Onset Detection

Corinne Darche

MUMT501: Digital Signal Processing

McGill University

April 20, 2022

What is an Onset?

Attack: Temporal interval in which the amplitude envelope increases

Transient: Interval when the signal changes unpredictably before stabilizing (steady-state)

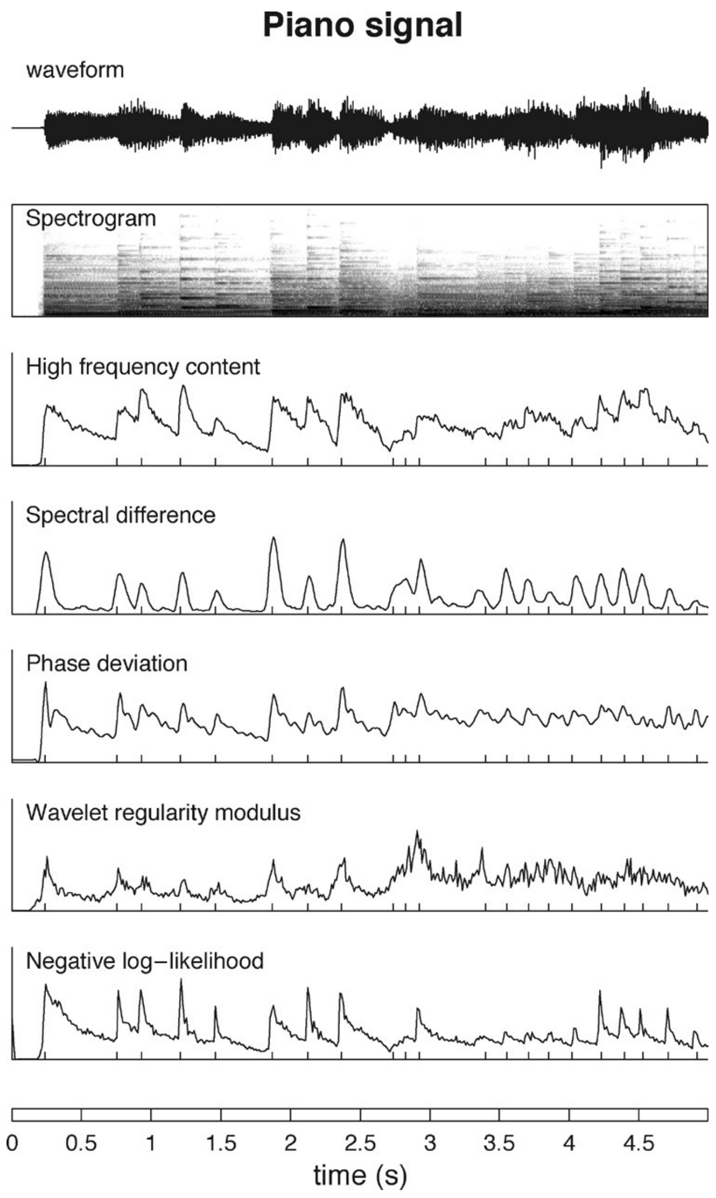
Onset: Instant in which the transient begins

Elements of an Onset Detection Algorithm

1. Preprocessing
    - Optional, but yields better results (separate signal into multiple frequency bands, isolate the transient)
2. *Reduction (Detection Function)*
3. *Postprocessing*
4. *Peak-picking*

Reduction

- Subsample the signal and pass it through detection functions to find the onset
- Signal Features
  - Temporal characteristics (amplitude increase)
  - Spectral features (STFT)
  - Phase deviation
- Probabilistic Model
  - Sequential probability
  - “Surprises”
- Some work better for certain sounds



From Bello et al.

Postprocessing and Peak-Picking

- **Postprocessing**
  - Improve detection function values for more accuracy in peak-picking
  - Smooth through filtering, normalization
- **Peak-Picking**
  - Select peaks above threshold, use parabolic interpolation
  - Can be further refined and pruned (ex.: adaptive thresholding)

Scherrer and Depalle: "Onset Time Estimation for the Exponentially Damped Sinusoids Analysis of Percussive Sounds"

- Goal: create an algorithm to detect onsets in pitched percussive sounds (ex.: piano, marimba)
- Detection Algorithms:
  - Frequency-domain detection function (rough estimate)
    - Start with STFT - $X[l,b] = \sum_{n=0}^{N-1} w[n]\, x[n+lH] e^{\frac{j2\pi nb}{N}}$ $with\ b \in [0; N-1]$
    - Compare each STFT frame - $d_f[l] = \sqrt{\sum_{b=0}^{\frac{N}{2}} (|X[l,b]| - |X[l-1,b]|)^2}$
  - Time-domain detection function (refined estimate)
    - Make up for delay by performing detection a few hop sizes before each rough estimate
    - $d_t[n] = \frac{1}{J}\log\left(\frac{\sum_{m=n+1}^{n+J} x^2[m]}{\sum_{l=n-J}^{n-1} x^2[l] + v}\right) \cdot \sum_{k=n+1}^{n+J} x^2[k]$

Scherrer and Depalle: “Onset Time Estimation for the Exponentially Damped Sinusoids Analysis of Percussive Sounds”

- Post-processing through zero-mean, normalizing, and smoothed with a normalized derivative filter
- Peak-picking above threshold α dB
- Adaptive thresholding and pruning applied to peaks
  - Adaptive thresholding to set a minimum value for a threshold at a given index
  - Prune “repeated” onsets (i.e. onsets within a given number of samples)
- Algorithm tested on synthetic and real sounds

Variables for Rough and Refined Onsets

- N: FFT size
- H: Hop size
- J: Number of samples for Eq. 3
- v: Regularization factor for Eq. 3
- γ: Coefficient for the normalized derivative filter at the post-processing stage
- τ: Absolute threshold for the adaptive threshold
- p: Order for the median filter
- l: Control for how much the median filtered function impacts the absolute threshold
- α: Threshold for peak-finding using parabolic interpolation (in dB)
- I: Number of samples used for pruning repeated onsets

Defining the Parameters

Synthetic Sounds

Rough onsets	**Refined onsets**
$N : 2048$	$J : 200$
$H : 1024$	$\upsilon : 10^{-4}$
$\gamma : 0.3$	$\gamma : 0.1$
$\tau : 0.1$	$\tau : 0.5$
$p : 5$	$p : 5$
$\ell : 0.5$	$\ell : 0.5$
α : 6dB	α : 6dB
$I : 900$	$I : 900$

Real Sounds (Guitar)

Rough onsets	**Refined onsets**
$N : 1024$	$J : 400$
$H : 512$	$\upsilon : 10^{-4}$
$\gamma : 0.3$	$\gamma : 0.1$
$\tau : 0.15$	$\tau : 0.5$
$p : 5$	$p : 5$
$\ell : 0.5$	$\ell : 0.5$
α : 6dB	α : 6dB
$I : 2205$	$I : 900$

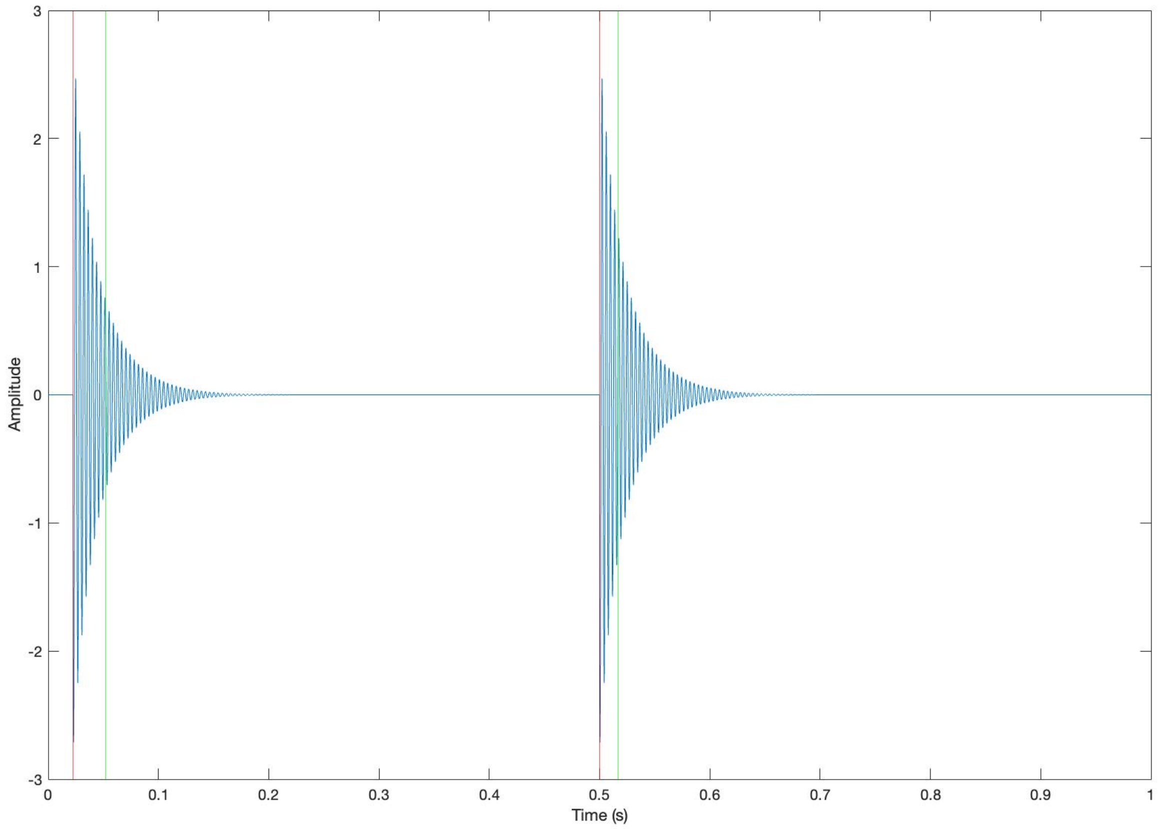
My Implementation: The Algorithm

- Recreated Scherrer and Depalle's algorithm in MATLAB
  - Equations detailed and each step outlined clearly
- Hanning window size for STFT → FFT size
- Pruning algorithm tricky to implement
  - Needed nested while loops for dynamic algorithm that adjusted as array changed size
- Sound files linked in the paper
  - Unfortunately, the music.mcgill.ca link is out of commission

My Implementation: Synthetic Sounds

- Used a combination of different exponentially damped sinusoids to create a signal with multiple partials and transients
  - $x_{EDS}[n] = Ae^{-an}\cos(2\pi f t + \phi)$
- Created a signal with 5 partials
  - Frequencies chosen to recreate C4 piano note
  - Each partial: two EDS with slightly different values of A, a, f, and phi
  - Two transients
    - Make it easy to visually determine the onsets by the waveform (no noise added, no overlapping transients)

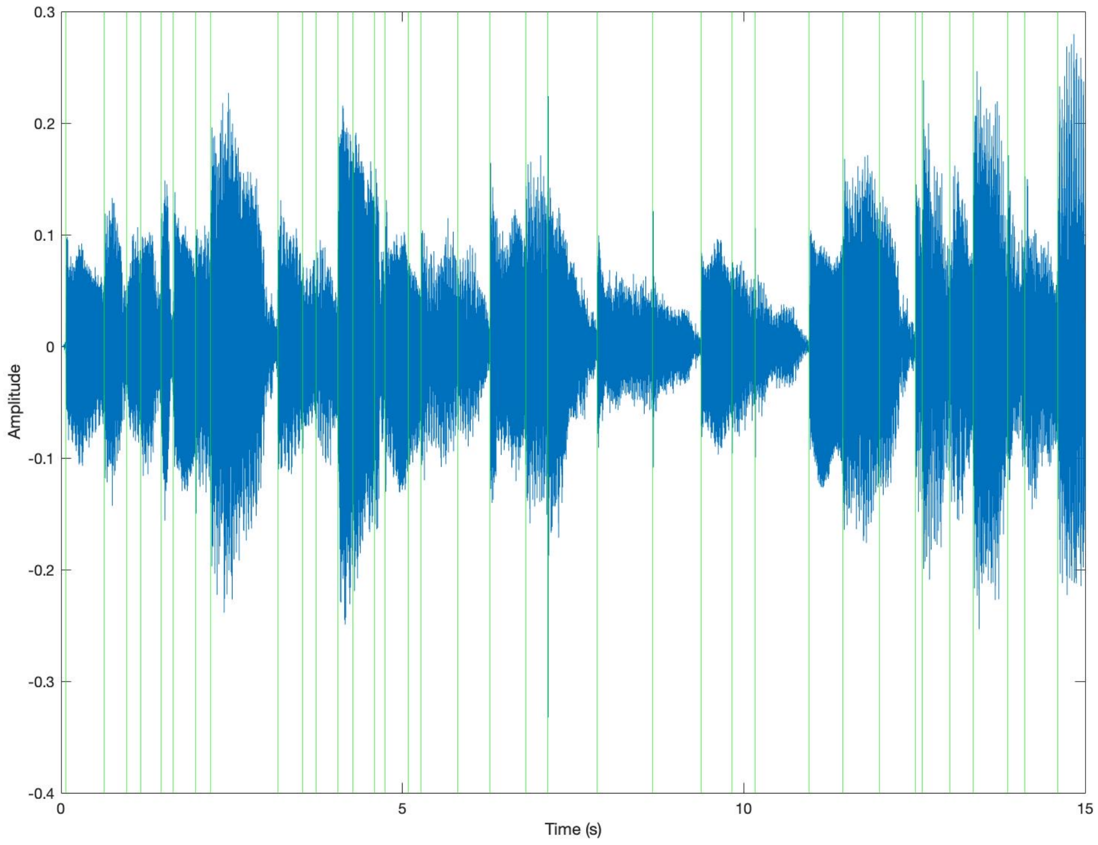
My Implementation: Synthetic Sound Results



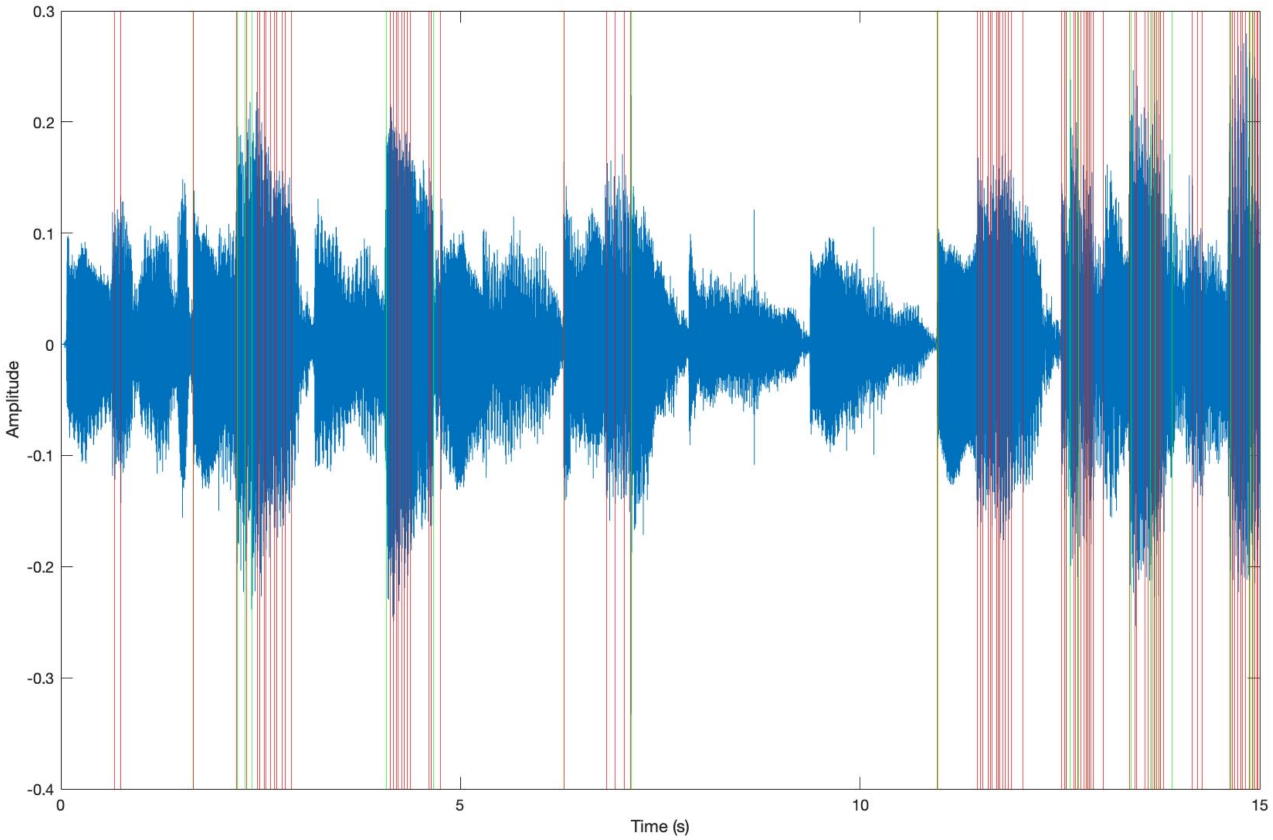
My Implementation: Real Sounds

- Used the sound file “guitar2.wav” from the Leveau dataset
  - External link mentioned briefly in the paper still worked!
  - Includes files with the correct onset annotations for easy answer-checking
- Paper didn’t specify which sound file was used when tuning parameters
  - Expecting inaccuracies in results

My Implementation: Real Sounds Expectation



My Implementation: Real Sound Results



In Summary…

- Onset detection algorithms are built from a similar foundation and adjusted based on the type of sound
- We looked at a specific implementation designed for percussive sounds
- Paper was very helpful in recreating the algorithm structure and design
  - Couldn’t access the specific synthetic sound files used
- Using paper parameters…
  - Accurate and reliable for synthetic sounds
  - Real sounds require extensive parameter tuning

References

J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler, “A Tutorial on Onset Detection in Music Signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, Sep., pp. 1035-1047, 2005.

B. Scherrer and P. Depalle, “Onset Time Estimation for the Exponentially Damped Sinusoids Analysis of Percussive Sounds,” In Proc. of the 17th International Conference on Digital Audio Effects, 2014, pp. 1-7.

S. Dixon, “Onset Detection Revisited,” In Proc. of the 9th International Conference on Digital Audio Effects 2006, pp. 1-6.

Thank you!

Any questions?