TIME-DOMAIN CHROMA EXTRACTION

Marco Fink, Martin Holters, Udo Zölzer

Dept. of Signal Processing and Communications,
Helmut Schmidt University
University of the Federal Armed Forces
Hamburg, Germany
{marco.fink, martin.holters, udo.zoelzer}@hsu-hh.de

ABSTRACT

In this paper, a novel chroma extraction technique called Time-Domain Chroma Extraction (TDCE) is introduced. In comparison to many other known schemes, the calculation of a time-frequency representation is unnecessary since the TDCE is a pure sample-by-sample technique. It mainly consists of a pitch tracking module that is implemented with a phase-locked loop (PLL). A set of 24 bandpass filters over two octaves is designed with the $F0$ output of the pitch tracker to estimate a chroma vector. To verify the performance of the TDCE, a simple chord recognition algorithm is applied to the chroma output. The experimental results show that this novel time-domain chroma extraction technique yields good results while requiring only minor complexity and thus, enables the extraction of musical features in real-time on low-cost DSP platforms.

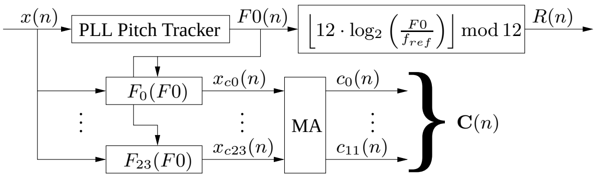

Figure 1: *Time-Domain Chroma Extraction*

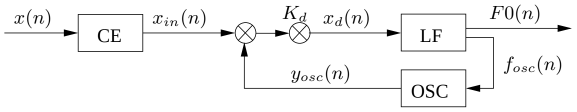

Figure 2: *PLL pitch tracker*

1. INTRODUCTION

The typical application of pitch tracking is to implement pitch-based audio effects, as proposed in [1, 2]. In this paper we suggest to use pitch tracking for typical tasks from the field of Music Information Retrieval (MIR) as well. In this case, a harmonic representation of the input signal is desired. A typical mid-level feature describing harmonic progression over time is the so-called chromagram. It can be used for identifying harmonic relations or temporal structures of musical content.

In Sec. 2, the TDCE system is introduced. The experiments which were used to evaluate the TDCE system, and the corresponding results are presented in Sec. 3. Final thoughts about possible extensions and impairments of the TDCE system are described in Sec. 4.

2. TDCE

The overall system for TDCE is shown in Fig. 1. It consists of two basic modules: the PLL pitch tracker and the chroma extraction module, which is controlled by the $F0$ output of the pitch tracker.

2.1. Pitch Tracker

The pitch tracking module used is based on a PLL with the original purpose of implementing a highly-efficient frequency demodulator [3]. It differs mainly in the used loop filter that modifies the system to form a third-order PLL which is operating in a nonlinear mode. The basic structure of the pitch tracker is shown in Fig. 2. Modulations in the amplitude of a signal disturb the functionality of the PLL. Therefore, the input signal $x(n)$ is divided by its envelope to make it constant within the constant envelope (CE) module. The envelope of the time-domain signal is estimated by lowpass filtering the full-wave rectified input signal $x(n)$.

Fig. 3 visualizes the effect of the CE Module. The upper plot shows the original waveform of a percussive guitar riff with a strong variation in its amplitude. The bottom plot shows the output of the CE module $x_{in}(n)$ with a significantly smaller amplitude variation.

Afterwards, the output of the CE module $x_{in}(n)$ is multiplied with the output of the feedback oscillator $y_{osc}(n)$ and the loop gain coefficient K_d. The feedback oscillator is implemented as a recursive quadrature oscillator where the quadrature component is used for the phase error estimation and the inphase component could optionally be used to implement a lock indicator. The Loop Filter LF consists of a lowpass path $h_{LP}(n)$ and a direct path which together form a second-order shelving filter. The oscillator frequency $f_{osc}(n)$ is obtained by applying the shelving filter while the desired pitch $F0$ is computed by filtering $x_d(n)$ with the lowpass path exclusively and a following multiplication by a factor of 2. This can be written as

$$F0(n) = [(x_{in}(n) \cdot y_{osc}(n) \cdot K_d) * h_{LP}(n)] \cdot 2 \,. \qquad (1)$$

More details about the configuration, functional limits, and resulting properties of the pitch tracker can be found in [4].

2.2. Chroma Extraction

A chroma vector is a simple mid-level representation of the harmonic structure of a time-domain signal $x(n)$. Most known chroma

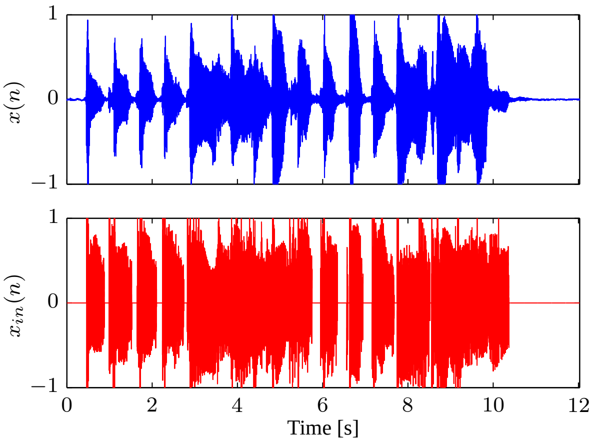

Figure 3: *Waveform of percussive guitar riff without (top) and with (bottom) constant envelope processing*

computation methods require a frequency representation with a sufficient frequency resolution. Several octaves are folded into a single octave and the corresponding pitch classes are directly extracted from this spectral representation with a set of bandpass filters, which are tuned to fixed values of the 12 semitones of western music. Typically, several octaves (e.g., 3-8. see, [5]) are considered to compute a chroma vector. We assume that the pitch tracking module tracks the frequency with the strongest amplitude. Therefore, it is sufficient to consider only the two octaves above the fundamental frequency $F0$ since these should include the relevant harmonic information. The extraction process is usually applied in the frequency domain. In contrast, we compute the chroma vector $\mathbf{C}$ with a parallel set of second-order recursive bandpass filters $F_i(F0)$ that are controlled with the $F0$ output of the pitch tracker (see Fig. 1). The center frequencies fc_i of the filters $F_i(F0)$ are defined by

$$f_{c_i}(F0) = F0 \cdot 2^{\frac{i}{12}},\ i \in [0, \dots, 23]. \tag{2}$$

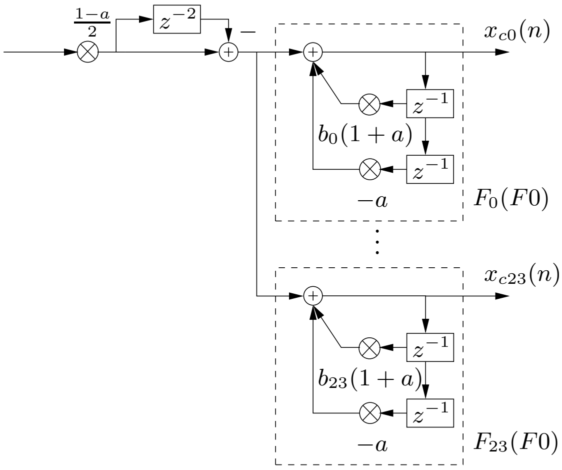

Figure 4: *Signal flow diagram of chroma filter implementation*

The bandpass filters are implemented with the transfer function given by [6]

$$H_{Fi}(z) = \frac{1-a_i}{2}\,\frac{1-z^{-2}}{1-b_i(1+a_i)z^{-1}+a_iz^{-2}}, \tag{3}$$

where the coefficients a_i and b_i are computed by

$$\begin{aligned}
a_i &= \frac{1-\sin(\Omega_{bi})}{\cos(\Omega_{bi})},\\
b_i &= \cos(\Omega_{ci}),\\
B_{wi} &= \frac{f_{ci}(F0)}{Q},\\
\Omega_{ci} &= 2\pi\frac{f_{ci}(F0)}{f_s},\\
\Omega_{bi} &= 2\pi\frac{B_{wi}}{f_s}.
\end{aligned}$$

B_{wi} is the bandwidth of the filter, Q is the quality factor of the bandpass filters and f_s is the sampling frequency. Note that a_i is assumed to be relatively constant within two octaves. Therefore, only $a = a_1$ is used in the following. The complete chroma filter implementation is shown in Fig. 4. The corresponding frequency responses are depicted in Fig. 5.

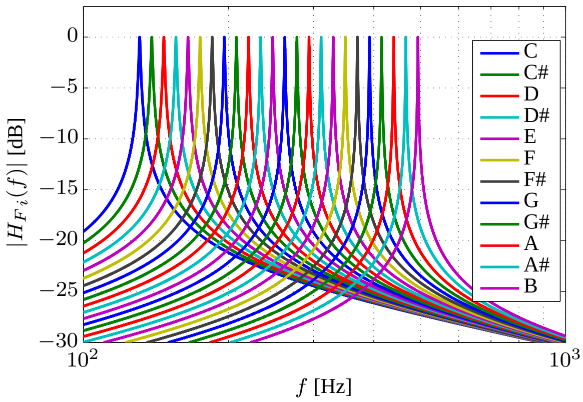

Figure 5: *Frequency responses* $|H_{Fi}(f)|$ for $F0 = 130.82$ Hz, $Q = 150$ and $f_s = 44100$ Hz

It is apparent that only the recursive part of the filter, which consists of two multiplications in the feedback paths, has to be computed for every semitone per sample. The non-recursive part of the filter is a single multiplication and an addition with a twice delayed sample. Since two octaves are analyzed using 12 filters per octave, the total number of multiplications for the chroma filterbank adds up to 49.

To approximate the power of the present chroma values $c_i(n)$, the average of the absolute filter output values $x_{ci}(n)$ of L samples is computed within the moving average module (MA in Fig. 1) by

$$c_i(n) = \frac{1}{L}\sum_{k=n-L+1}^{n} |x_{ci}(k) + x_{ci+12}(k)|\,. \tag{4}$$

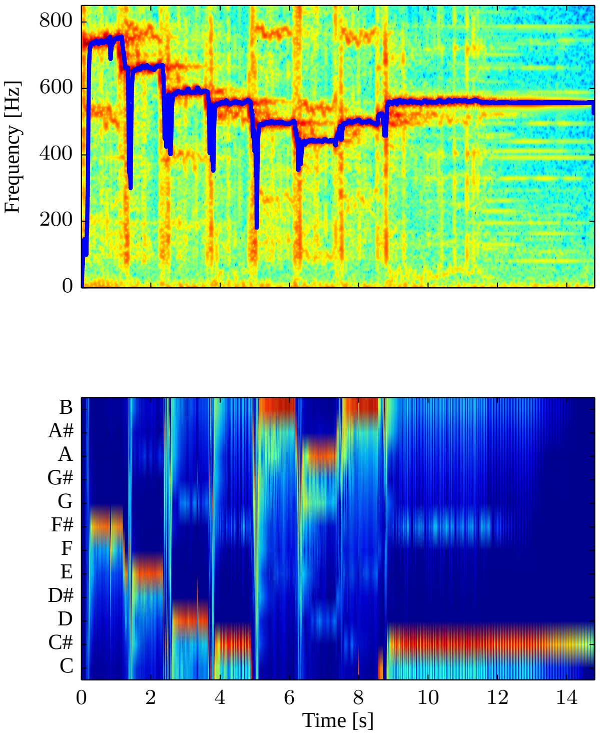

Figure 6: *$F0$ track on top of spectrogram (top), and chromagram (bottom) of a lead guitar riff*

3. EXPERIMENTS AND DISCUSSION

To evaluate the performance of the proposed chroma extraction scheme the authors choose to apply typical MIR tasks. Namely, the TDCE system was used to realize transcription, chord recognition, and onset detection. The first test analyses the capability of transcription with a real signal. Therefore, a simple lead guitar riff consisting of 8 notes was recorded. The spectrogram and the $F0$ track in the upper plot of Fig. 6 illustrate the resulting changes of frequency over time. The $F0$ track also shows that the melody was played with a slight vibrato. Although the tracked frequency $F0$ varies, the corresponding chromagram in the bottom plot of Fig. 6 remains stable and thus, allows the correct identification of the played notes. This resistance against pitch variations can be explained by the fact that the center frequency of the chroma extraction filters are not fixed but depend on the tracked fundamental frequency.

To identify reasonable parameters for the amount of considered octaves and the quality factor of the chroma Filters $F_i(F0)$ a synthesized guitar chord sequence was processed with the TDCE system with varying system parameters except the filter length of the MA module which was fixed to $L = 32$ for all experiments. Fig. 7 shows the chromagrams for different Q factors of the chroma extraction filters and a different number of analyzed octaves. It is clearly apparent that a higher Q factor of about 150 is beneficial in terms of clear chromagrams. Many more noise-like artifacts are involved in the chromagrams for $Q = 50$, due to an inappropriate setting of the filter bandwidth. Fig. 7 also demonstrates nicely that a single octave is not sufficient for the application of chord recognition. The first played chord is a B minor, consisting of the root B, minor third D, and perfect fifth F#. The note D, defining the chordal quality minor, is played in the second octave above the root note in the standard barre chord. Therefore, the minor third D of the first chord (marked magenta in Fig. 7) is only visible in the chromagrams on the right side, which were created using two octaves.

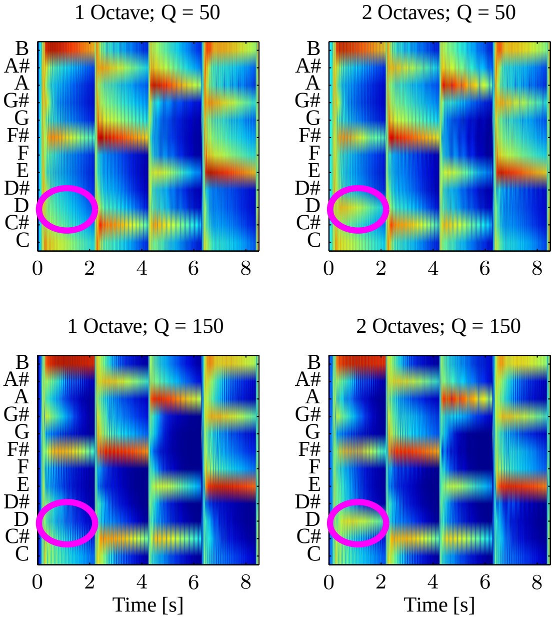

Figure 7: *TDCE chromagrams for Q = [50,150] and for a single and two octaves above $F0$*

The most typical application for chroma vectors is chord recognition. Several techniques to map chroma vectors to chord labels are known. In this work, a simple template-based approach as proposed in [7] is used.

A set of binary chord templates M for different chordal qualities is defined. For example, a C major chord consists of the three notes C, E and G. The resulting binary chord template leads to

$$M_{maj} = [1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0],$$

where the first coefficient represents the base note C. To find the best match between chroma vector and chord template, a normalized vector difference between the chroma vector and every binary chord template is calculated. The template with the smallest difference to the current chroma vector is assumed to represent the present chordal quality M. Only templates for triad chords, namely major, minor, augmented and diminished are used for these experiments.

To identify the corresponding root note of the chord, the index R of the pitch class vector

[C, C#, D, D#, E, F, F#, G, G#, A, A#, B] can directly be calculated from the $F0$ value as proposed by [7]

$$R = \left\lfloor 12 \cdot \log_2 \left(\frac{F0}{f_{ref}} \right) \right\rfloor \bmod 12, \tag{5}$$

where f_{ref} is C_2 (65.41 Hz). This choice of f_{ref} allows full usability even for lower, dropped guitar tunings. Finally, the base tone R and the chordal quality M can be combined to describe the harmonic structure at time point n with the common chord syntax.

The next experiment shall demonstrate the chord recognition capability of the proposed system. Fig. 8 illustrates the time-domain signal (a), the corresponding $F0$ track on top of the spectrogram (b) of the same test item as in the last experiment. It can be seen that the pitch tracker follows the dominant tones in the spectrum very accurately. All root notes of the chords are identified correctly, although the pitch tracker tends to preferably identify harmonics instead of the fundamental frequencies (e.g., second chord). In the case of a transient event, the strum of a chord in this example, the TDCE seems to cause erroneous $F0$ results in form of large deviations of the pitch track. Fig. 8c displays the resulting chromagram implemented with the method explained in Sec. 2.2. The base tones that are computed with Eq. 5 are visualized in Fig. 8d. The location within the diagram represents the pitch class of the base tone, whereas the color indicates the chordal quality, calculated with the method explained above. Since the $F0$ track is almost flawless, the computation of the base tones is also very accurate. Also the identification of the chordal quality is stable to the same degree.

As another test, a sequence of arpeggios (sequentially played chords) is generated. As before, the time-domain signal, spectrogram, and $F0$ track are plotted in Fig. 9a and Fig. 9b. Instead of visualizing the chromagram, the absolute derivative of the $F0$ track exceeding a certain threshold thr is shown in Fig. 9c. This modification of the original $F0$ track is a good indicator for the rhythmic structure of the source signal. In other words, also onset detection can be realized with the TDCE system. In this case the notes are picked in straight eighths. Fig. 9d shows the nearly flawless transcription of the synthesized arpeggios.

4. CONCLUSIONS

A novel method to compute a chroma vector was presented. The most significant difference to known techniques is the TDCE's property of being a time-domain method. Therefore, typical problems of frequency-domain methods like insufficient frequency resolution, spectral leakage, memory and computation resources to calculate the transformations, etc. can be neglected. Furthermore, only a small delay caused by the attack time of the PLL and the length L of the moving average filters is introduced, whereas block-based methods introduce at least a delay of their block length. Hence, TDCE might be a reasonable alternative to realize typical MIR tasks on cheap, energy-saving real-time DSP systems. The applicability of TDCE was demonstrated with three small experiments: Basic chord recognition, note transcription and onset detection. In addition, the TDCE could be used in many other applications with a musical context that have to be investigated in a next step. The experiments revealed some weaknesses of the system. Transient signal conditions led to incorrect pitch tracking result. The authors assume that this faulty behaviour can be avoided by increasing the robustness of the CE and LF modules. In addition, in some cases the pitch tracker tended to track harmonic frequencies instead of the fundamental ones. Hence, a mechanism supporting the PLL in finding the fundamental frequency has to be found.

5. REFERENCES

[1] U. Zölzer, "Pitch-based Digital Audio Effects," in *5th Int. Symposium Communications, Control and Signal Processing (ISCCSP)*, Rome, May 2012.

[2] U. Zölzer, *Digital Audio Effects*, J. Wiley & Sons, New York, USA, 2nd edition, 2011.

[3] F. Gardner, *Phaselock Techniques*, J. Wiley & Sons, New York, USA, 3nd edition, 2005.

[4] U. Zölzer, S. Sankarababu, and S. Möller, "PLL-Based Pitch Detection and Tracking for Audio Signals," in *Proc. Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Piraeus, July 2012.

[5] M. Goto, "A Chorus-section Detecting Method for Musical Audio Signals," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '03)*, April 2003, vol. 5, pp. 437–440.

[6] U. Zölzer, "Digital Audio Signal Processing - Filters Web Summary," Available at `http://www.hsu-hh.de/download-1.4.1.php?brick_id=6NHR15i6H9dn5YjK`, accessed March 28, 2012.

[7] T. Fujishima, "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music," *Proc. ICMC, 1999*, pp. 464–467, 1999.

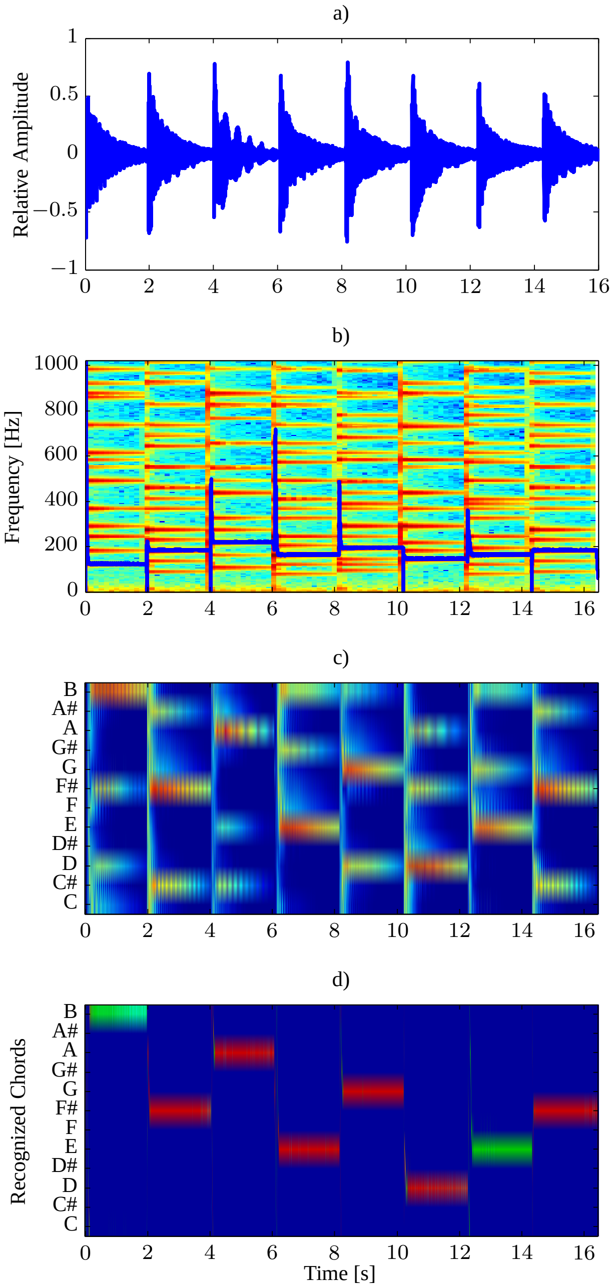

Figure 8: *Chord sequence with a) time-domain signal, b) $F0$ track on top of spectrogram, c) chromagram, and d) recognized chords. Color red indicates major, green minor and cyan a single detected note*

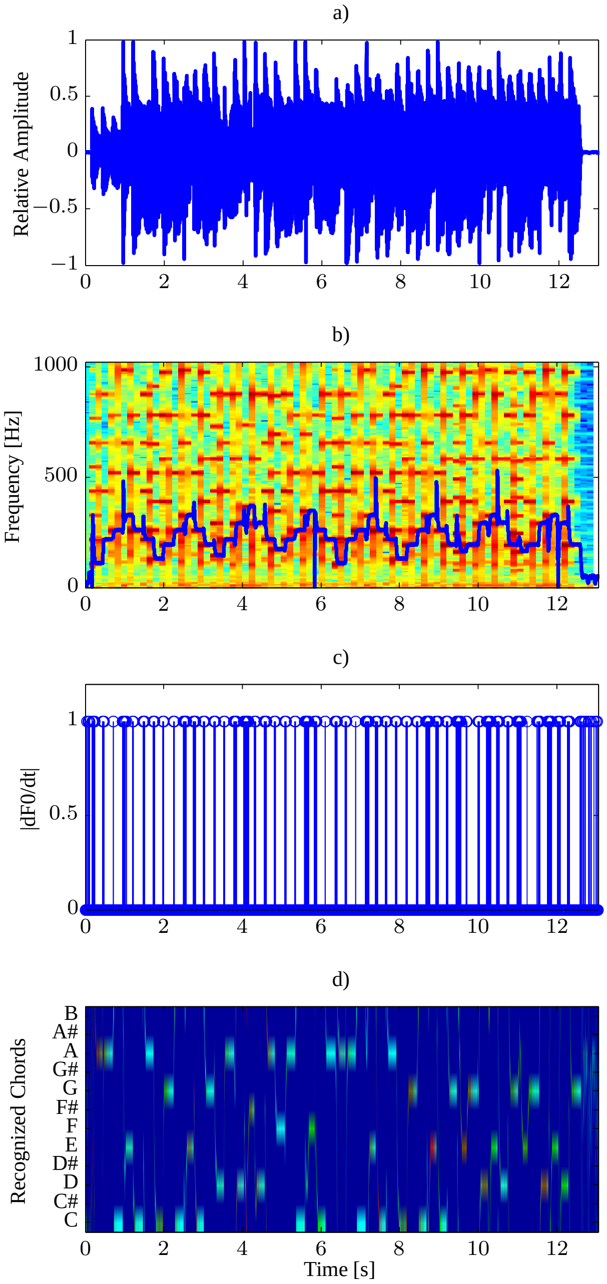

Figure 9: *Arpeggio sequence with a) time-domain signal, b) $F0$ track on top of spectrogram, c) absolute derivative of $F0$ track, and d) recognized sequence of notes*