Feature Extraction and Database Design for Music Software

Stephen Travis Pope, Frode Holm, and Alexandre Kouznetsov

Center for Research in Electronic Art Technology (CREATE)

University of California, Santa Barbara (UCSB)

ICMC, Miami, November, 2004

Outline

- Introduction
- Music database (MDB) systems: design issues and dimensions
- Domains of audio feature extraction
- MDB application requirements
- Design guidelines
- Examples

Introduction

- The field
  - Music information retrieval (MIR) literature
  - DSP for audio feature extraction
- This presentation
  - Tutorial aspect
  - Rant aspect
  - Terminology issues

Databases for Music (MDB)

- Uses of databases in music software
  - DB access as application (query/search)
  - Music SW that uses music/audio DB (increasingly common in the future)
- Databases offer
  - Persistency
  - Query capacity
  - Multiple access

My Premise

- In the general MIR literature, many systems suffer from weak feature vectors or too simplistic feature extraction.
- Most of what's missing isn't that difficult to do right.

Background

- MDB applications (not strictly "MIR")
  - DB management, query/access
  - Segmentation
  - Classification
  - Summarization (thumb-nailing)
  - Content identification (finger printing)
  - Preference matching (recommendation)
  - Integration with other music SW? (e.g., DAW, composition tools)

The Big Issues on MDB Systems

- Feature vector design
  - Simple/flat vs. complex/hierarchical
  - General purpose vs. application-specific
- Signal analysis (feature extraction) for a given feature vector
- DB storage of content and meta-data
  - Data volume, indexing, rights management, sharing/distribution

MDB Development

Developing MDB Systems

- Dimensions
  - Content format
  - Low-level analysis procedures
  - High-level derived features
  - DB design
  - Application flow and integration
- Design Issues
  - System architecture and design impacted by each of the MDB dimensions

Content Format

- Impacts all levels of system
  - Data volume, storage options, analysis DSP, DB design, etc.
- Systems may or may not maintain original source content (vs. metadata)
- Systems may preserve several formats of source and metadata (n-tier)
- This is typically a given rather than a design option

Content Formats

- Audio-based
  - Properties/volume of source recordings
- MIDI-based
  - Problems with MIDI, assumptions to make
  - Human-performed vs "dead pan" MIDI
- Score image based
  - Useful, but not treated here
- Formal language-based
  - SCORE, SMDL, Smoke, etc.

Low-level Analysis Procedures

- Audio DSP issues
  - Signal buffering and windowing
    - Window size, hop size -- compute load and data volume
  - Streaming vs. seekable sources
- Basic DSP analysis
  - Time-domain techniques
  - Frequency-domain techniques
  - Others (wavelets, time-domain granular summation)
  - Peak locators and trackers

High-level Derived Features

- Adding semantics to low-level features
- Perceptual mapping
- Data pruning
- Run-length encoding
- Segmentation
- Instrument identification (signatures)
- Location of important sections/notes

Database Design

- A big deal; lots of DB design issues, many of which are impacted by media data
- Relational DB models
- Object/network DB models
- Performance (on queries, inserts, etc.)
- Handling of data volume
- Multi-format systems (content and metadata stored separately)

Application Flow and Integration

- Normally a given (based on user requirements)
- Special case: general-purpose MDB design or MDB frameworks (e.g., Marsyas, FASTLab)
- Analysis engine architecture
- Query generation & on-the-fly analysis
- Result pruning and browsing

Feature Extraction

Domains of Audio Feature Extraction

- DASP techniques
  - Time-domain analysis
  - Frequency-domain analysis
  - Other signal analysis techniques
  - Second-stage data analysis
    - Data smoothing and massaging
    - Peak location and tracking
    - Running difference functions and distance metrics for segmentation

Time-domain Analysis

- Windowed RMS amplitude envelopes
  - Choice of window size, hop size, window function shape
  - May use several frequency bands (kick drum vs. hi-hat)
  - Useful for silence detection, beat tracking, simple segmentation, summarization, etc.
  - Simple, effective, well-understood techniques, many options

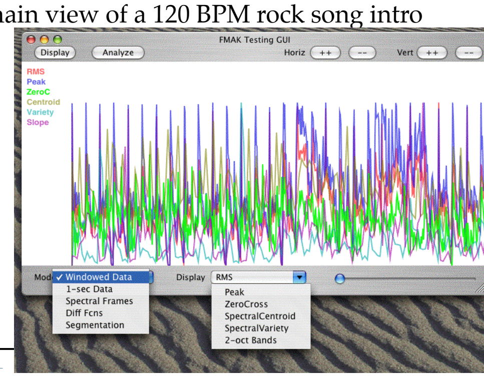
Higher-level Time-domain

- Derived data
  - Beat hierarchies
  - Tempo curves
  - Time signature (strong / weak beats)
- Time-domain segmentation (for human-performed music)
- Recognizing and handling click-track music

Tool Example: FMAK GUI

- Time-domain view of a 120 BPM rock song intro

- 4-bar intro
- Entrance of vocals near the middle of the screen



Frequency-domain Analysis

- Short-time Fourier transform
  - Configuration options and trade-offs
  - Interpretation / weighting of spectral bins
- Other frequency-domain techniques
  - Filter banks
  - Linear prediction
  - Filter matching

Spectral Peaks

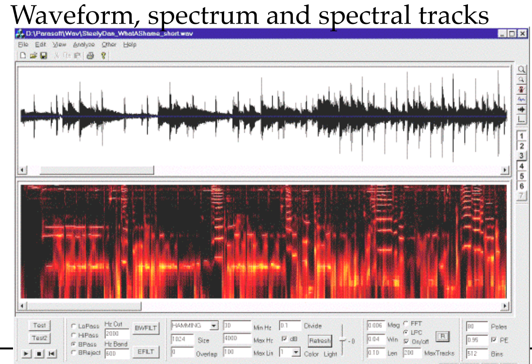
- Spectral peak detection and tracking
  - Simple methods: bin sorting, frame matching
  - Sophisticated methods: polynomial interpolation, drift management
- Spectral track birth / death statistics
  - Can be used for tempo tracking, etc.

Other Spectral Features

- Instrument signatures
  - Solo instrument identification
  - Location of style-indicative instruments
  - Still a challenge in mixed / compressed music
- Pitch tracking
  - Melody instrument vs bass tracking
  - Generally requires segmentation and / or band-pass filtering
- Spectral statistics
  - Slope, centroid, band weights, variety, etc.

Tool Example: FASTLab GUI

- Waveform, spectrum and spectral tracks



Wavelet and Granular Techniques

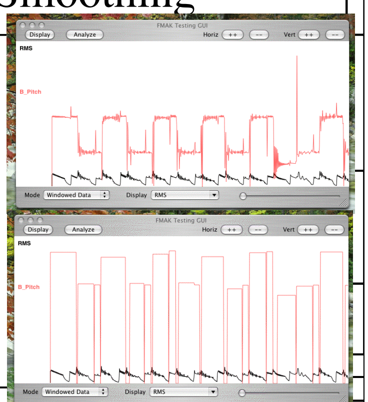
- Deterministic wavelet transform (iterative filtering and decimation)
- Interpretations of wavelet coefficients
- Time-domain decomposition and granular representations
  - "Matching pursuits" solutions
  - Decomposition dictionaries
    - Gaborlets vs. noise bursts
  - Current systems are unacceptably slow

Multi-pass and Cross-domain Analysis

- Initial segmentation, ID of significant notes or sections based on time-domain or spectral track statistics
- Detailed analysis of short segments (see Fig 1 above to find first note of vocal)
- Suggests heuristic approach, agents, or blackboard architecture

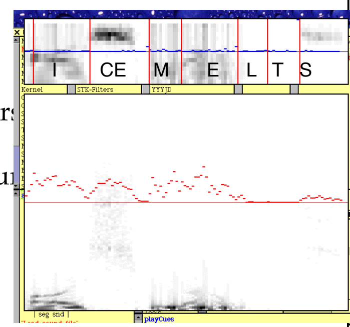
Noisy Data and Smoothing

- Windowing and processing artifacts
- Use statistical processing and perceptual limits
- Example: bass line for 12-bar blues, before and after smoothing



Segmentation

- Inter-frame distance metrics (feature weightings)
- Hierarchical segmenter assumptions about regular musical structu
- Example: 8S speech segmenter (simple)



Segmented Feature Vectors

- Multi-stage feature extraction
  - 1: Data analysis and segmentation (using a complex set of distance metrics)
  - 2: Prune the data to maintain a small set of feature vectors (1-per-segment & song peak/average data).
- Delivers a complex feature matrix with segment-length data, segmentation confidence, and a collection of individual feature vectors

MDB Applications

MDB Application Requirements

- Different application domains require different feature vectors!
- There are many options.
- It's an interesting design problem.
- Let's look at a few examples…

Players, Indexing

- The most common MDB application!
- Players often suffer from poor (or no) derived metadata or sophisticated indexing
- Some recent systems are changing this…
- There are many hard issues

MDB for Players

- Preference-matching recommender systems
- Automatic playlist generation
- Both require a powerful feature vector, genre classifier (possibly unlabeled), and user preference statistics (good style distance metrics)

Search, Matching

- Search in music databases based on text feature vectors is common (GraceNote, on-line stores)
- "Query by humming" in melody databases depends on melody extraction (batch and run-time)
- Timbral similarity measures are found (MuscleFish system)

Search, Matching Applications

- As above, these generally require
  - Large (segmented and pruned) feature vectors
  - Support utilities (clusterer, classifier, run-time analysis for queries)
  - Run-time search on complex feature set
  - Possible pruning of large return data sets

Summarization

- Audio thumb-nailing assumes good segmentation, and location of the most typical short segment
  - Choose most-frequently repeated section (appropriate?)
  - Choose first incidence of common section
  - Some applications require a single note or a common signal window?

Content Identification

- Important for indexing and stream-following (e.g., segmentation of video)
- Genre classification (speech, music, effect) useful for segmentation and indexing
- Exact finger-printing for digital rights management (some systems in use)
- Still difficult to avoid false positives and not be easily spoofed

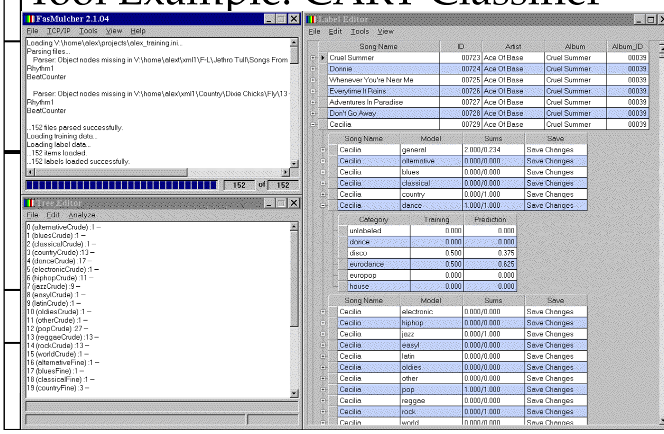
Identification Systems

- Is there a smaller-dimension unique identifier that's robust in the face of processing, editing, and even spoofing?
- Should it be based on:
  - Segmentation -- what about edited versions?
  - Averaged timbre of a selected section?
  - Spectral variety measures?
  - Voice or solo instrument signature?

Classification

- Data clustering in n-dimensional non-linear spaces is complex (KNN, Cure)
- Mapping discovered clusters to musical genres isn't obvious
- Rule-based techniques are useful
- Easy for a small number of genres (< 10)
- Quite a challenge for many genres (> 50)
- Requires a large segmented feature vector

Tool Example: CART Classifier

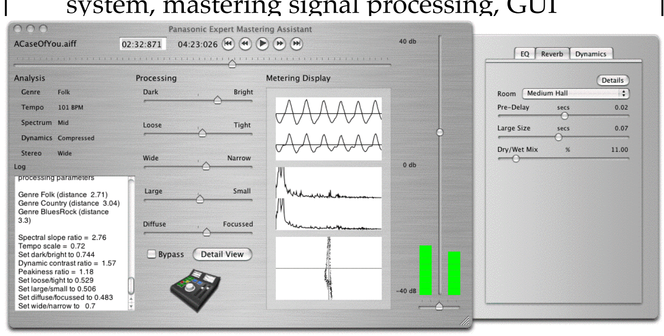


Mapping

- Mapping data from a (rich, segmented) feature vector to a set of processing parameters for some application:
  - Automatic spatialization/distribution
  - Automatic 2-to-5.1-channel up-mix
  - Tuning of post-processing or mastering expert systems
  - More to come…
- Special feature vector requirements

Tool Example: Expert Mastering Assistant

- Analysis, classification, mapping expert system, mastering signal processing, GUI



Design Guidelines

CREATE MAT Media Arts and Technology Graduate Program

Design Guidelines

- Questions in the paper (our advice was mixed in to the discussion above)
  - Source content and metadata formats
  - Low-level features and signal analysis
  - High-level derived features and statistics
  - Perceptual mapping and data reduction
  - Segmentation and feature vector pruning
  - Back-end database issues

CREATE MAT Media Arts and Technology Graduate Program

General Advice

- Basic data analysis (RMS, FFT) is fast and simple
- Advanced analysis techniques are either expensive (TDD) or hard to interpret (Wavelets)
- Deriving higher-level features is complex
  - Perceptual mapping
  - Feature vector statistics
- Many assumptions about musical form and instrumental timbres are not simply to codify, calling for heuristic programming techniques

CREATE MAT Media Arts and Technology Graduate Program

General Advice (2)

- Good musical segmentation and pruned feature vectors are key to the next level of MDB functionality
- Clustering and classification are easier given a sufficiently sophisticated feature vector
- Many applications still use quite shallow and simplistic feature extraction…

CREATE MAT Media Arts and Technology Graduate Program

Examples (see above figures)

- CREATE MDB Projects since 1996
  - *Paleo*: (1996-9, experimental) derive expressive performance data from MIDI; query on differences between performances
  - *Opera*: (1999-2001, experimental) search a DB of manuscript images based on the formal structure of an opera (*Wozzeck*); allow for users to add metadata and links

CREATE MAT Media Arts and Technology Graduate Program

Examples (2)

- *8S*: (1997-2000, in-use) large database of segmented speech and query tools built into a composition environment (Siren)
- *FASTLab*: (1999-2002, commercialized) fine-grained listener preference matching (used in Predixis music browser on MusicMatch)
- *EMA*: (2003/4, commercialized) expert system for mastering and post production (segmentation, classification, mapping, RT DSP, GUI, etc.)

CREATE MAT Media Arts and Technology Graduate Program

Questions?

(STP's in Malibu…)