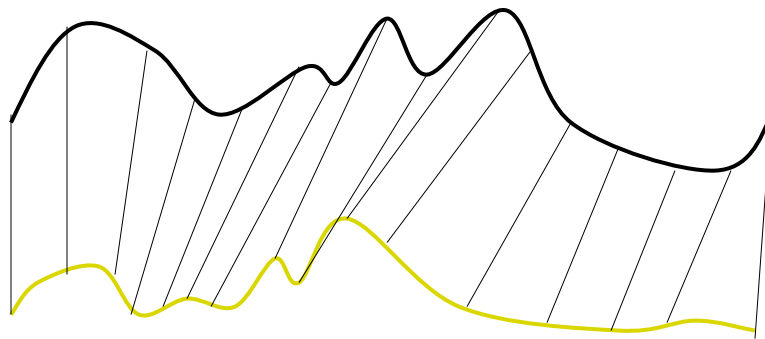
- G G A G C B G G ...
- → GGA, GAG, AGC, GCB, CBG, BGG, ...
- A common text search technique
- Rate documents by number of matches
- Fast search by index (from n-gram to documents containing the n-gram)
- Term Frequency Weighting
 - $tf = \text{count or percentage of occurrences in document}$
- Inverse Document Frequency Weighting
 - $idf = \log(\#docs / \#(docs \text{ with matches}))$
- Does not work well (in our studies) with sung queries due to the high error rates:
 - n-grams are either too short to be specific or
 - n-grams are too long to get exact matches
- Need something with higher precision

21

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Dynamic Time Warping



22

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Dynamic Time Warping (2)

		60.1	60.2	65	64.9	...	Query Data
Target Data	60						
	60						
	65						
	65						
	...						

23

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

DP vs DTW

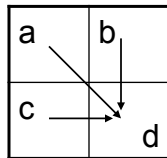
- Dynamic Time Warping (DTW) is a special case of dynamic programming
- (As is the LCS algorithm)
- DTW implies matching or alignment of time-series data that is sampled at equal time intervals
- Has some advantage for melody matching – no need to parse melody into discrete notes

24

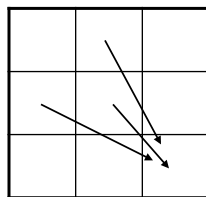
Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Calculation Patterns for DTW



$$d = \max(a, b + \text{deletecost}, c + \text{insertcost}) + \text{distance}$$



The slope of the path is between $\frac{1}{2}$ and 2. This tends to make warping more plausible, but ultimately, you should test on real data rather than speculate about these things. (In our experiments, this really does help for query-by-humming searches.)

25

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Hidden Markov Models

- Queries can have many types of errors:
 - Local pitch errors
 - Modulation errors
 - Local rhythm errors
 - Tempo change errors
 - Insertion and deletion errors
- HMMs can encode errors as states and use current state (error type) to predict what will come next
- Best match is an “explanation” of errors including their probabilities

26

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Dynamic Programming with Probabilities

- What does DP compute? *Path length*, a sum of costs based on mismatches, skips, and deletions.
- Probability of independent events:
$$P(a, b, c) = P(a)P(b)P(c)$$
- So, $\log(P(a, b, c)) = \log(P(a)) + \log(P(b)) + \log(P(c))$
- Therefore, *DP computes the most likely path*, where each branch in the path is independent, and where skip, delete, and match costs represent logs of probabilities.

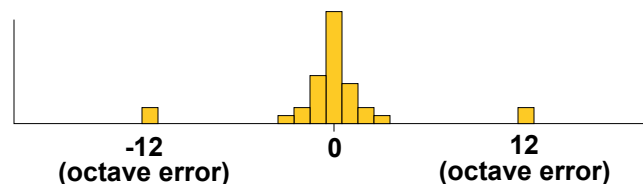
27

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Example for Melodic Matching

- Collect some “typical” vocal queries
- By hand, label the queries with correct pitches (what the singer was *trying* to sing, not what they actually sang)
- Get computer to transcribe the queries
- Construct a histogram of relative pitch error:



- With DP string matching, we added 1 for a match. With this approach, we add $\log(P(\text{interval}))$. Skip and deletion costs are still ad-hoc.

28

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Audio to Score Alignment

Ning Hu, Roger B. Dannenberg and George Tzanetakis

Carnegie Mellon University

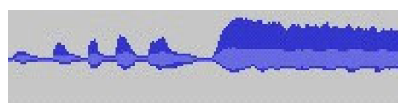


Music Representations

- Symbolic Representation
 - easy to manipulate
 - “flat” performance

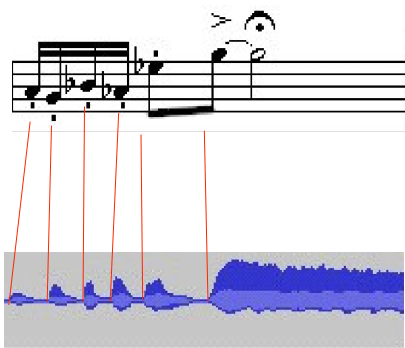


- Audio Representation
 - expressive performance
 - opaque & unstructured



Music Representations

- Align**
- Symbolic Representation
 - easy to manipulate
 - “flat” performance
 - Audio Representation
 - expressive performance
 - opaque & unstructured



31

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Motivation

- Query-by-Humming: find audio file from sung query
- Where do we get a database of melodies (can't extract melody from general audio)?
- Melodies can be extracted from MIDI files
- Can we then match the MIDI files to audio files?

32

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Alignment to Audio

- Related work: please see paper & ISMIR03
- Obtain features from audio and from score
 - Chromagram
 - Pitch Histogram
 - Mel Frequency Cepstral Coefficients (MFCC)
- Use DTW to align feature strings

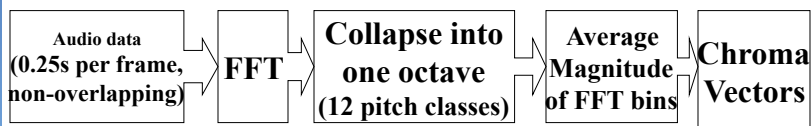
33

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Acoustic Features – Chromagram

- Sequence of 12-element Chroma vectors
- Each element represents spectral energy corresponding to one pitch class (C, C#, D, ...)
- Computing process:



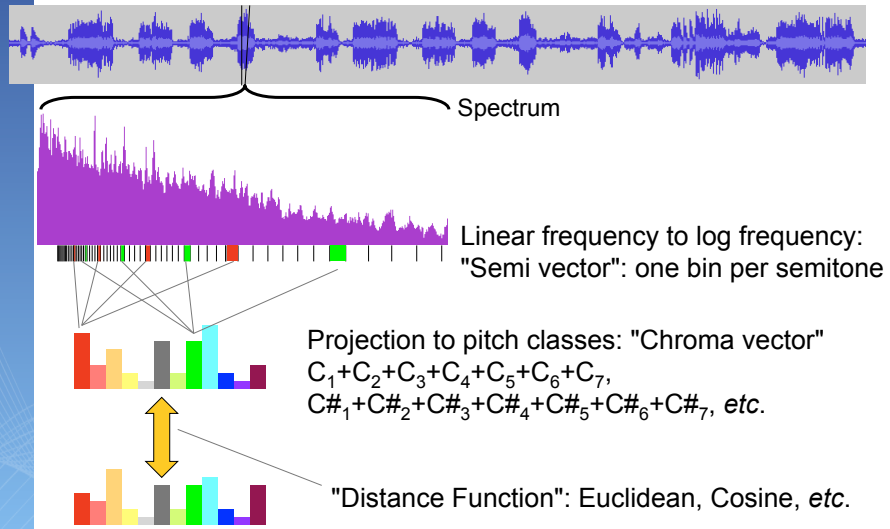
- Advantages:
 - Sensitive to prominent pitches and chords
 - Insensitive to spectral shape

34

Carnegie Mellon University

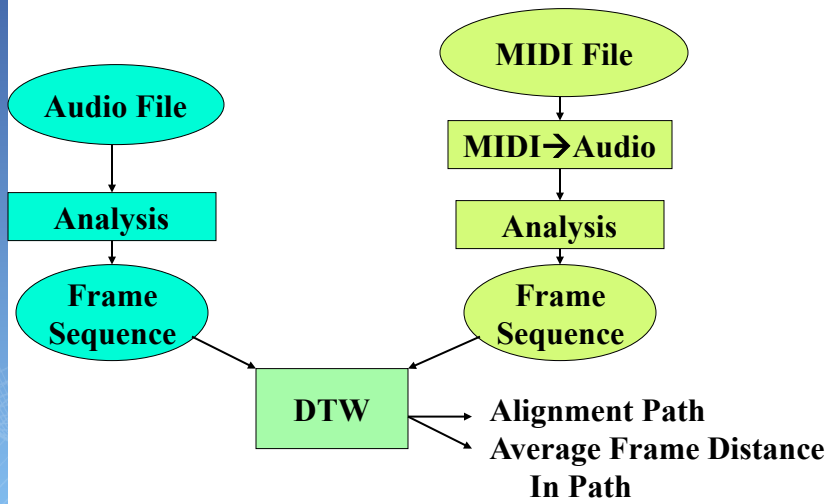
© 2019 by Roger B. Dannenberg

Chromagram Representation



5

Alignment



36

Comparing & Matching Chroma

- Two sequences of chroma vectors
 - Audio from MIDI (using Timidity renderer)
 - Acoustic recording
- Chroma comparison
 - Normalize chroma vectors ($\mu = 0, \sigma = 1$)
 - Calculate Euclidean distance between vectors
 - Distance = 0 \Rightarrow perfect agreement

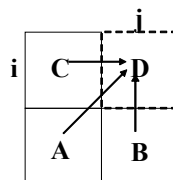
37

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Locate Optimal Alignment Path

- Dynamic Time Warping (DTW) algorithm



$$D = M_{i,j} = \min(A, B, C) + \text{dist}(i, j)$$

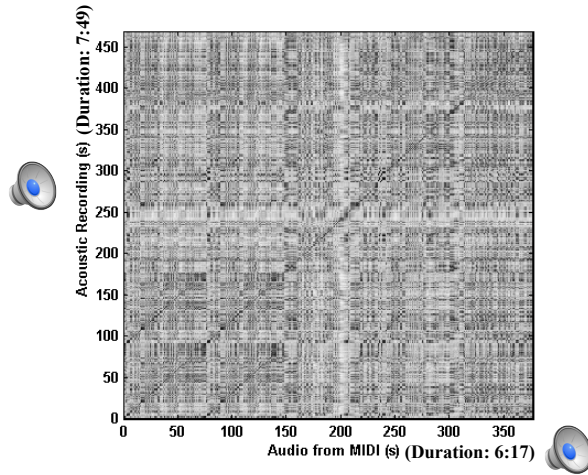
- The calculation pattern for cell (i,j) in the matrix

38

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Similarity Matrix



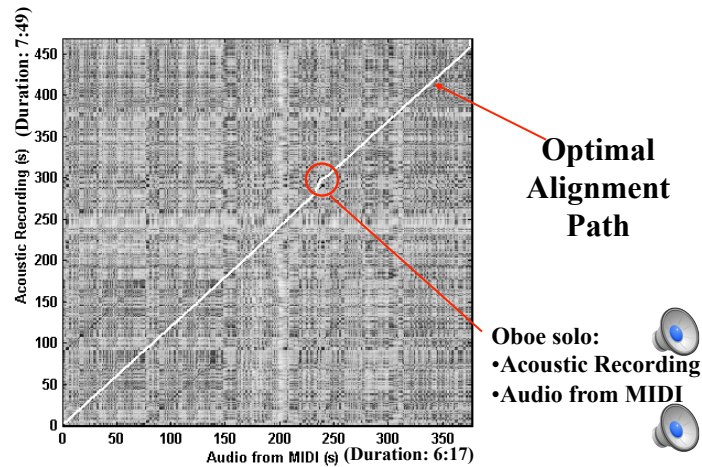
Similarity Matrix for Beethoven's 5th Symphony, first movement

39

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Similarity Matrix



Similarity Matrix for Beethoven's 5th Symphony, first movement

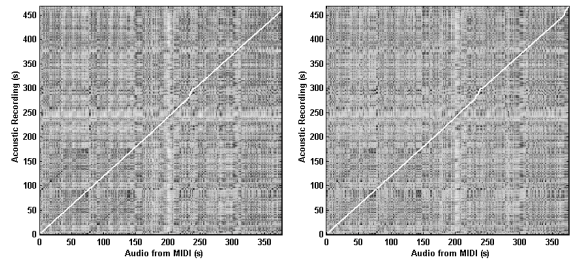
40

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Optimization

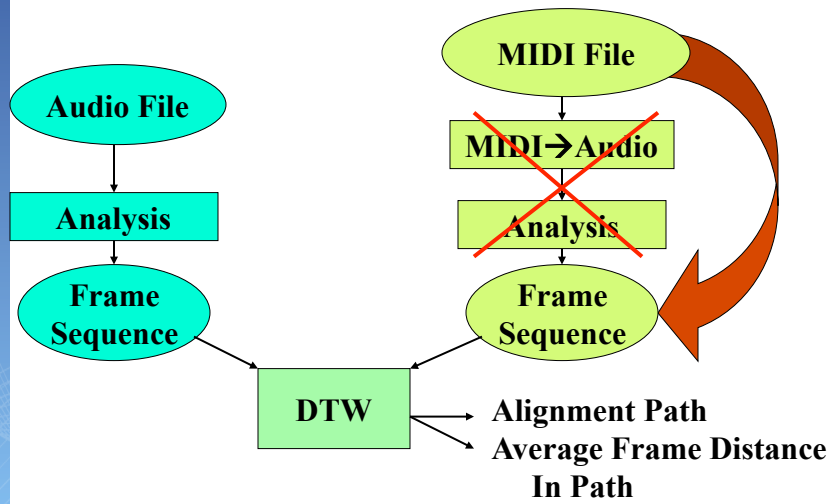
- Chroma is not sensitive to timbre



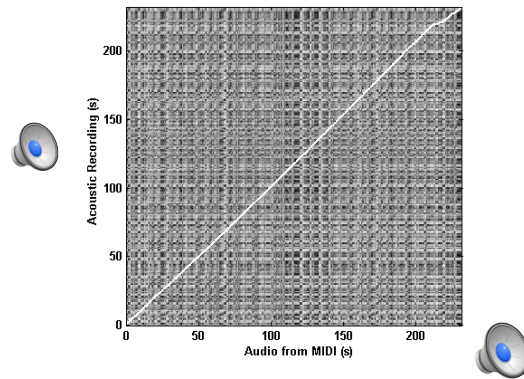
MIDI synthesized using original symphonic instrumentation MIDI synthesized using only piano sound

- Avoid MIDI synthesizing & extracting chroma vectors
 - Map each pitch to a chroma vector
 - Sum vectors & then normalize

Alignment



Alignment Successful Even With Vocals



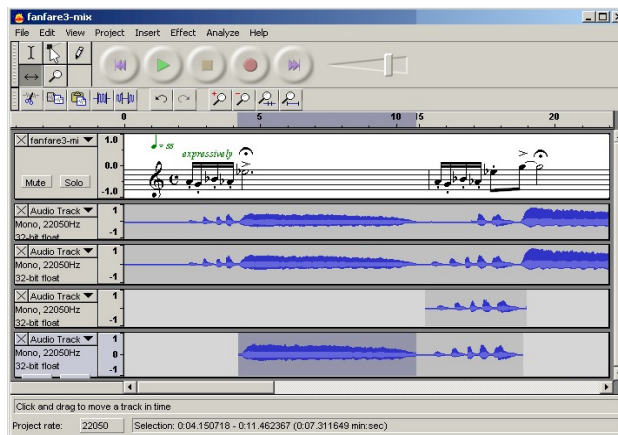
“Let It Be” with vocals matched with MIDI data

43

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Intelligent Audio Editor Mock-up



44

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Summary & Conclusions (on Audio to Score Alignment)

- How to align MIDI to Audio
- Simple computation – no learning, few parameters to tune
- Evaluated several different features
- Investigated searching for MIDI files given audio
- Building a bridge between signal and symbol representations
- In many cases, serves as a replacement for polyphonic music transcription

45

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Music Fingerprinting



photo by Philips



Music Fingerprinting

- Motivation: How do you...
 - ... find the title of a song playing in a club
 - ... or on the radio
 - ... generate playlists from radio broadcasts for royalty distribution
 - ... detect copies of songs
 - ... find original work, given a copy
- Note: recordings and copies have many kinds of distortion and time stretching

47

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Audio Fingerprinting Problem Statement

- Given: a partial copy of a music recording (usually about 10 or 15 seconds),
- with some distortion
 - E.g. cell phone audio
 - Radio stations often shorten songs
- Given: a database of original, high-quality audio
- Find: audio in database that is, with high probability, the original recording

48

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

How It Works (General)

- Find some unique audio features that survive distortion and transformation with high probability
- Build an index from (quantized) features to database
- Search:
 - Calculate (many) features from query
 - Look up matching songs in database
 - Output song(s) with sufficient number of matches

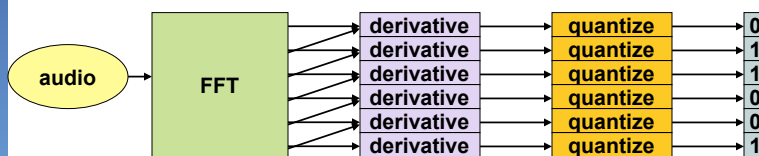
49

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Features: Spectral Flux

- Philips system uses spectral flux:



- Output is stream of 32-bit words
- Each word is indexed
- Search looks for a number of exact matches that indicate a roughly constant time stretch

50

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Comparing Fingerprints

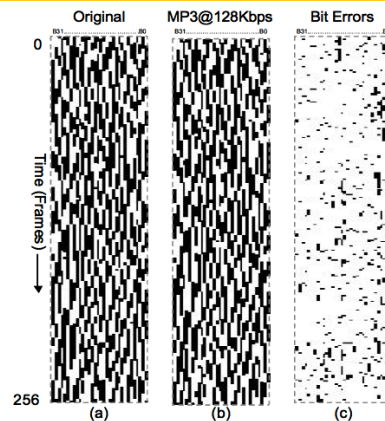


Figure 2. (a) Fingerprint block of original music clip, (b) fingerprint block of a compressed version, (c) the difference between a and b showing the bit errors in black (BER=0.078).

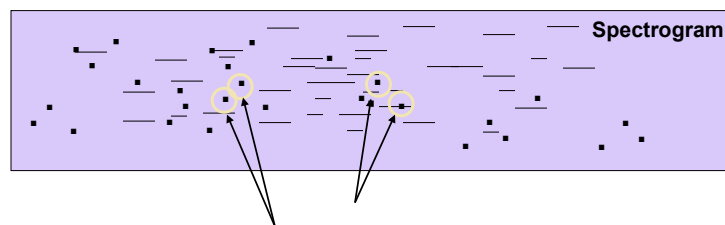
51

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Features: Spectral Peaks

- Shazam uses pairs of spectral peaks:



- Peaks are likely to survive any distortion and time stretch
- Pairs are unique enough to serve as a good index

52

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Performance and Business

- Shazam: >10M tunes, 30s retrieval by cell phone
- Gracenote bought Philips' technology (Gracenote is behind CDDB). Says 28M songs (wikipedia). Mobil Music ID - phone in song to buy matching ringtone.
- NTT and others announced systems in the past
- Echo Nest (bought by Spotify)
- Last.fm

53

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Query-By-Humming with Audio Database

- Problem: given an *audio* database, find songs that match a sung audio query
- So far, extracting melody from audio is quite difficult and error prone.
- QBH with symbolic data is already pretty marginal
- A few systems have been built – SoundHound, Midomi – but results are not nearly as strong as with music fingerprinting

54

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Finding “Covers”

- A cover is a performance of a song by someone other than the “original” artist
- Finding covers in a database, given the original recording is similar to music fingerprinting, but...
- Music Fingerprinting uses distinctive acoustic features,
- Not high-level semantic features that are shared between originals and covers
- Some success matching chromagram features computed at very low (1 second) rates – averages almost all but chord/key change/very prominent melodic material.

55

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Music Information Retrieval Summary

- Query-By-Humming:
 - Techniques
 - String matching techniques
 - Dynamic Time Warping
 - Hidden Markov Models
 - N-Grams
 - Representation is critical
 - Tie DP & DTW to (log) probabilities

56

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Music Information Retrieval Summary (2)

- Audio to Score Matching
 - Chromagram representation is very successful
 - Robust enough for real-world applications now
- Audio Fingerprinting
 - Key is to find robust and distinctive acoustic features
 - Indexing used for fast retrieval
 - Some post processing to select songs with multiple consistent hits
 - Already a big business

57

Carnegie Mellon University

© 2019 by Roger B. Dannenberg

Summary

- Music Fingerprinting works by forming an index of features that are highly reproducible from (re)recorded audio
- Audio-to-Symbolic Music Alignment works well, at least with limited temporal precision
- Other MIR tasks: Query by Humming and Cover Song Detection are much more difficult; no general and robust solutions exist.

58

Carnegie Mellon University

© 2019 by Roger B. Dannenberg