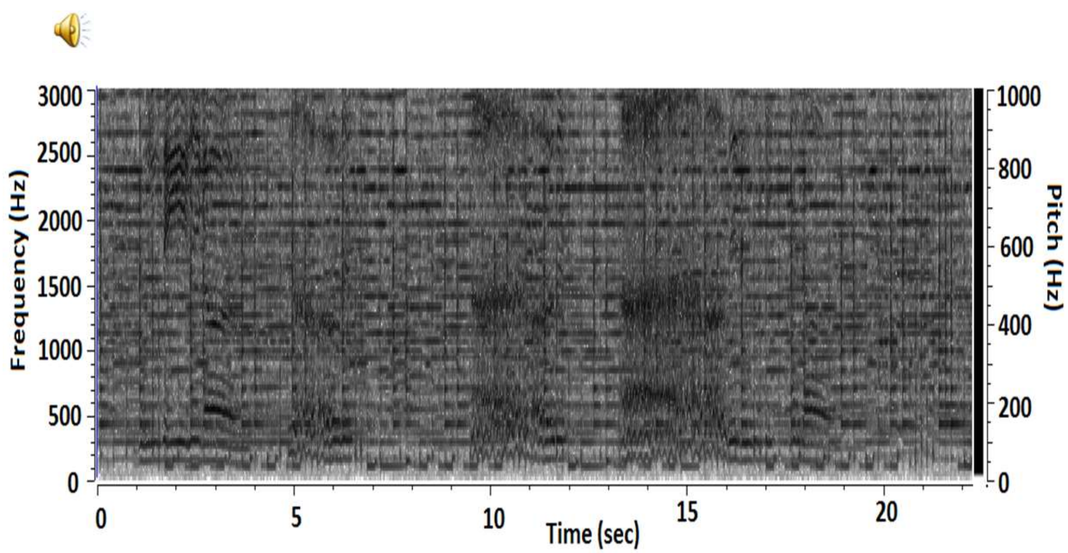
Class 3

- Melody detection in polyphony: how is it different?

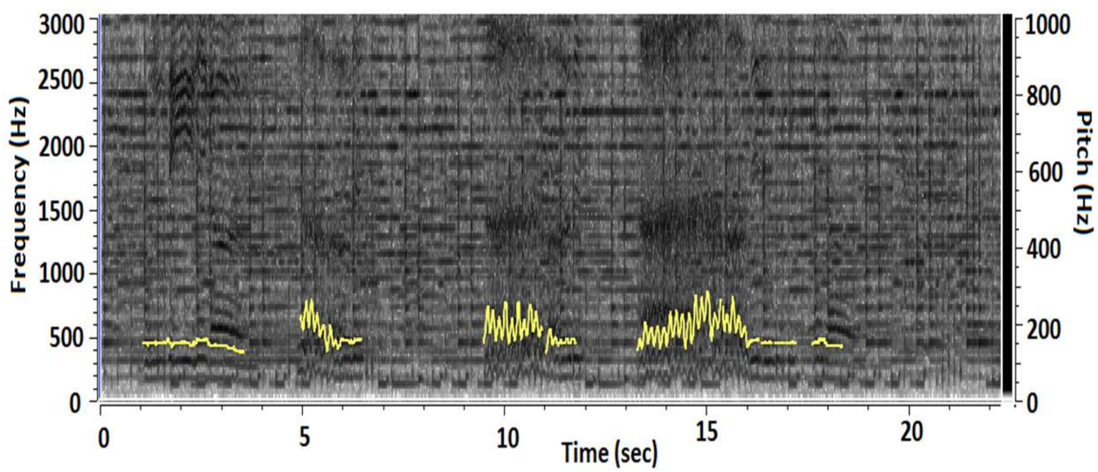
- Audacity: Payoji_Maine vs Jaydev_vibrato (local periodicity in the waveform)

The music



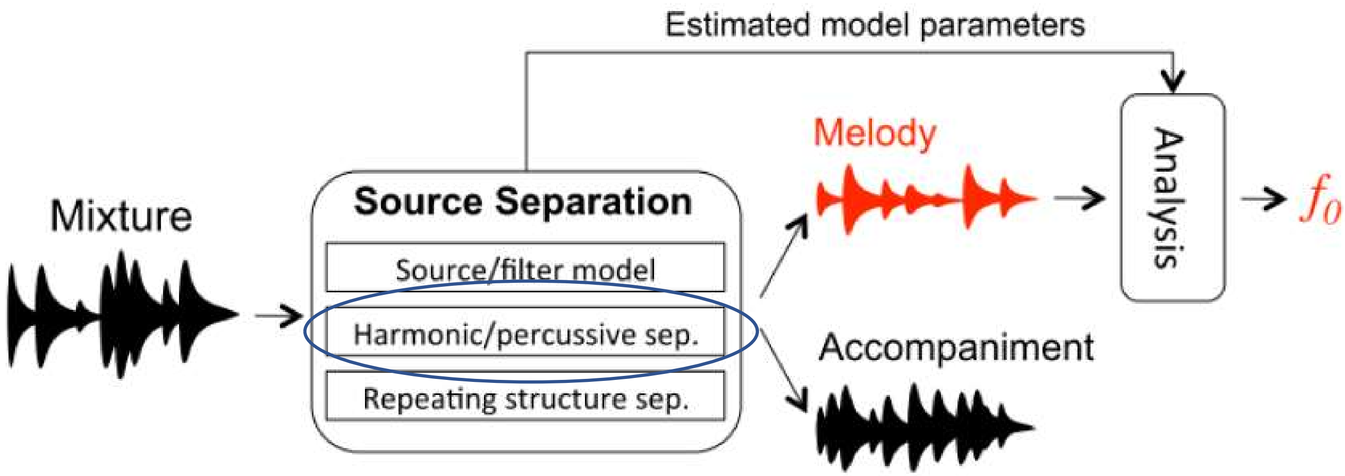
Bhimsen Joshi, Marwa, Tintal
Bandish: Guru Bina Gyan Na
Pave

The "melody"



Bhimsen Joshi, Marwa, Tintal
Bandish: Guru Bina Gyan Na
Pave

Source separation based



G. Richard, Tutorial on Melody Extraction from Polyphonic Signals, 2014

Harmonic-percussive separation

- Harmonic events tend to form horizontal structures and
- Percussive events tend to form vertical structures in a spectrogram.

Violin note + castanets: Median filtering

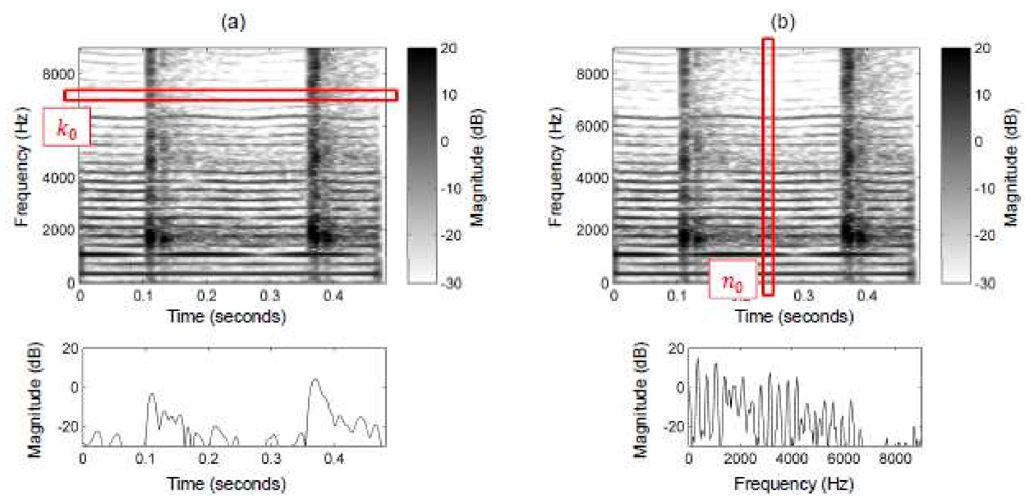


Fig. 8.4 Interpretation of harmonic and percussive components for an audio recording of a note played on a violin superimposed with two click sounds generated by castanets (similar to Figure 8.2). **(a)** Spectrogram $\mathcal{Y}$ and function $\mathcal{Y}^{k_0}$ for some fixed frequency parameter k_0. Percussive events lead to spikes in $\mathcal{Y}^{k_0}$. **(b)** Spectrogram $\mathcal{Y}$ and function $\mathcal{Y}_{n_0}$ for some fixed time parameter n_0. Harmonic events lead to spikes in $\mathcal{Y}_{n_0}$.

Suppressing a component
with median filtering:
dependence on filter length

Short filters
L_h, L_p

Long filters
L_h, L_p

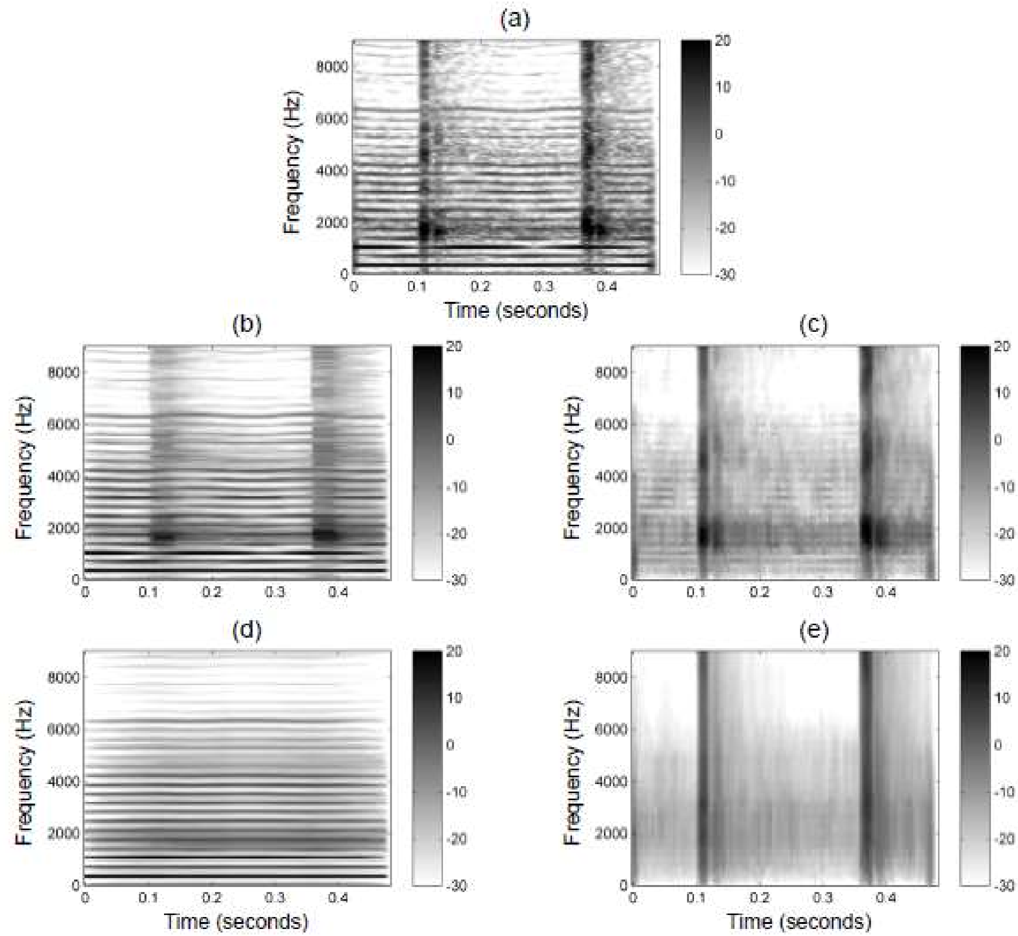


Fig. 8.5 Continuation of the example from Figure 8.4. **(a)** Original spectrogram $\mathcal{Y}$. **(b)** Filtered spectrogram $\tilde{\mathcal{Y}}^{\mathrm{h}}$ using a small length L^{h}. **(c)** Filtered spectrogram $\tilde{\mathcal{Y}}^{\mathrm{p}}$ using a small length L^{p}. **(d)** Filtered spectrogram $\tilde{\mathcal{Y}}^{\mathrm{h}}$ using a large length L^{h}. **(e)** Filtered spectrogram $\tilde{\mathcal{Y}}^{\mathrm{p}}$ using a large length L^{p}.

Masks derived from median-filtered specgrams

Applying mask to get 'separated' spectrograms

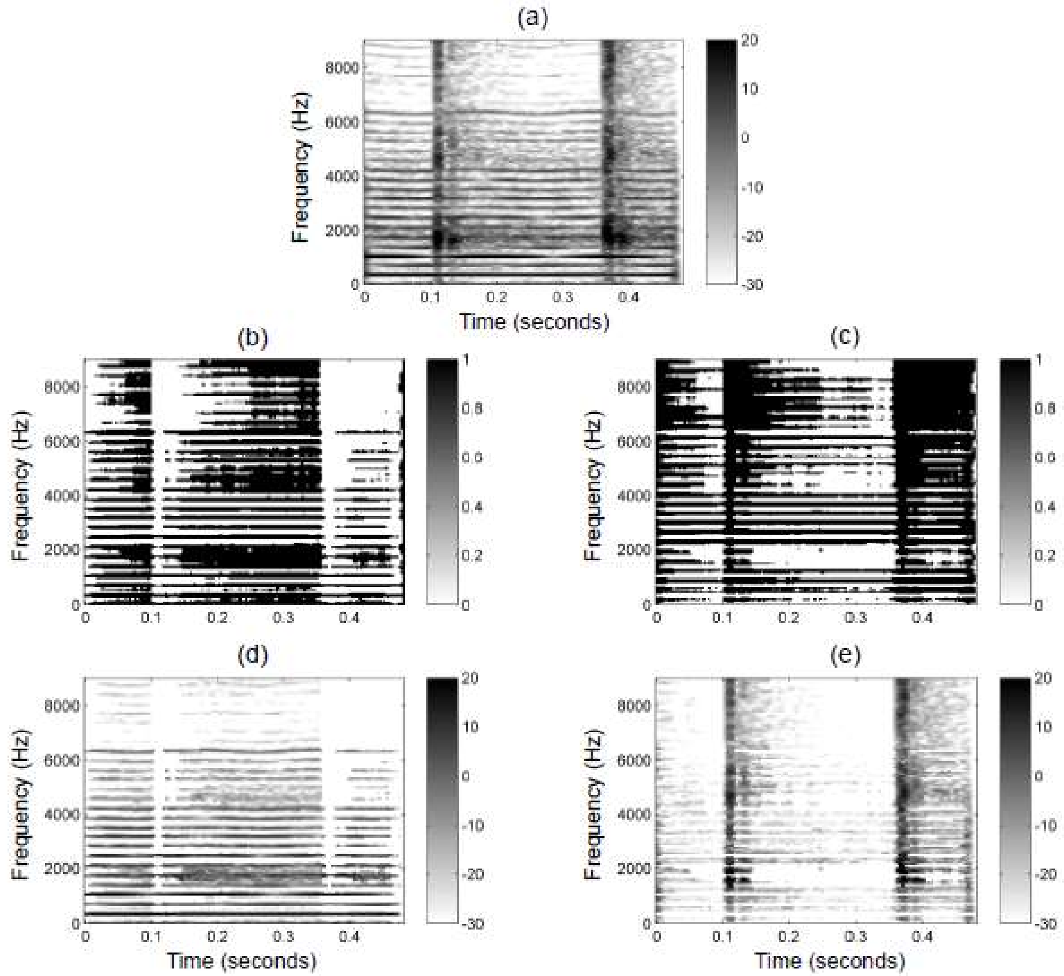


Fig. 8.6 Continuation of the example from Figure 8.5. For computing the masks, the filtered spectrograms from Figure 8.5d and Figure 8.5e are used. (**a**) Original spectrogram $\mathcal{Y}$. (**b**) Binary mask $\mathcal{M}^{\mathrm{h}}$. (**c**) Binary mask $\mathcal{M}^{\mathrm{p}}$. (**d**) Masked spectrogram $\mathcal{Y}^{\mathrm{h}}$. (**e**) Masked spectrogram $\mathcal{Y}^{\mathrm{p}}$.

- Go to the notebook C8S1_HPS.html: show the code (under HPS Implementation). Demo the Section under ‘Experiment: Music Recordings’

- **C8S1_HPS html** """Harmonic-percussive separation (HPS) algorithm

def hps(x, Fs, N, H, L_h, L_p, L_unit='physical', mask='binary', eps=0.001, detail=False):

Notebook: C8/C8S1_HPS.ipynb

Args:

x (np.ndarray): Input signal

Fs (scalar): Sampling rate of x

N (int): Frame length

H (int): Hopsize

L_h (float): Horizontal median filter length given in seconds or frames

L_p (float): Percussive median filter length given in Hertz or bins

L_unit (str): Adjusts unit, either 'pyhsical' or 'indices' (Default value = 'physical')

mask (str): Either 'binary' or 'soft' (Default value = 'binary')

eps (float): Parameter used in soft maskig (Default value = 0.001)

detail (bool): Returns detailed information (Default value = False)

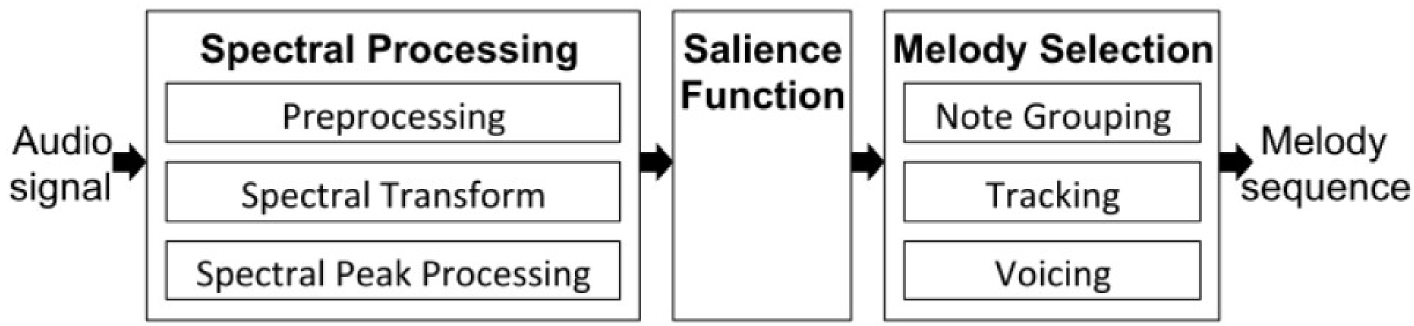
Returns:

x_h (np.ndarray): Harmonic signal

x_p (np.ndarray): Percussive signal

details (dict): Dictionary containing detailed information; returned if ``detail=True``

Salience based melody estimation



G. Richard, Tutorial on Melody Extraction from Polyphonic Signals, 2014

Harmonic summation

A musical tone's energy is not only contained in the fundamental frequency, but spread over the entire harmonic spectrum.

Harmonic sum, computed over all n,k (on zero-padded spectrogram Y)

provides a Salience representation:

$$\tilde{\mathcal{Y}}(n,k) := \sum_{h=1}^{H} \mathcal{Y}(n,k \cdot h)$$

Go to: C8S2_SalienceRepresentation and Fundamental Freq Tracking.html

Log frequency spectrogram

Allows for non-uniform frequency resolution:

- High freq resolution for lower harmonics of low pitches (e.g. to discriminate 1 semitone at 55 Hz = 3 Hz; at 440 Hz, it is 25 Hz)
- High time resolution for high frequency transients.

Audacity: FMP_C1_scale_Cmajor*