Music Alignment and Applications

Roger B. Dannenberg
Schools of Computer Science, Art, and Music

Introduction

- Music information comes in many forms
  - Digital Audio
  - Multi-track Audio
  - Music Notation
  - MIDI
  - Structured meta-data (e.g. AMG)
  - Unstructured meta-data (e.g. tags, blogs)

Overview

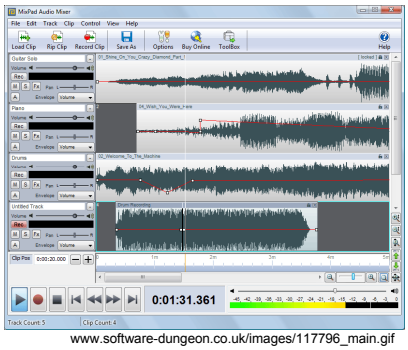
- Music Representations
- Music Alignment
  - Chromagrams
  - Dynamic Programming
- Some Applications
  - Audacity implementation
  - Onset detection

Music Audio

- Millions of files online
- Usually considered the "true" document
  - What people listen to
  - Details at all levels, from composition to signal
- Limitations:
  - Does not contain any explicit abstract information:
    - Notes, chords, rhythm, sections, instrumentation
  - Can't automatically extract a note-level description
  - Source separation problem (unsolved)

Multi-track Music Audio

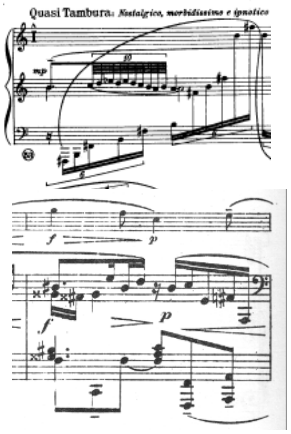
- Most music is recorded on separate "tracks"
  - Stereo has left and right
  - Master (source) recordings typically have "piano" track, "vocal" track, "bass" track, etc.
- Allow studios to manipulate audio in interesting ways without solving the source separation problem.



www.software-dungeon.co.uk/images/117796_main.gif

Music Notation

- Mostly quantized or symbolic representation of music
- "Deep Structure" explicitly denotes much (not all) abstract information
- To derive (musical) audio requires musicians to perform the music



http://www.informatics.indiana.edu/donbyrd/InterestingMusicNotation.html

MIDI

Musical Instrument Digital Interface

- Designed to capture *music keyboard* performance information: key number+velocity, key up, volume pedal, etc.
- Some MIDI files are "quantized" and contain *some* music notation info.
- Usually, instrument info (sound source) is available.
- Convert to audio with synthesis, but usually not great sound.

http://www.les-stooges.org/pascal/midiswing/

Meta-Data and Text

- An interesting topic, but I will not talk about it today.

Linking/Sync'ing Different Representations

- Music alignment is not trivial:
  - Music is somehow "the same" at different speeds
  - Performers are not exact, so no two performances have the same tempo
  - Radio stations typically time-scale recordings to make them shorter(!)
  - Music notation leaves exact timing to performers
  - Performers take liberties with timing for expression, e.g. timing details are important to communicate emotion

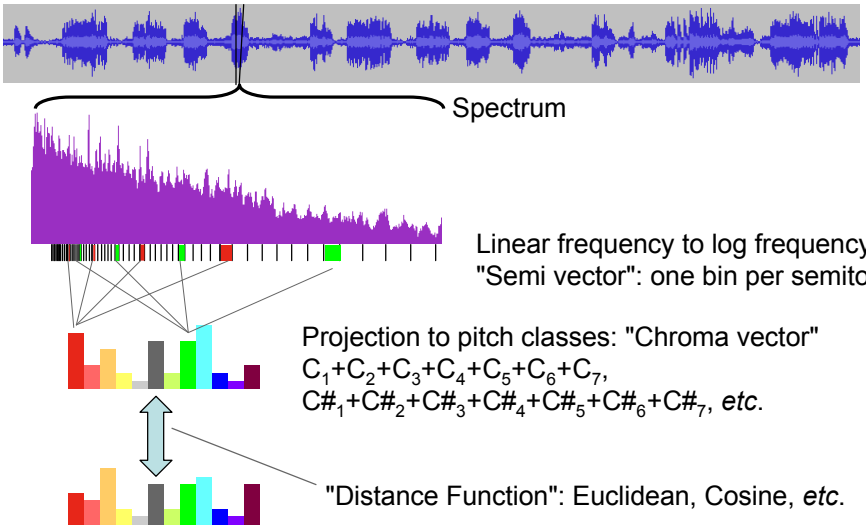
Linking/Sync'ing Different Representations (2)

- Music alignment is interesting:
  - Requires some abstract "understanding"
    - Automatic abstraction is inherently interesting
  - "Poor Man's Transcription": Aligned MIDI data gives pitch, timing, and source instrument information without solving automatic transcription
  - Automatic Page Turning: computers can "listen" to audio and turn pages of aligned music notation
  - Compare great performances: How does Mario Lanza compare to Luciano Pavarotti?
  - Search: "Let's listen to the oboe solo at measure 200"
  - Editing: "Let's replace the audio from Thursday where someone coughed with the same spot recorded on Friday"

Linking/Sync'ing Different Representations (3)

- Music alignment is (partially) solved (robustly)
- Let's see how:
  - Step 1: Chromagram representation
  - Step 2: Distance function
  - Step 3: Dynamic programming
  - Step 4: Smoothing

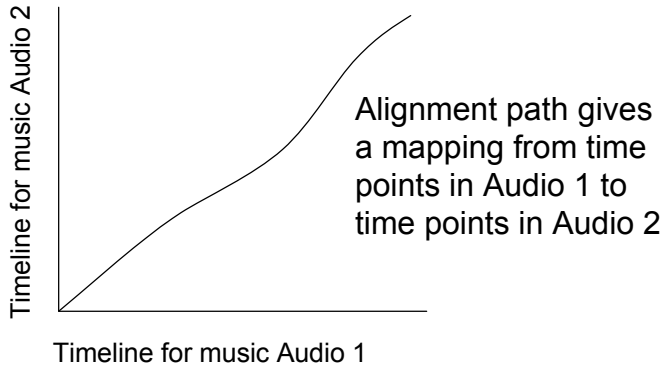
Chromagram Representation



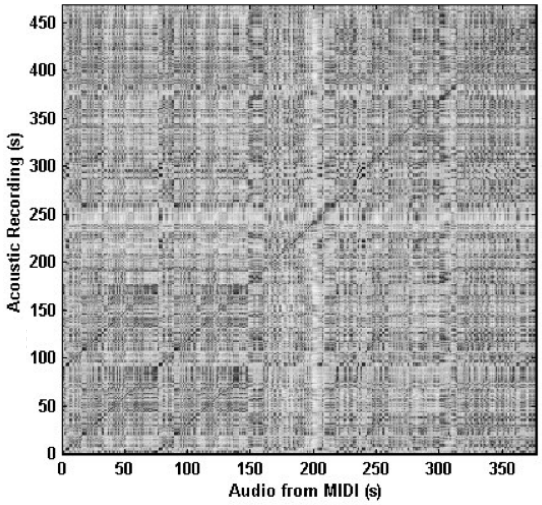
Distance Function

- Sometimes normalize each chromagram to a variance of 1 and a mean of 0: amplitude variations may not be consistently reproduced, so best to normalize them out
- Sometimes keep a "13th" vector element to indicate "silence": normalizing background noise during silence makes it hard to align silence to silence
- Euclidean distance works well
- Some use vector cosine (especially if vectors are not normalized)

Alignment: What Is It?



Dynamic Programming



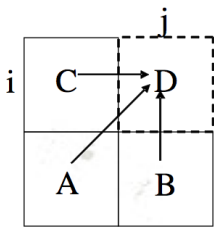
Extract feature vector for each frame of Audio 1 and Audio 2.

Compare NxM feature vectors (Euclidean, Cosine, etc.): *DISTANCE MATRIX*

Find lowest-cost path.

Dynamic Programming (2)

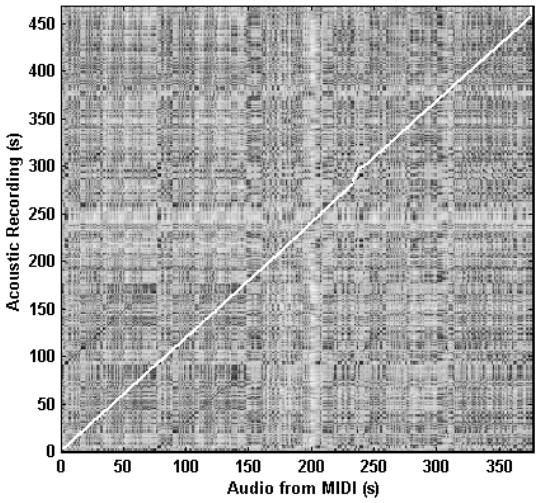
- Objective: find the path from [1,1] to [m,n] that minimizes the sum of distances along the way.
- Exponential number of paths: you can go left, right, or diagonal at each step.



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

- Trick: Store the *lowest* cost from [1,1] to [i,j] and compute cost incrementally in terms of previous solutions.

Dynamic Programming (3)



Computed Alignment Path

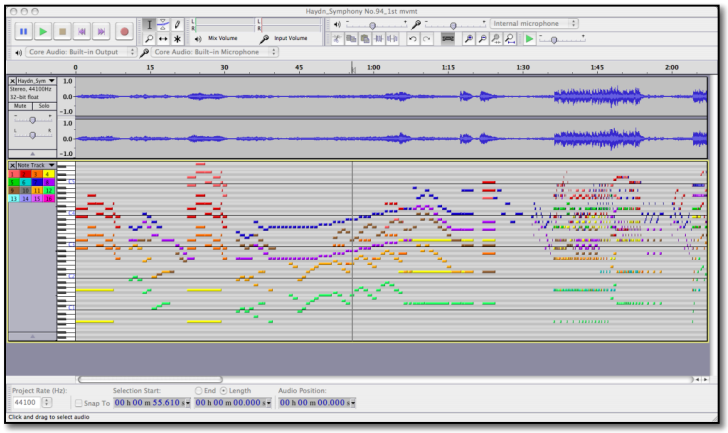
Smoothing

- Alignment tends to have some local irregularities: horizontal and vertical segments in path correspond to small but abrupt jumps in time
- Sometimes smoothing can help: fit smooth curves to approximate the alignment path

Chromagrams and MIDI

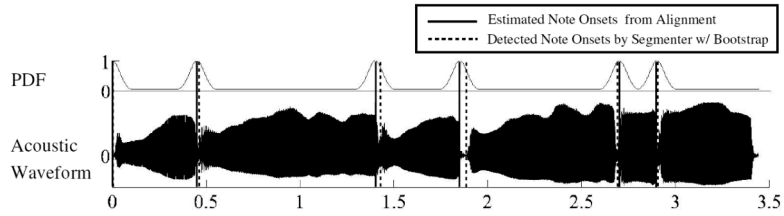
- Option 1: synthesize MIDI to audio, compute chromagrams as usual
- Option 2: set chroma vector bin to the count of all notes (or the sum of their velocities) in that pitch class

Score Alignment in Audacity



Finding Note Onsets

- Not all attacks are clean
- Slurs do not have obvious (or fast) transitions
- We can use score alignment to get a rough idea of where the notes are (~1/10 second)
- Then, machine learning can create programs that do an even better job (bootstrap learning).



Conclusions

- Music alignment based on DP is robust, fast, and has many applications.
- Still some bothersome problems:
  - Detecting beginning and ending (local alignment) is a problem
  - Tradeoffs between smoothness, local timing accuracy, and global robustness