



POLITECNICO
MILANO 1863

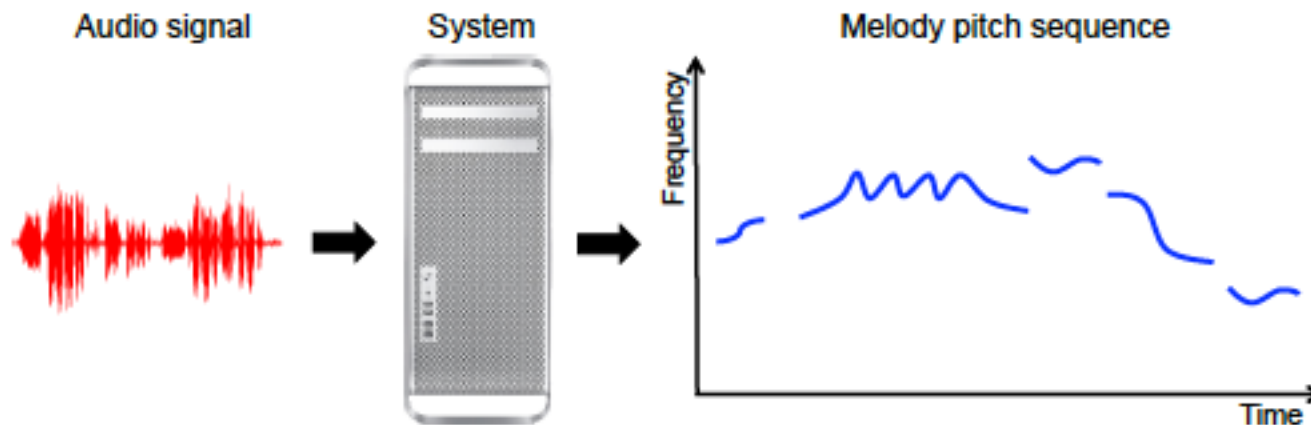


Melody extraction from polyphonic music signals

- Introduction and problem statement
- Salience-based approaches
- Source separation-based approaches
- Alternative approaches
- Examples of systems (updated to 2013)
- Evaluation metrics
- Comparison between different systems

Introduction and problem statement

- In order to interact with music repositories for services of searching, indexing, description, it is of paramount importance to develop methods that are able to analyze the musical content of audio files
- One of the most important parameters that must be analyzed is the melody.
- Known as “Melody extraction”, “Audio Melody Extraction”, “Predominant Melody Extraction”, “Predominant Fundamental Frequency Estimation”, this task involves obtaining a sequence of frequency values representing the pitch of the dominant melodic line from recorded music audio signals.



Introduction and problem statement

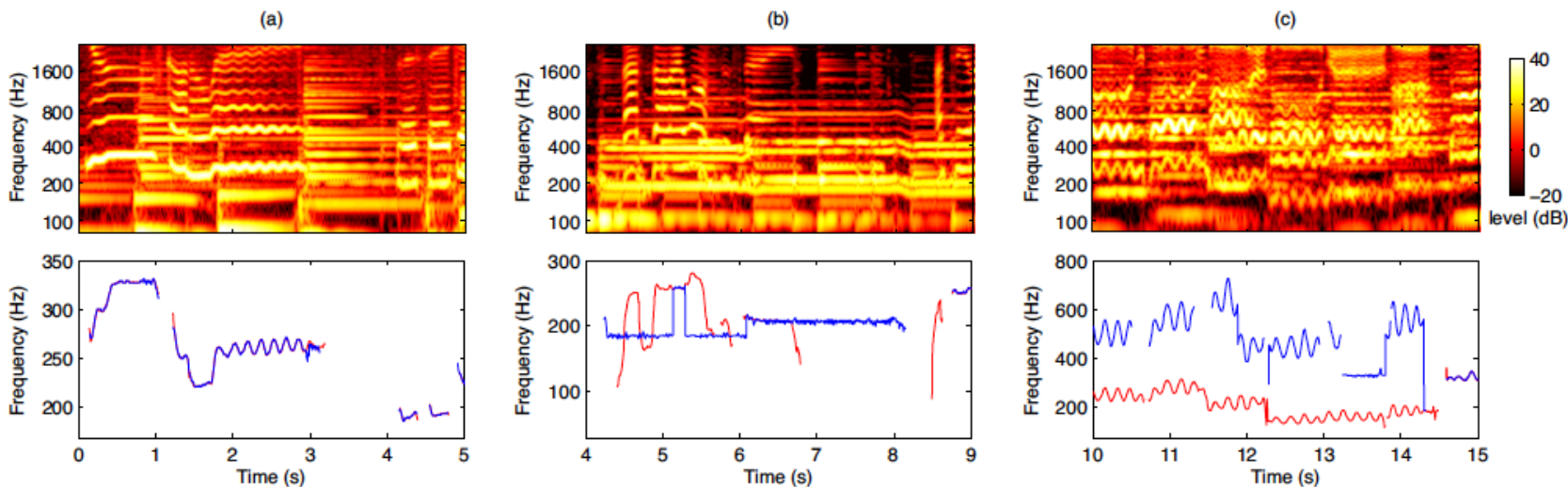
- Melody transcription is a task that can be achieved by a trained music student.
- It is very difficult to approach in an automated fashion, due to the complex and overlapped spectral structure of musical harmonies.
- It is easier to focus on the more limited transcription task of recovering a single melody line as the “strongest” pitch in the melody range at any time.
- The definition of pitch that in the Music Information Retrieval community is currently adopted is: “the melody is the single (monophonic) pitch sequence that a listener might reproduce is asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the essence of that music when heard in comparison”.
- In this context, polyphonic refers to any type of music in which two or more notes can sound simultaneously, be it on different instruments or on a single instrument.

Introduction and problem statement

- After the above definition of polyphonic music, we could restate the melody extraction problem as “Given a recording of polyphonic music, we want to estimate the sequence of fundamental frequencies that corresponds to the pitch of the lead voice of the instrument. Moreover, we must estimate the time intervals when this voice is not present”.
- Difficulties encountered when approaching this task in an automated fashion:
 1. A polyphonic music is the superposition of the sound waves produced by all instruments in the recording. The frequency components of the different sources superimpose making it very hard to attribute specific energy levels in specific frequency bands to the notes of the individual instruments. Mixing and mastering can add reverberation or apply dynamic range compression.
 2. Even after obtaining a pitch-based representation of the audio signal, we still need to determine which pitch belong to the predominant melody.

Case study examples

- Top row: log-frequency spectrograms of vocal jazz (a), pop music (b), and opera (c).
- Bottom row: final melody line estimated by the algorithm (blue) overlaid on top of the ground truth (red).



- Estimated and ground truth melody sequences overlap almost perfectly.
- Pitch errors (seconds 4 to 7) and voicing errors (seconds 7 to 9) are present. In this case, the backing vocals are confused with the lead singer.
- The algorithm makes octave errors. The pitch class of the melody is correctly estimated, but in the wrong octave. This is due to the fact that the second harmonic of opera singers is often stronger than the fundamental

Monophonic pitch trackers

Monophonic pitch trackers typically take the audio signal $x(t)$ and calculate a function $S_x(f_\tau, \tau)$ evaluated across a range of candidate pitch frequencies that indicates the score or likelihood of the pitch candidates at each time frame τ .

- The function can be calculated in the time domain (e.g. autocorrelation evaluated over a range of lags), or in the frequency domain (a function of the magnitude spectrum evaluated over a range of frequencies).
- The local estimates of period are subject to sequential constraints.

Monophonic pitch trackers

Estimated sequence of pitch values:

$$\hat{\mathbf{f}}_{\text{mon}} = \arg \max_{\mathbf{f}} \sum_{\tau} S_x(f_{\tau}, \tau) + C(\mathbf{f})$$

where f_{τ} is the element of \mathbf{f} at time τ and $C(\mathbf{f})$ accounts for the temporal constraints.


Possible choice for the likelihood function: autocorrelation function

$$S_x(f, \tau) = r_{xx}\left(\frac{1}{f}, \tau\right) = \frac{1}{W} \int_{\tau-W/2}^{\tau+W/2} x(t)x\left(t + \frac{1}{f}\right) dt$$

W length of the autocorrelation analysis window

Extension to polyphonic music

For the purpose of melody extraction, we conceive the polyphonic music as the target signal from which we aim at estimate the pitch, plus accompaniment “noise”:

$$y(t) = x(t) + n(t)$$


Salience-based melody extraction: improve the robustness of the underlying pitch candidate scoring function, so it continues to reflect the desired pitch even in the presence of other periodicities

$$\hat{\mathbf{f}}_{\text{sal}} = \arg \max_{\mathbf{f}} \sum_{\tau} S'_y(f_{\tau}, \tau) + C'(\mathbf{f})$$

Separation-based melody extraction: decompose the mixed signal into separate sources, where $\hat{x}(t)$ contains the melody signal to a degree that makes it suitable for a largely unmodified pitch tracker

$$\hat{\mathbf{f}}_{\text{sep}} = \arg \max_{\mathbf{f}} \sum_{\tau} S_{\hat{x}}(f_{\tau}, \tau) + C'(\mathbf{f})$$

Salience-based melody extraction: salience function

Salience-based methods compute the salience of a candidate frequency f as the weighted sum of its harmonics:

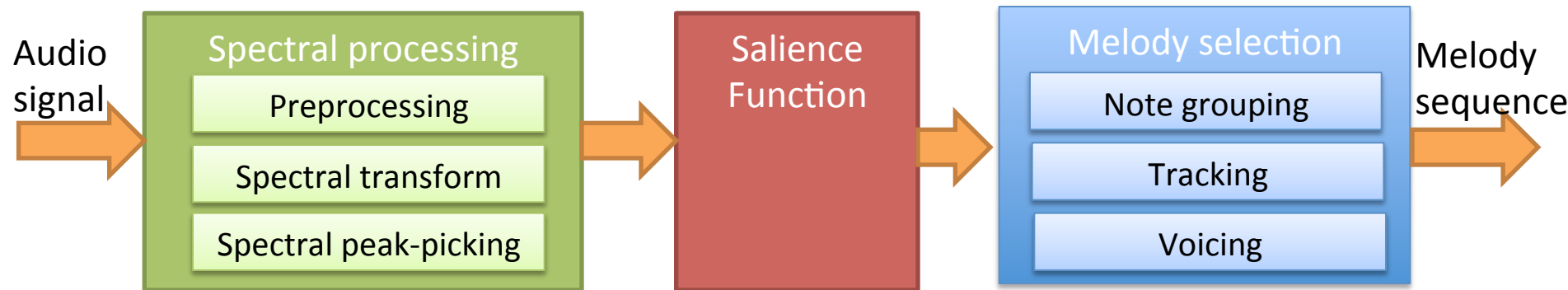
$$S'_y(f_\tau, \tau) = \sum_{h=1}^{N_h} g(f_\tau, h) |Y(h \cdot f, \tau)|$$

where

- N_h is the number of harmonics in the summation
- $g(f_\tau, h)$ is a harmonic weighting function
- $Y(f, t)$ is the short-time Fourier transform

Salience-based melody extraction: block diagram

General architecture of salience-based techniques



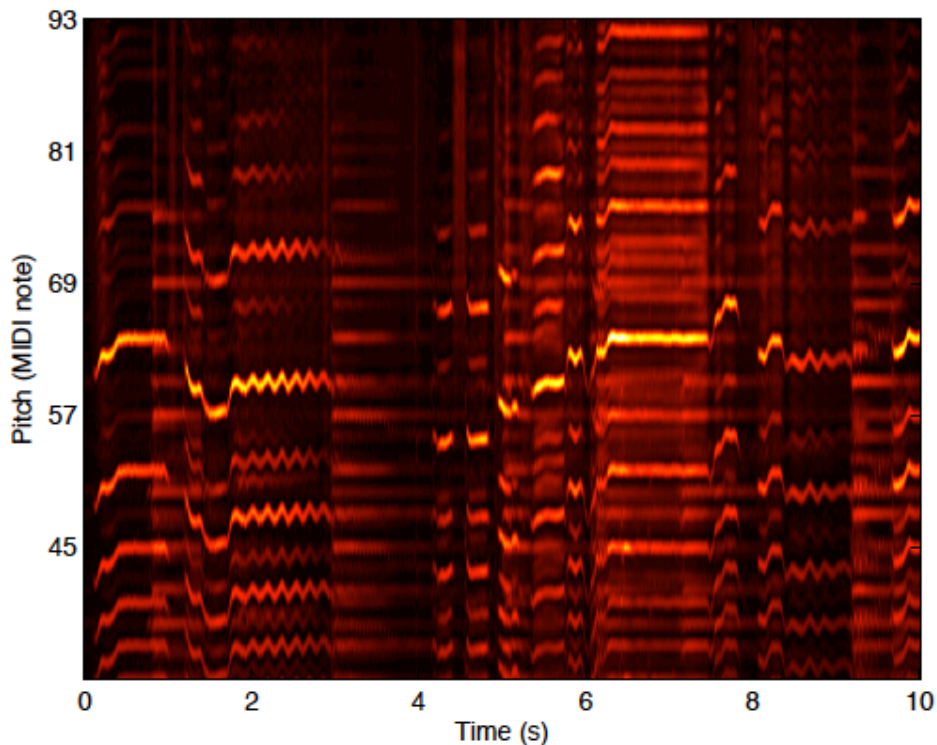
- **Preprocessing:** a filter to enhance the frequency content where we expect to find the melody. Could be a bandpass filter or a perceptually motivated equal loudness filter. Other approaches use a preliminary separation to enhance the melody signal before it is further processed.

Saliency-based melody extraction: spectral transform

- **Spectral transform and processing:**
 - a. the signal is chopped into time frames and a transform is applied. Possible choices:
 1. STFT with frame lengths between 50 and 100 ms.
 2. Constant-Q transform or multi-resolution FFT
 3. Techniques that imitate the human auditory system.
 - b. Most approaches only use spectral peaks for further processing. Some techniques pick only peaks that are related to a sinusoidal content (i.e. discard peaks that are not sufficiently isolated), others apply a spectral magnitude normalization to reduce the influence of timbre on the analysis.

Salience-based melody extraction: salience function

The salience function provides an estimate of the salience of each possible pitch value over time.



Example of the output of a salience function for an excerpt of vocal jazz. Peaks of this function are taken as possible candidates for the melody. Along with the function already described, also statistically motivated techniques are used to select the fundamental frequency f_0 as the one that maximizes the maximum a posteriori probability of the tone model.

Salience-based melody extraction: ghost effect

The salience function is subject to the appearance of “ghost” pitch values whose f_0 is an exact multiple (or sub-multiple) of the f_0 of the actual pitched sound. This effect can lead to what is commonly referred to as octave errors, in which an algorithm selects a pitch value which is exactly one octave above or below the correct pitch of the melody.

In order to address this problem, different methods have been developed:

- Based on the observation of frequency relationships among peaks in a given time frame (i.e. attenuating the salience of peaks that occur at frequencies multiple of other peaks)
- By looking at the trajectory followed by peaks at multiple time frames (looking at the presence of duplicated trajectories separated by exactly one octave, or removing portions of trajectories that appear irregularly, which can be due to the octave error).

Salience-based melody extraction: tracking

The remaining task is to determine which peaks belong to the melody.

- Most techniques directly track the melody from the salience peaks.
- Some include a preliminary grouping stage to group peaks into continuous pitch contours.
- Given the pitch contours or the salience peaks, different tracking techniques have been developed. Examples:
 - Clustering;
 - Heuristic-based tracking agents;
 - Dynamic programming
 - Delete all the pitch contours or salience peaks that do not belong on the melody, based on predetermined rules.

Salience-based melody extraction: voicing

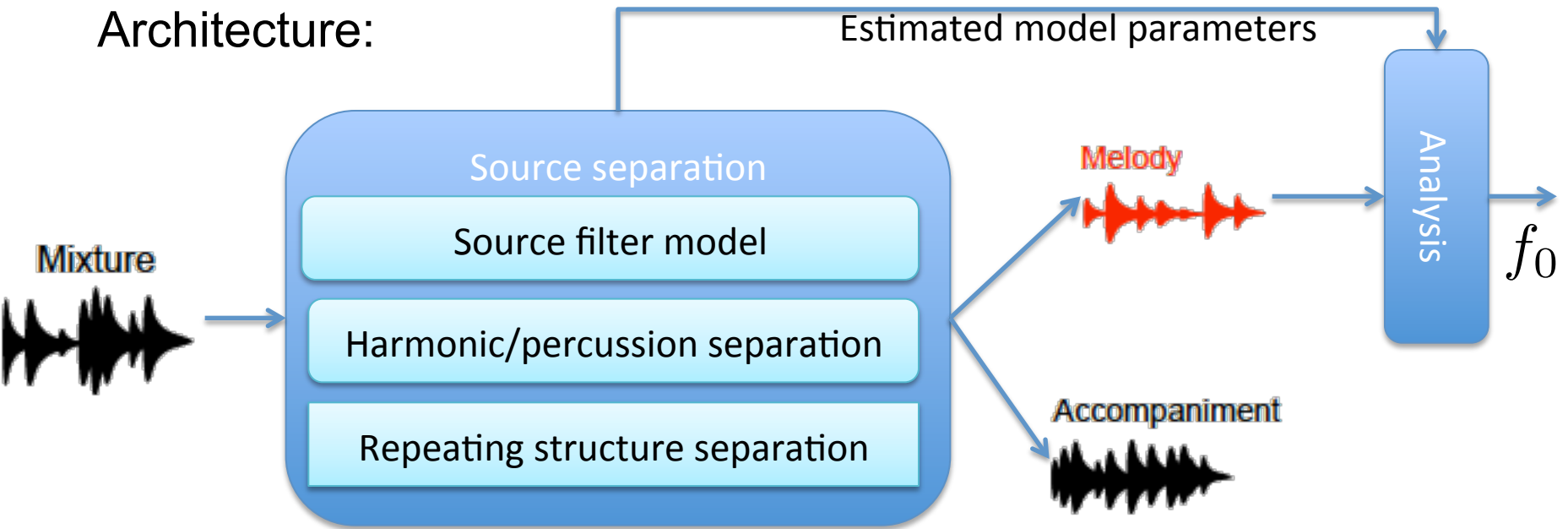
Voicing refers to the functionality of discriminating between parts where the melody is present and where it is not.

Common approach: use a fixed or dynamic threshold based on the salience function.

Alternatives:

- use an HMM classification that is trained on distinguishing between voiced and unvoiced parts.
- Use a classification system based on timbre that is able to identify the presence of the voice or of the lead instrument.

Separation-based approaches: architecture



Separation based approaches have recently gained popularity.

Separation-based approaches

In [2] the authors adopt a representation of the lead voice as a source-filter model:

- a Smooth Instantaneous Mixture Model (SIMM), which represents the lead voice or instrument as the instantaneous mixture of all possible notes;
- a Smooth Gaussian Scaled Mixture Model (SGSMM), which allows one source/filter couple to be active at any time. It is more realistic but computationally heavier.
- In both cases training is done using Expectation/Maximization.

The contribution of the accompaniment is the sum of an arbitrary number of sources with distinct spectral shapes.

Once the model parameters are estimated, the final melody is obtained using the Viterbi algorithm to find a smooth trajectory through the model parameters.

Voicing (i.e. selecting the time frames where the lead voice/instrument is present) is accomplished through the analysis of energy of the melody.

Separation-based approaches

In [3] authors perform a preliminary Harmonic-Percussion Sound Separation (HPSS), to separate harmonic from percussive elements. It exploits the fact that the former are smooth in time, while the latter are smooth in frequency.

- By changing the window length used for the analysis, the algorithm can be used to separate “sustained” (i.e. chord) sounds from “temporally variable” (melody + percussive) sounds.
- Once the accompaniment is removed, the algorithm is run again, this time in its original form in order to remove percussive elements.
- The melody frequency sequence is obtained directly from the spectrogram of the enhanced signal using dynamic programming by finding the path which maximizes the MAP of the frequency sequence, where the probability of a frequency given the spectrum is proportional to the weighted sum of the energy at its harmonic multiples, and transition probabilities are a function of the distance between two subsequent frequency values.
- Voicing detection is done by setting a threshold on the (Mahalanobis) distance between the two signals produced by the second run of the HPSS algorithm (the melody signal and the percussive signal).

In order to analyze the effectiveness of a melody extraction technique it is of fundamental importance to select suitable measures. Some examples:

- **Voicing Recall Rate:** The proportion of frames labeled as melody frames in the ground truth that are estimated as melody frames by the algorithm.

$$\text{Rec}_{\text{vx}} = \frac{\sum_{\tau} v_{\tau} v_{\tau}^*}{\sum_{\tau} v_{\tau}^*}$$

- **Voicing False Alarm Rate:** The proportion of frames labeled as non-melody in the ground truth that are mistakenly estimated as melody frames by the algorithm.

$$\text{FA}_{\text{vx}} = \frac{\sum_{\tau} v_{\tau} \bar{v}_{\tau}^*}{\sum_{\tau} \bar{v}_{\tau}^*}$$

- **Raw pitch accuracy:** The proportion of melody frames in the ground truth for which f_τ is considered correct (i.e. within half a semitone of the ground truth f_τ^*).

$$\text{Acc}_{\text{pitch}} = \frac{\sum_{\tau} v_{\tau}^* \mathcal{T}[\mathcal{M}(f_{\tau}) - \mathcal{M}(f_{\tau}^*)]}{\sum_{\tau} v_{\tau}^*} \quad \mathcal{T}[a] = \begin{cases} 1 & \text{if } |a| < 0.5 \\ 0 & \text{if } |a| \geq 0.5 \end{cases}$$

And \mathcal{M} maps a frequency in Hertz to a melodic axis as a real-valued number of semitones above an arbitrary reference frequency

$$\mathcal{M}(f) = 12 \log_2 \left(\frac{f}{f_{\text{ref}}} \right)$$

- **Raw Chroma Accuracy:** As raw pitch accuracy, except that both the estimated and ground truth f_0 sequences are mapped onto a single octave. This gives a measure of pitch accuracy which ignores octave errors, a common error made by melody extraction systems:

$$\text{Acc}_{\text{chroma}} = \frac{\sum_{\tau} v_{\tau}^* \mathcal{T} [\langle \mathcal{M}(f_{\tau}) - \mathcal{M}(f_{\tau}^*) \rangle_{12}]}{\sum_{\tau} v_{\tau}^*}$$

Octave equivalence is achieved by taking the difference between the semitone-scale pitch values modulo 12 (one octave), where

$$\langle a \rangle_{12} = a - 12 \lfloor \frac{a}{12} + 0.5 \rfloor$$

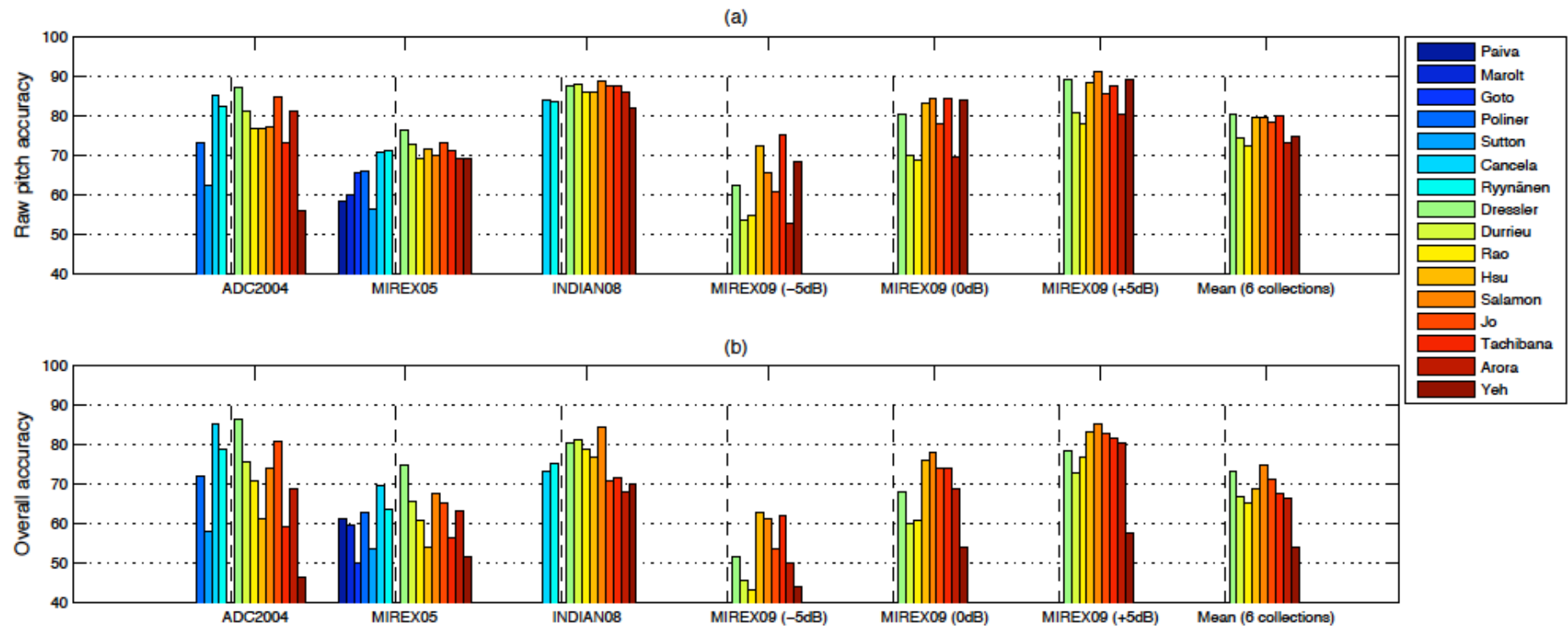
- **Overall accuracy:** this measure combines the performance of the pitch estimation and voicing detection tasks to give an overall performance score for the system. It is defined as the proportion of all frames correctly estimated by the algorithm, where for non-melody frames this means the algorithm labeled them as non-melody, and for melody frames the algorithm both labeled them as melody frames and provided a correct f0 estimate for the melody (i.e. within half a semitone of the ground truth):

$$\text{Acc}_{\text{ov}} = \frac{1}{L} \sum_{\tau} v_{\tau}^* \mathcal{T} [\mathcal{M}(f_{\tau}) - \mathcal{M}(f_{\tau}^*)] + \bar{v}_{\tau}^* \bar{v}_{\tau}$$

where L is the total number of frames

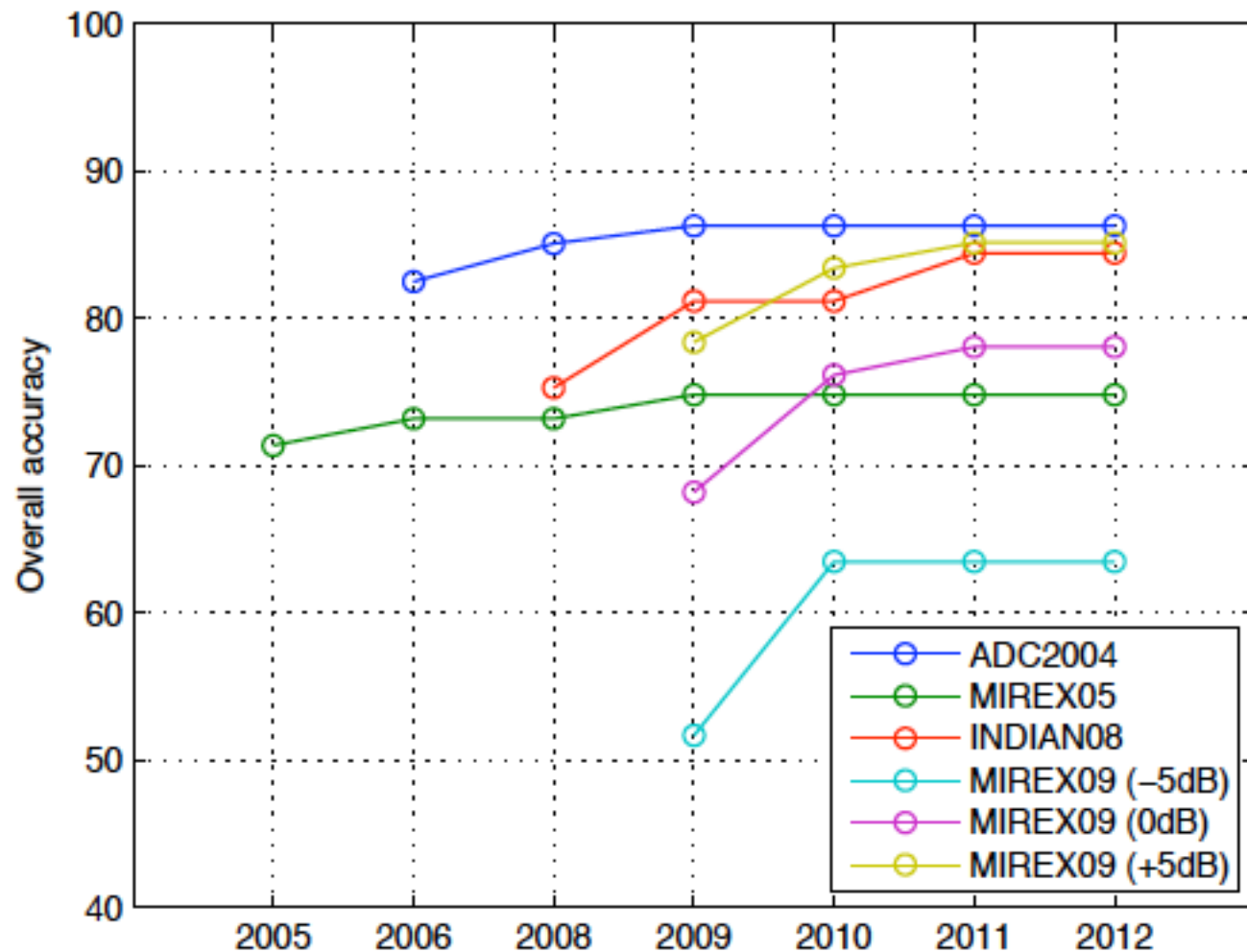
Performance of different systems

Performance for different systems on several annotated databases:



Improvement during the years

Evolution of the best overall accuracy result over the years for the six MIREX collections.



- [1] J. Salamon, E. Gomez, D. P. W. Ellis and G. Richard, "Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges," in IEEE Signal Processing Magazine, vol. 31, no. 2, pp. 118-134, March 2014.
- [2] J.-L. Durrieu, G. Richard, B. David, and C. F´evotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," Trans. Audio, Speech and Lang. Proc., vol. 18, no. 3, pp. 564– 575, March 2010.
- [3] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, "Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source," in IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Mar. 2010, pp. 425–428.