Outline

- MIR Applications
- Signal Analysis and Feature Extraction
- MIR Application Design
- Feature-vector Design
  - Time-domain Features
  - Windowed Feature Extraction
  - Frequency-domain Features
  - Spatial-domain Features
  - Other Feature Domains
- APIs
Introductions, Context

Leigh Smith
- Comp Sci Dept. University of Western Australia
- Universiteit van Amsterdam – EmCAP project.
- IRCAM – Quaero project.
- Imagine Research Inc. now part of iZotope Inc.
- CCRMA MIR Workshop 2011

- lsmith@izotope.com
- http://www.leighsmith.com/Research
Problem Statement: MIR Applications

Examples
- Automatic playlist generation
- Audio transcription

Courtesy Stephen T. Pope
Example App:

- Classification

Non-silence Audio Clip
- With-Speech
  - Non-pure Speech
    - SVM2
  - Pure Speech
    - SVM1
- Background Sound
  - SVM3
- Non-speech
  - Music

Welcome to the Imagine Research sound classification demonstration.

Powered by our MediaMined™ sound-object recognition platform, this website recognizes sounds. Upload sounds such as MP3s or user-generated recordings. MediaMined will listen to the track and determine whether the recording is primarily speech, music, or other sound. Our system is learning hundreds of real-world sounds - we are just warming up...

How to Use
- Drag and drop of sounds only supported on Firefox.
- Upload a sound file with any content.
- Supports wav, aiff, mp3.
- Files must be less than 8MB.
- Only files with a sample rate greater than 11.25KHz will be analyzed.

The MediaMined collection currently recognizes:
- Music
- Speech
- Other
Signal Analysis and Feature Extraction for MIR Applications

- What do we want to do?
  - Match, search, index, transcribe, src-sep, ...

- What do we need to know to do it?
  - Basic feature set
  - Higher-level features
  - Feature data post-processing
  - Application integration

- MIR application design
  - How does the metadata fit in?

- Feature vector design for applications
Dimensions of Music Information Retrieval Applications

- Indexing, query, access
  - Use content or metadata for query
- Understanding, transcription
  - Derive (music/speech) model
- Clustering, classification
  - Feature vector for discrimination
- Content identification, finger-printing
- Preference-matching, recommendation
MIR/MDB Applications

Indexing
- Content description
- Time segmentation
- Separation
- Tempo tracking
- Pitch
- Other analyses

Classification
- Browsing

Real-time multichannel audio

Rendering
- Navigation

Recorded audio

Authoring

Metadata Operation scripts

Performing

Local & remote storage - Internet access - Peer-to-peer Sharing
Tzanetakis’s “MIR Pipeline”

Hearing Representation → Understanding Analysis → Acting Interaction

Signal Processing → Machine Learning → Human Computer Interaction
<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
<th>Matching</th>
<th>Features</th>
<th>Indexing</th>
<th>Collection Size (Records)</th>
</tr>
</thead>
<tbody>
<tr>
<td>anidentify!</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Inverted files</td>
<td>15,000</td>
</tr>
<tr>
<td>C-Brahms</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>none</td>
<td>278</td>
</tr>
<tr>
<td>CubyHum</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>LET</td>
<td>510</td>
</tr>
<tr>
<td>Cuidado</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>not described</td>
<td>works for &gt; 100,000</td>
</tr>
<tr>
<td>GUIDO/MIR</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Tree of transition matrices</td>
<td>150</td>
</tr>
<tr>
<td>Meldex/Greenstone</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>none</td>
<td>9,354</td>
</tr>
<tr>
<td>Musipedia</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Vantage objects</td>
<td>&gt; 30,000</td>
</tr>
<tr>
<td>notify! Whistle</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Inverted files</td>
<td>2,000</td>
</tr>
<tr>
<td>Orpheus</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Vantage objects</td>
<td>476,000</td>
</tr>
<tr>
<td>Probabilistic “Name That Song”</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Clustering</td>
<td>100</td>
</tr>
<tr>
<td>PROMS</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Inverted files</td>
<td>12,000</td>
</tr>
<tr>
<td>Cornell’s “QBI”</td>
<td>•</td>
<td></td>
<td>•</td>
<td>none</td>
<td>183</td>
</tr>
<tr>
<td>Shazam</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Fingertips are indexed</td>
<td>&gt; 2.5 million</td>
</tr>
<tr>
<td>SOMEJB</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Tree</td>
<td>359</td>
</tr>
<tr>
<td>SoundCompass</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Yes</td>
<td>11,132</td>
</tr>
<tr>
<td>Super MBox</td>
<td>•</td>
<td></td>
<td>•</td>
<td>Hierarchical Filtering</td>
<td>12,000</td>
</tr>
<tr>
<td>Themefinder</td>
<td>•</td>
<td></td>
<td>•</td>
<td>none</td>
<td>35,000</td>
</tr>
</tbody>
</table>
MIR Application
Design Stages

• Considerations
  – 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 these decisions.
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
  - MP3/AAC/WMA decoders

- MIDI-based
  - Problems with MIDI, assumptions to make
  - Human-performed vs “quantized” MIDI

- Score image based
  - Useful, but not treated here – genre specific.

- Formal language-based
  - SCORE, SMDL, Smoke, etc.
  - MusicXML
Numerical Processing

- Data Reduction, Smoothing
- Correlation, Grouping
- Princ./Indep. Component Analysis
- Audio Segmentation and Musical Form
- Clustering and Classification
Typical Processing Stages

- **Input processing**
  - Streaming, decompression, reformatting
- **Signal segmentation, windowing in time/freq**
  - Window size, share, overlap
- **1st-pass windowed feature extraction**
  - Basic time-, freq-domain features
- **2nd-pass feature processing**
  - Feature massaging, smoothing, pruning
  - 2nd-pass features (tempo, segmentation)
- **Post-processing, data output**
  - Many options
Signal Analysis

- Time-domain Audio Analysis
  - Windowed RMS Envelope Extraction
  - Beat Detection and Rhythm Analysis
  - Time-based signal segmentation

- Frequency-domain Analysis
  - Pitch Detection Techniques
  - Spectral Analysis and Interpretation
  - Spectral Peaks and Tracking
  - Other Spectral Measures

- Other Kinds of Analysis: Wavelets

- Cross-domain analysis
Real Applications

- Query systems, browsers, and MIR frameworks
- DBMS issues
- Machine Learning
- Informed tools
- Stand-alone delivery applications
Implementation example
Databases & Applications

• Searching, Indexing, and Players
• Audio Summarization and Thumb-nailing
• Content Matching and Finger-printing
• Data Clustering and Genre Classification
• Other Applications
Database Technology

- Database Designs: Schema vs. None?
- Relational DBMS (MySQL/Oracle/PostgreSQL)
  - Fixed table-formatted data
  - Few data types (number, string, date, ...)
  - One or more indices/table (part of DB design, application-specific, impacts performance)
  - Cross-table indexing and joins
  - SQL examples (create, insert, update, select)
- Media data (historically images)
  - Volume (large single items)
  - Format (items no known structure)
  - Content and metadata (required for usage)
  - Handling of Large/Dynamic Feature Vectors (MongoDB)
- Consider Application Requirements and Design
Feature-vector Design

• Application Requirements
  – Labeling, segmentation, etc.
  – Derive feature vector from the app requirements

• Kinds/Domains of Features
  – Time-domain
    • Simple features, onset detection
    • Rhythm, segmentation
  – Frequency-domain
    • Spectrum, spectral statistics
    • Pitch, chroma, key

(See e.g: http://www.create.ucsb.edu/~stp/PostScript/PopeHolmKouznetsov_icmc2.pdf)
Feature Vectors and Indexing

**Feature** = derived (numerical) parameter

- **Feature vector** = list of features for a single point/window in time, or average for an entire selection

- **Feature table** = list of feature vectors for several time slices (not always used/stored)
Example Features

Features:
- Time-domain, low-level
  - Windowed RMS amplitude
- Time-domain, high-level
  - Tempo, beat structure, segmentation
- Frequency-domain, low-level
  - Pitch, spectrum, spectral peaks
- Frequency-domain, high-level
  - Peak track birth/death statistics, instrument ID
- Many other possibilities (see below)
<table>
<thead>
<tr>
<th>Field</th>
<th>Bringin' Da Noise</th>
<th>I'll Be Your Everything</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume Width</td>
<td>48.126621</td>
<td>47.903584</td>
<td>0.182064596871</td>
</tr>
<tr>
<td>LPC Avg-Track-Dur</td>
<td>260.071</td>
<td>291.654</td>
<td>0.246736669056</td>
</tr>
<tr>
<td>Bass Loudness</td>
<td>-3.82097</td>
<td>-3.48169</td>
<td>1.151592910141</td>
</tr>
<tr>
<td>Spectral Contrast</td>
<td>17.8124</td>
<td>27.7138</td>
<td>1.260984294687</td>
</tr>
<tr>
<td>LPC Track-Harmo...</td>
<td>1.15606</td>
<td>1.10925</td>
<td>1.386355020613</td>
</tr>
<tr>
<td>BusyMid</td>
<td>399.87873138</td>
<td>382.9394489400</td>
<td>2.090529929650</td>
</tr>
<tr>
<td>Freq Max</td>
<td>0.579932</td>
<td>0.629061</td>
<td>2.756166578401</td>
</tr>
<tr>
<td>Average Volume</td>
<td>34.344021</td>
<td>37.742193</td>
<td>3.092778888824</td>
</tr>
<tr>
<td>Freq Avg</td>
<td>0.004416</td>
<td>0.004209</td>
<td>3.27324781783</td>
</tr>
<tr>
<td>Tempo</td>
<td>111.966</td>
<td>105.943</td>
<td>3.872166433080</td>
</tr>
<tr>
<td>LPC Peaks-Per-S...</td>
<td>258.61</td>
<td>229.837</td>
<td>5.144795608229</td>
</tr>
<tr>
<td>LPC Freq-Deviation</td>
<td>6257.06</td>
<td>5584.61</td>
<td>5.146495852036</td>
</tr>
<tr>
<td>% Freq Over Avg</td>
<td>24.050509</td>
<td>21.898819</td>
<td>5.313072728419</td>
</tr>
<tr>
<td>Spectral Variety</td>
<td>57.0208</td>
<td>97.2588</td>
<td>5.591531132924</td>
</tr>
<tr>
<td>BusyLow</td>
<td>412.44579522</td>
<td>341.0040312499</td>
<td>6.891936456624</td>
</tr>
<tr>
<td>Spectral Saturation</td>
<td>0.712956</td>
<td>0.651703</td>
<td>7.476978442821</td>
</tr>
<tr>
<td>LPC Tracks-Per-S...</td>
<td>56.5431</td>
<td>48.2628</td>
<td>7.601499754612</td>
</tr>
<tr>
<td>Snare Strength</td>
<td>0.328855</td>
<td>0.235586</td>
<td>8.982285629537</td>
</tr>
<tr>
<td>Overall Grunge</td>
<td>0.248330529671</td>
<td>0.067614786427</td>
<td>12.20650524954</td>
</tr>
<tr>
<td>% Rhythm</td>
<td>99.48301435406</td>
<td>97.82279545454</td>
<td>N/A</td>
</tr>
<tr>
<td>BEAT: hiqout</td>
<td>5.2</td>
<td>5.8</td>
<td>N/A</td>
</tr>
<tr>
<td>BEAT: maxscore</td>
<td>1550.0</td>
<td>926.0</td>
<td>N/A</td>
</tr>
<tr>
<td>BEAT: spikewon</td>
<td>0.0</td>
<td>0.0</td>
<td>N/A</td>
</tr>
<tr>
<td>BEAT: window</td>
<td>20.0</td>
<td>20.0</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Example: FMAK3 Feature Table

class FeatureTable {
    // FeatureTable is a root object (no parents)

public:
    // Data members (instance variables)
    float mTimeStamp; // When do I start?
    float mTimeDur; // How long a time-span do I represent?
    // Time-domain features
    unsigned int mRMSErrorSize; // Size of RMS window
    FeatureDatum mRMS; // Rectangular-windowed RMS amplitude
    FeatureDatum mPeak; // Max sample amplitude
    FeatureDatum mLPRMS; // RMS amplitude of LP-filtered signal
    FeatureDatum mHPRMS; // RMS amplitude of HP-filtered signal
    size_t mZeroCrossings; // Count of zero crossings
    FeatureDatum mDynamicRange; // RMS dynamic range of sub-windows
    FeatureDatum mPeakIndex; // RMS peak sub-window index
    FeatureDatum mTempo; // RMS/FWT instantaneous tempo estimate
    FeatureDatum mTimeSignature; // Time signature guess
    FeatureDatum mBassPitch; // Bass pitch guess in Hz
    unsigned int mBassNote; // Bass note (MIDI key number) guess
    FeatureDatum mBassDynamicity; // Bass note dynamicity (size of histogram)
    // Spatial features
    FeatureDatum mStereoWidth; // L/R difference
    FeatureDatum mSurroundDepth; // Front/Surround difference
    FeatureDatum mCenterDistinction; // Center vs. L/R sum difference
Example: FMAK3 Feature Table, cont’d

```c
// Frequency-domain features
unsigned int mFFTWindowSize;  // Size of FFT window
FtVector mSpectrum;           // Hanning windowed FFT data (1024 points, or NULL)
FtVector mReducedSpectrum;    // 1-octave FFT data (10-12 points)
FtVector mBandSpectrum;       // 2.5-octave FFT data (4 points -- spectral bands)
FPartialVector mSpectralPeaks; // List of major spectral peak indeces
FPartialVector mSpectralTracks; // List of tracked peak frequencies
FeatureDatum mSpectralCentroid; // Spectral centroid measure
FeatureDatum mSpectralSlope;   // Spectral slope measure
FeatureDatum mSpectralVariety; // Inter-frame spectral variety measure
                            // Hi-frequency properties
FeatureDatum HiFreqBalance;   // Relative HF level
FeatureDatum HiFreqVariety;   // HF inter-frame spectral variety
FeatureDatum HiFreqCorrelation; // Correlation between HF and audio-band tracks
FeatureDatum mSTrackBirths;    // Spectral peak track births and deaths
                            // LPC features
unsigned int mLPCWindowSize; // Size of LPC window
FPartialVector mLPCFormants;  // List of LPC formant peaks
FPartialVector mLPCTracks;    // List of tracked LPC formants
FeatureDatum mLPCResidual;    // LPC residual level (noisiness)
FeatureDatum mLPCPitch;       // Pitch estimate
FeatureDatum mLTrackBirths;   // LPC formant peak track births, deaths
                            // Wavelet-domain (FWT) features
FtVector mWaveletCoeff;       // FWT coefficient or NULL
FtVector mWTNSpectrum;        // Reduced FWT HiFreq noise spectrum
FtVector mWTTracks;           // List of tracked FWT peaks
FeatureDatum mWTNoise;        // FWT noise estimate
```

Monday, June 25, 2012
Intermission

To be continued!
Analysis Domains and Transformations

- Time-domain Audio Analysis and Applications
  - Windowed RMS Envelope Extraction
  - Beat Detection and Tempo Analysis
  - Time-based signal segmentation
- Frequency-domain Analysis
  - Pitch Detection Techniques
  - Spectral Analysis and Interpretation
  - Spectral Peaks and Tracking
  - Other Spectral Measures
- Other Kinds of Analysis: Wavelets
- Cross-domain analysis
Feature Extraction and Signal Analysis

Multi-step process:
- Read input
- Apply window
- Derive several low-level features
- Map, derive next-level features
  - Possible heuristics determine which next-level features are relevant
- Prune data when appropriate
Time-domain Features

- RMS, Peak
- LPF/HPF RMS
  - e.g., F < 200 Hz, F > 2000 Hz
- Dynamic range
  - What window for calc?
- Zero-crossing rate (time- or freq-domain?)
- Higher-level statistics
  - Mean/variance
  - Variance of sliding windows
  - Spacing of peaks/troughs
  - Many other options
- Time-domain onset-detection & beats

\[ RMS = \sqrt{\frac{1}{N} \sum_{n} x_n^2} \]
Time Sequences, Windowing

- Read audio input
- Vector multiply by window function
- Perform analysis
- Step to next window
- Hop size not normally = window size (overlap)
- Window features
  - Main lobe width, side lobe level, side lobe slope
Time-domain Audio Analysis and Applications

- Use rectangular window if no overlap or triangular window if overlapping
- Medium-sized window (10 Hz or better resolution desired)
- Derived windowed RMS value
- Count zero crossings
Windowed RMS Envelope Extraction

pseudo-code for envelope extraction:

- Outer loop for windows
- Inner loop to run window and compute RMS value
- Silence threshold (noise gate)
- Note-on trigger (peak detector)
- Example sound: piano sample, drum loop
Optional Time-domain Steps

- Pre-filter to get low-freq and high-freq RMS values
- Process stereo channels to get M/S (sum/difference) signals
- Noise detection
- Silence detection
Windowed Feature Comparison
Windowed 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
Frequency-domain Features

- Spectrum, Spectral bins
  - Window/hop sizes
  - Improving spectral data: phase unwrapping, time realignment

- Spectral measures (statistical moments)
- MFCCs
- Peak-picking and peak-tracking
- Pitch-estimation and pitch-tracking
Frequency-domain Analysis

- Short-time Fourier transform
  - Configuration options and trade-offs
  - Interpretation/weighting of spectral bins (perceptual scales)
- Other frequency-domain techniques
  - Filter banks
  - Linear prediction
  - Filter matching
- Loads of options
Speech Spectrogram

- Kinds of spectral plots
- Features
Windows and their Spectra

- Trade-offs between window characteristics
- Different windows for different analysis domains
Advanced Windows for Spectral Analysis

Blackman Window
\[ a_0 = 0.427, \ a_1 = 0.497, \ a_2 = 0.077 \]

Blackman Window’s Transform
\[ a_0 = 0.427, \ a_1 = 0.497, \ a_2 = 0.077 \]

Kaiser Window
\( (\alpha = 3.5) \)

Kaiser Window’s Transform
\( (\alpha = 3.5) \)

Blackman-Harris Window
\( (4\text{-term}, \ -96 \text{ dB}) \)

Blackman-Harris Window’s Transform
\( (4\text{-term}, \ -96 \text{ dB}) \)
Windowing and the STFT

Input

Time

Windowed excerpts

Time-varying spectra

Fast Fourier Transform FFT
The Pitch/Time Trade-off
Harmonics and Formants

- Source/Filter – instrument resonances

Formants
Harmonics
Eee vowel (beet)
Composite Spectra

How to disambiguate?
- Track birth/death statistics
- Vibrato (see figure)
- Statistical techniques
Spectral Analysis and Interpretation

• Spectral data extraction
  - Base frequency
  - Overtone spectrum
  - Formants, resonances, regions
  - Instrument signatures

• Spectral statistics
  - Peak, mean, average, centroid, slope, etc.
  - Spectral variety, etc.
Spectral Moments

- Spectral Centroid – (Mean) 1st moment
- Variance – 2nd moment
- Skew – 3rd Moment
- Kurtosis 4th Moment
Spectra as Time-varying

- Track peaks/regions between frames (requires thresholds of change)
- Model the dynamicity (e.g., formant trajectory, vibrato extraction)
Spectral Peaks and Tracking

- Peak finding (remember autocorrelation?)
- Peak discrimination
- Peak continuation: tracks and guides
- Derived statistics
- Problem cases
Peaks and Tracks

- **Peak-finding**
  - Thresholds, distances, heuristics
- **Peak-continuation**
  - Inter-frame distances and guides
  - Dropped frames and stretching
  - Track birth/death criteria
Spectral Peak-Tracking Example
Spectral Peak Detection Algorithm

From Blum et al. patent # 5,918,223
Spectral Smoothness Measure

(a)

(b)
Smoothed Spectrum Types
Pitch Detection Techniques

- Find the period of a “periodic” signal
  - First guess whether or not it’s periodic
- Simple techniques work for many signals
  - Zero-crossings (with direction, slope)
  - Autocorrelation (with range limitation)
- But it’s hard to tell when they fail
  - Random data, silence
  - Octave over/under-tone errors
Filter-based Pitch Detection

Simple adaptive process for single-frequency source with strong fundamental (i.e., many, but not all, instruments and voices)

- Easily implemented in analog circuitry
- Many variations
Auto-Correlation

- Slide a signal across itself, taking the vector product at each step
- This AC array has a peak at 0, and the period of the signal
- No peaks for noise
Harmonic Product Spectra

- Decimation of FFT spectra, summation, and spectral peak location
- Assumes overtones are significant, not that fundamental is
Equal-loudness Curves

- Fletcher–Munson vs. Robinson–Dadson

The curves represent equal loudness as perceived by the average human ear. The ear is less sensitive to low frequencies, and this discrimination against lows becomes steeper for softer sounds. The maximum sensitivity region for human hearing is around 3-4 kHz and is associated with the resonance of the auditory canal.

Sound intensity in decibels does not directly reflect the changes in the ear's sensitivity with frequency and with sound level.

Equal loudness in phons.

Intensity in decibels

Frequency (Hz)
Frequency Regions and Scaling

- Mel-warped frequency bands
Mel–Freq Cepstral Coefficients

Steps:
- Signal
- FT
- Log magnitude
- Phase unwrapping
- FT (or DCT)

Interpretations
- “ceps” = “Spec”trum of spectrum
- “Quefrency”
- Mel–scale
- Mel–scale filters

Instead of AC, use FFT or DCT of PDS

Leads to interesting statistics of higher–level spectral properties, see next section
MFCC Analysis

Analogy
- Start with log spectrum of mixed complex tones: several sets of related partial peaks
- Take, e.g., the autocorr. of the FFT PDS
- Warped frequencies of peaks correspond to fundamental frequencies of overtone series
Comparison With LPC (by Andrianakis & White)

FFT PSD | 13-pole LPC Spectrum | 20 Mel Spectrum

Train

Car

Frequency [Hz]
Spatial-domain Features

- M/S Encoding (stereo sum & difference)
- Surround-sound processing
  - L/R vs C
  - L/R vs Ls/Rs
- Frequency-dependent spatial separation
- Higher-dimensional sources
- Stem tracks
Other Feature Domains

- Other time-domain features
  - Beats, beat histograms
- Other frequency-domain features
  - Fluctuation patterns
- Other time–frequency transforms
  - Filter banks
- Wavelets
  - Trades off temporal & spectral resolution
- Linear Predictive Coding
  - Polynomial representation
APIs for MIR Tools

- Marsyas: G. Tzanetakis (11), flexible tool set, scripting language, segmentation and classification
- LibOFA: Holm/Pope (00), simple FV for unique ID comparing to a large pre-analyzed database
- D2K/M2K: West/MIREX (06), Java-based GUI related to D2K, many apps.
- LibTSP: P. Kabal (00), C routines for DASP & IO
- CSL: STP/MAT (05), C++ class library for DASP, synthesis, control, spatialization and MIR
APIs – 2

- Libxtract
- Aubio
- SonicVisualizer plug-ins
- Loris
- SPEAR
- CSL
- LibTSP
Spectral Tools

• SPEAR
• Loris
• Marsyas
• Sonic visualizer
Review

- MIR Apps
- Signal analysis processing chains
- Feature vector design from app requirements
- Kinds of audio features
- Basic feature statistics