Speech Enhancement

(/ Removal)

Amit Eliav

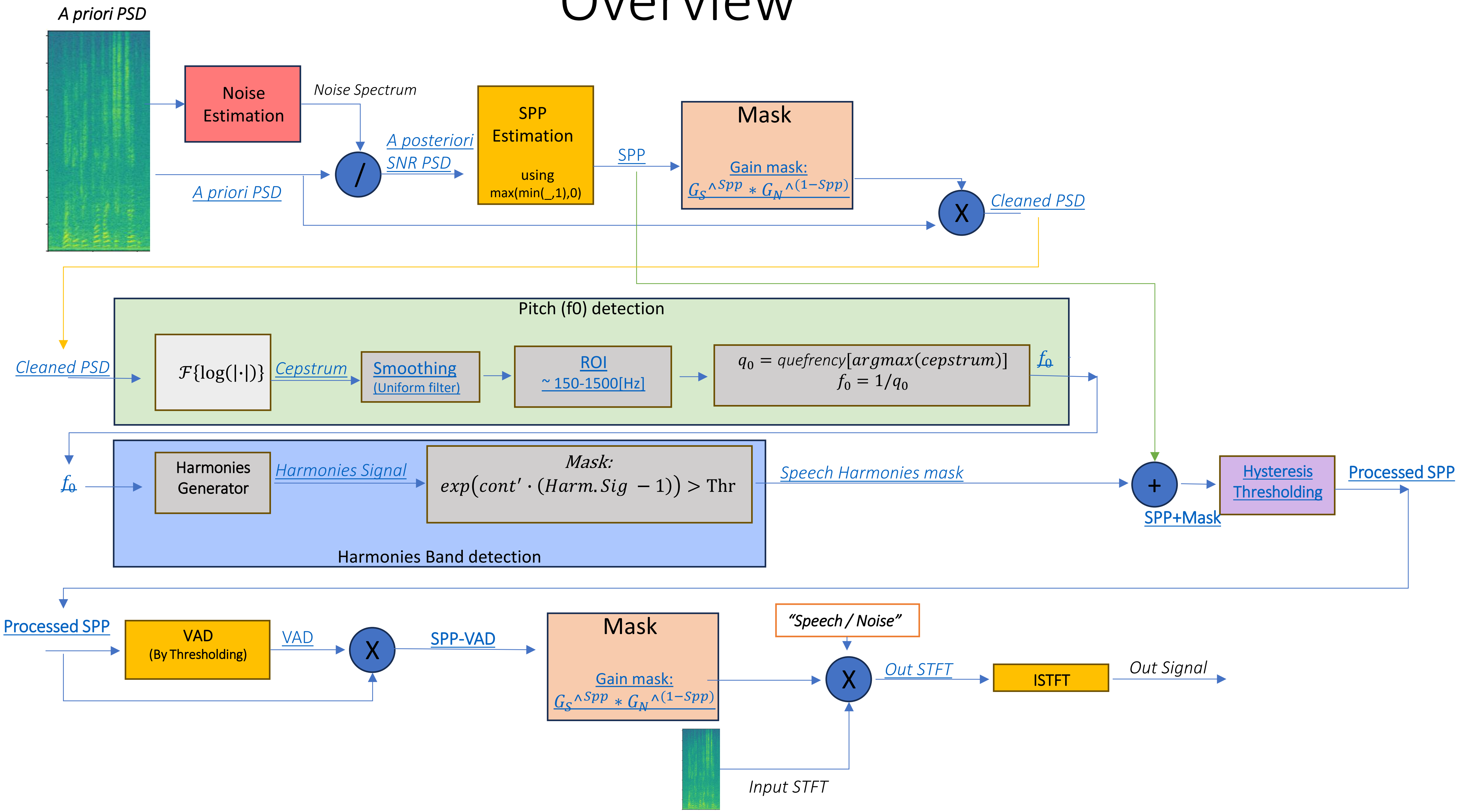
04.2024

Speech Enhancement / Removal Tool

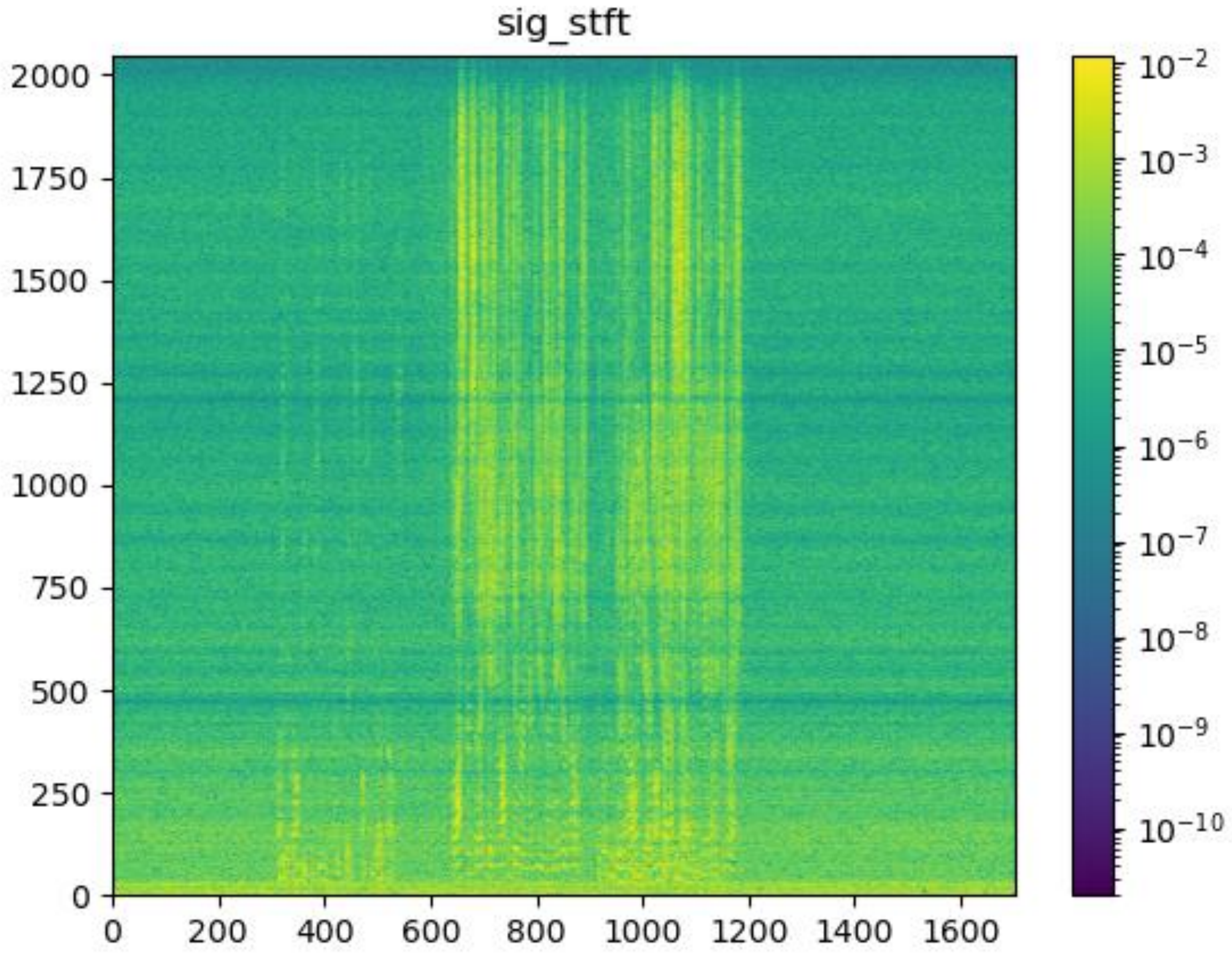
- Analytic speech enhancement or removal algorithm

- Based on the following steps:
  - Noise estimation
  - SPP estimation
  - Pitch tracking
  - Harmonies-based detection and masking

Overview

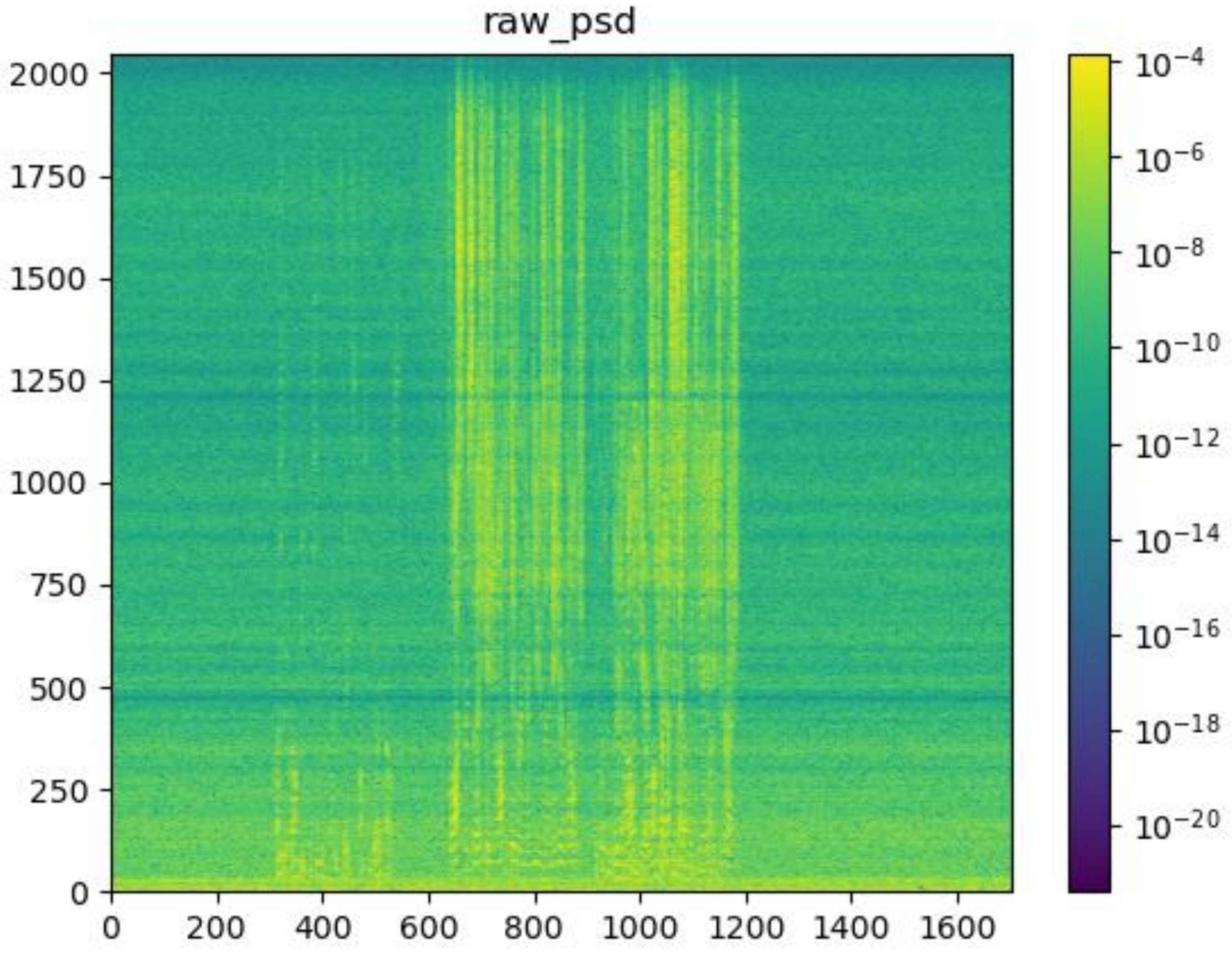

A priori PSD
Noise Estimation
Noise Spectrum
A posteriori SNR PSD
A priori PSD
SPP Estimation using max(min(_,1),0)
SPP
Mask
Gain mask:
G_S^Spp $*$ G_N^$(1-Spp)$
X
Cleaned PSD
Pitch (f0) detection
Cleaned PSD
$\mathcal{F}\{\log(|\cdot|)\}$
Cepstrum
Smoothing
(Uniform filter)
ROI
~ 150-1500[Hz]
$q_0 = quefrency[argmax(cepstrum)]$
$f_0 = 1/q_0$
f_0
f_0
Harmonies Generator
Harmonies Signal
Mask:
$exp(cont' \cdot (Harm.Sig - 1)) > \mathrm{Thr}$
Harmonies Band detection
Speech Harmonies mask
+
SPP+Mask
Hysteresis Thresholding
Processed SPP
Processed SPP
VAD
(By Thresholding)
VAD
X
SPP-VAD
Mask
Gain mask:
G_S^Spp $*$ G_N^$(1-Spp)$
“Speech / Noise”
X
Out STFT
ISTFT
Out Signal
Input STFT

Input Spectrogram



Overview

Input PSD



Overview

A posteriori SNR PSD

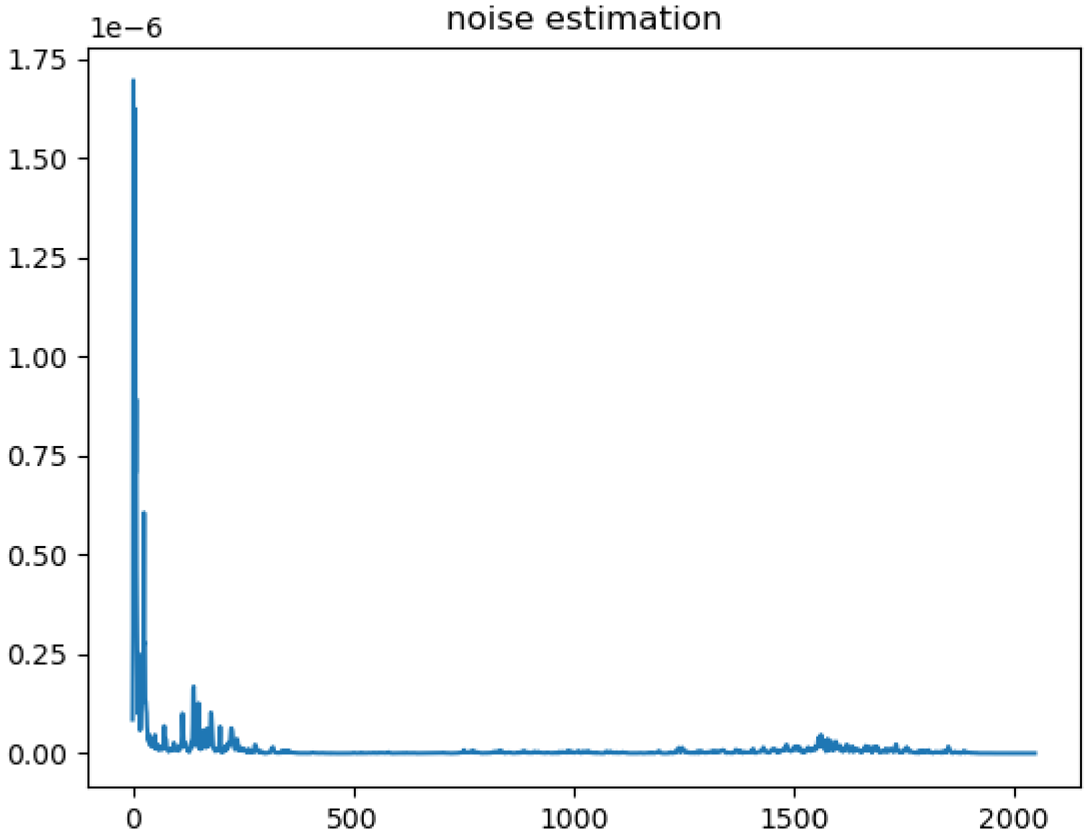
Estimated Noise Spectrum

Overview

Using simple method of the most frequent value.

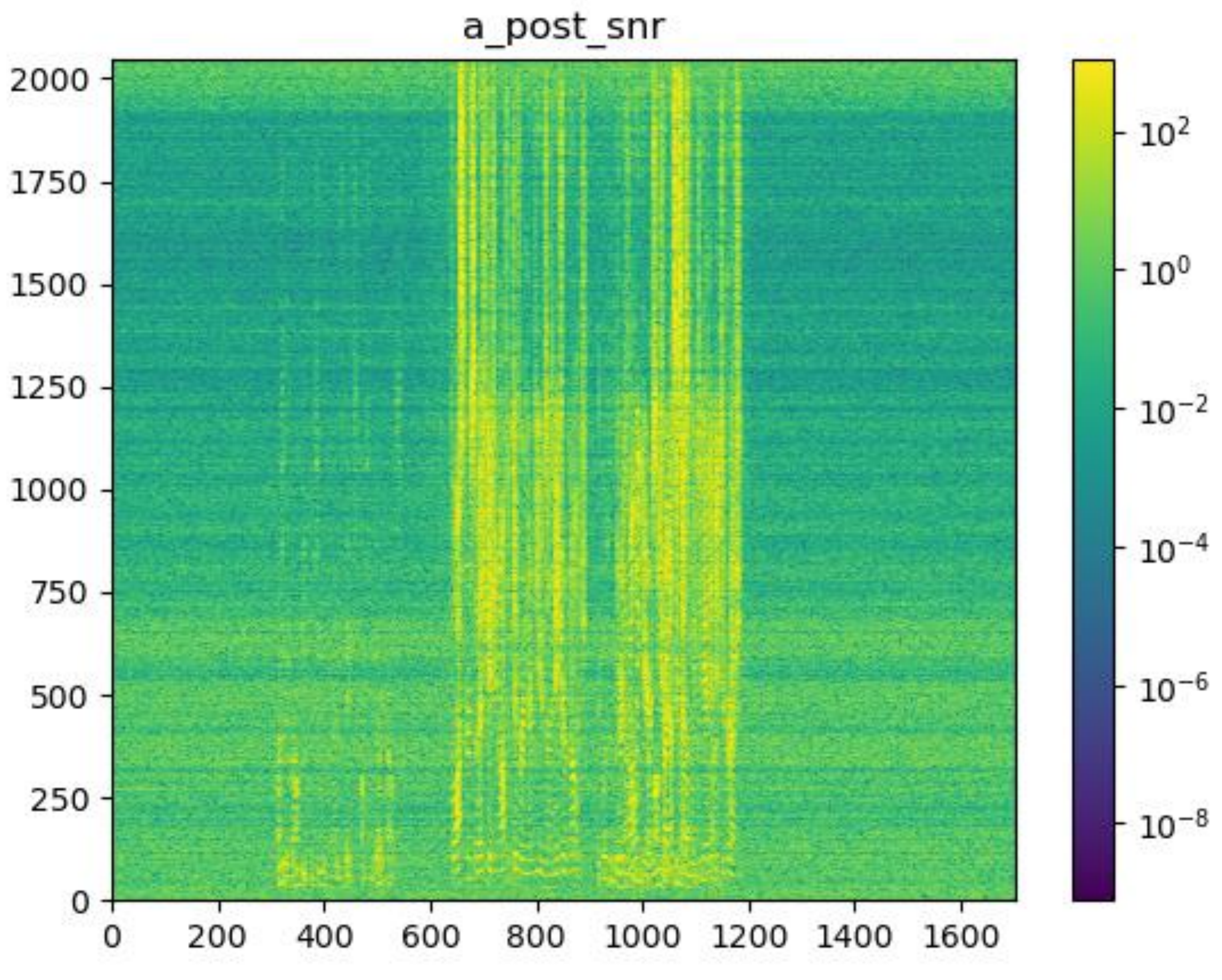
For each frequency bin we take the histogram, and find the argmax(pdf) and return the corresponding value

We can change this block to a more complex estimator such as **MCRA**




```
pdf, power_bin_edges = np.histogram(x, bins=bins)
power_axis = (power_bin_edges[1:]+power_bin_edges[:-1])/2  # calculates the center points of the bins
mode_power = power_axis[pdf.argmax()]  # this is a scalar
return mode_power
```

A posterior SNR PSD



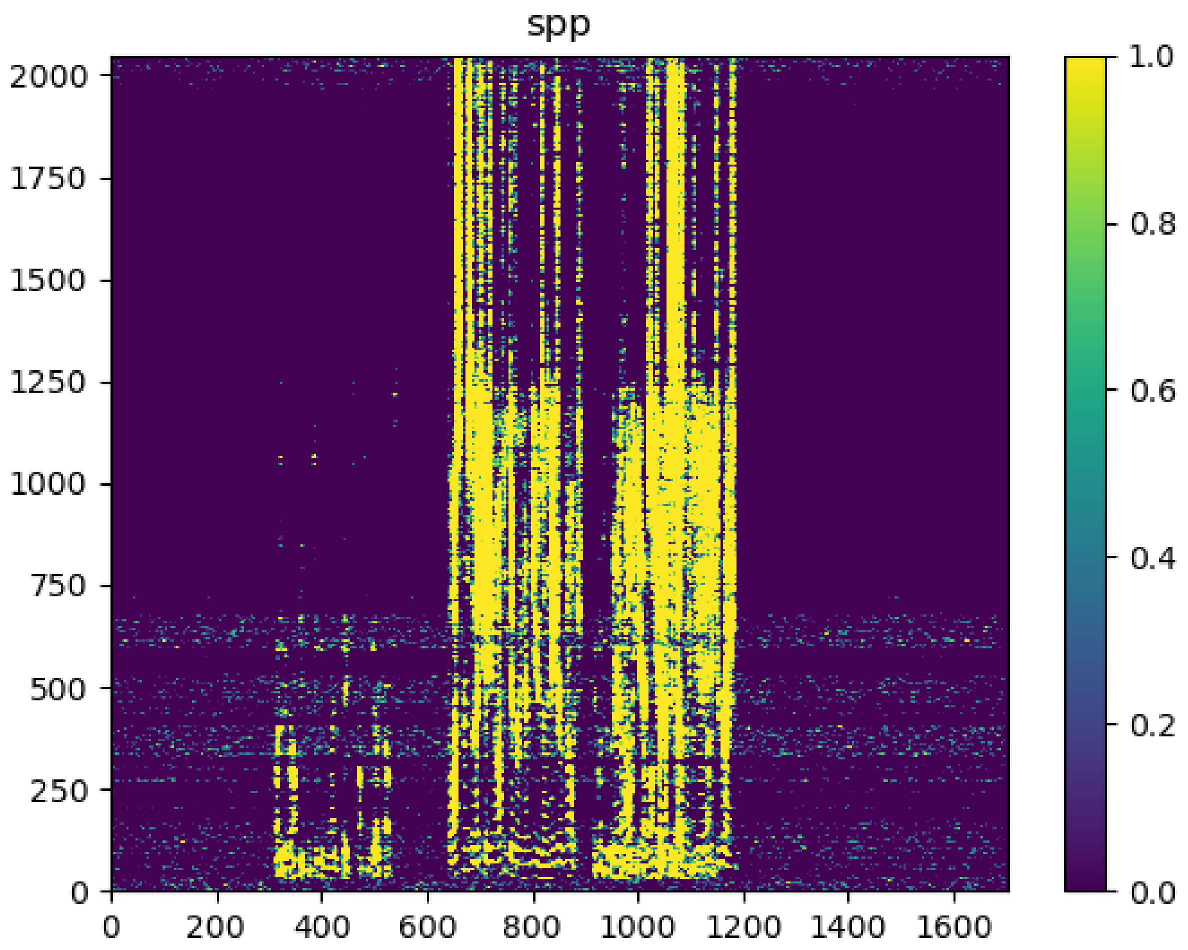
Overview

A priori PSD

Cleaned PSD

SPP

1. Smooth
2. Into dB
3. Define SNR-min, SNR-max
4. $\max(\min(\frac{X - SNR_{min}}{SNR_{max} - SNR_{min}}, 1), 0)$



Overview

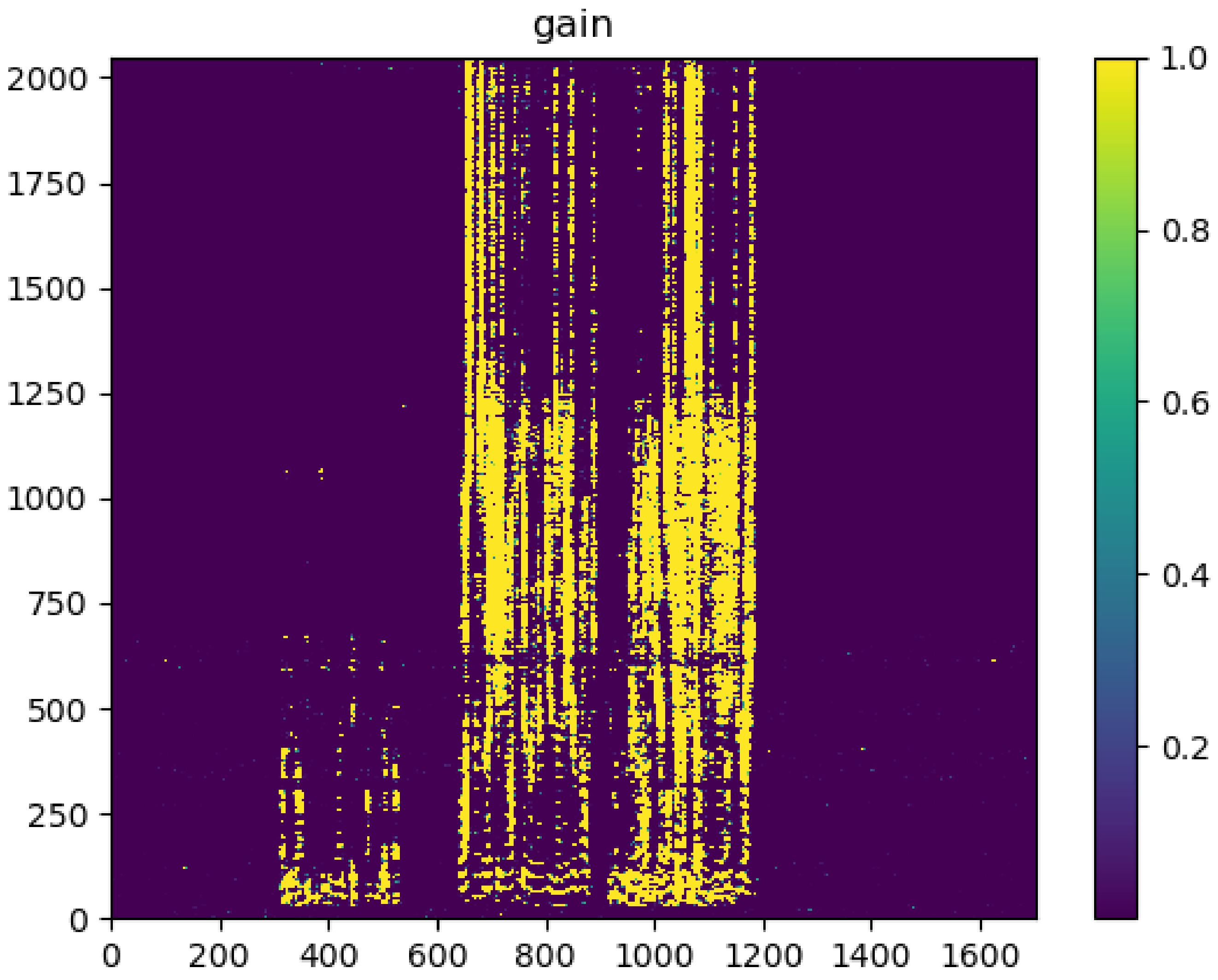
SPP+MASK

Processed SPP

Gain

Gain mask:

$$G_S{}^{\wedge Spp} \ * \ G_N{}^{\wedge(1-Spp)}$$

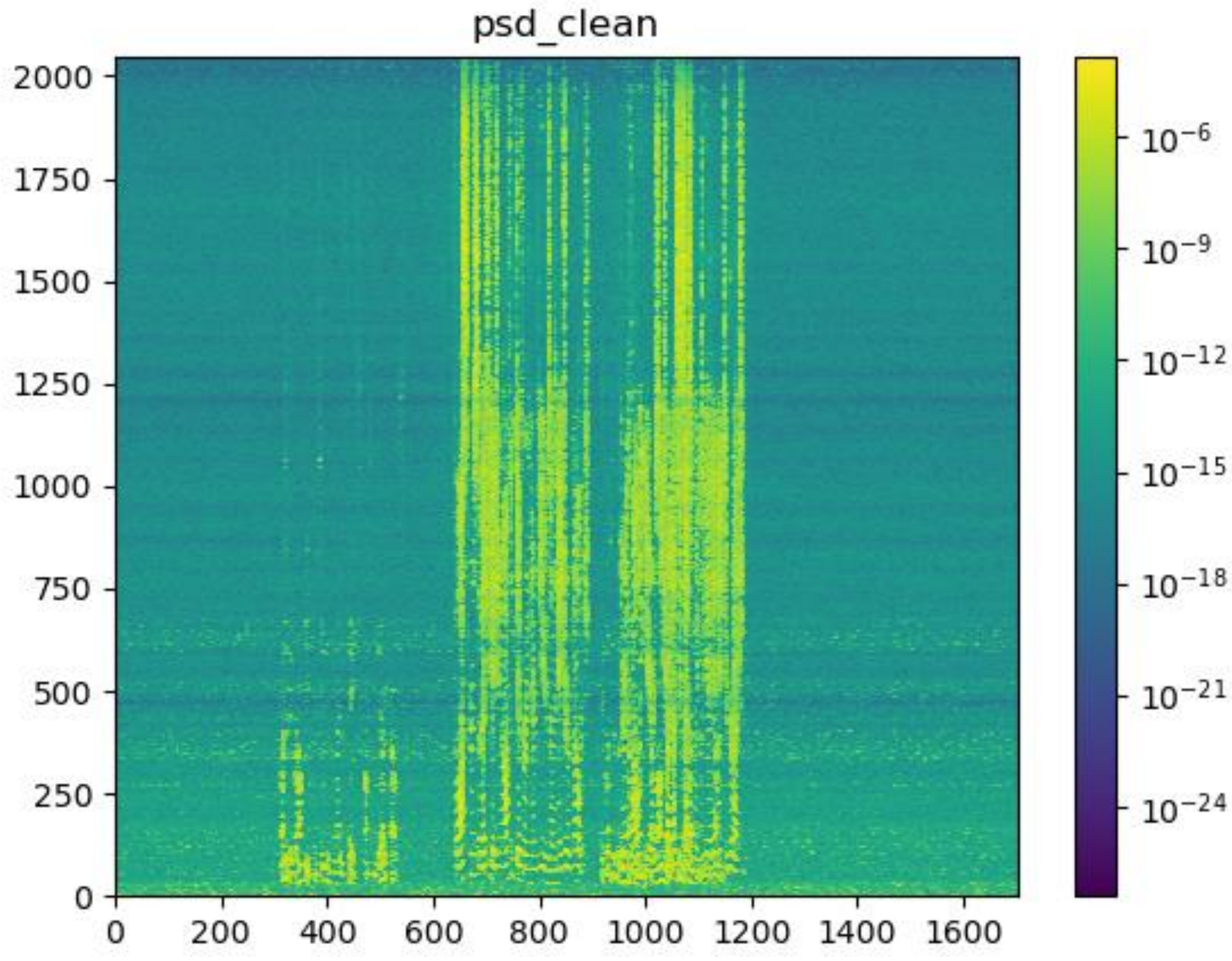


Overview

Cleaned PSD

Cleaned-PSD:

$$Gain * Input\ PSD$$



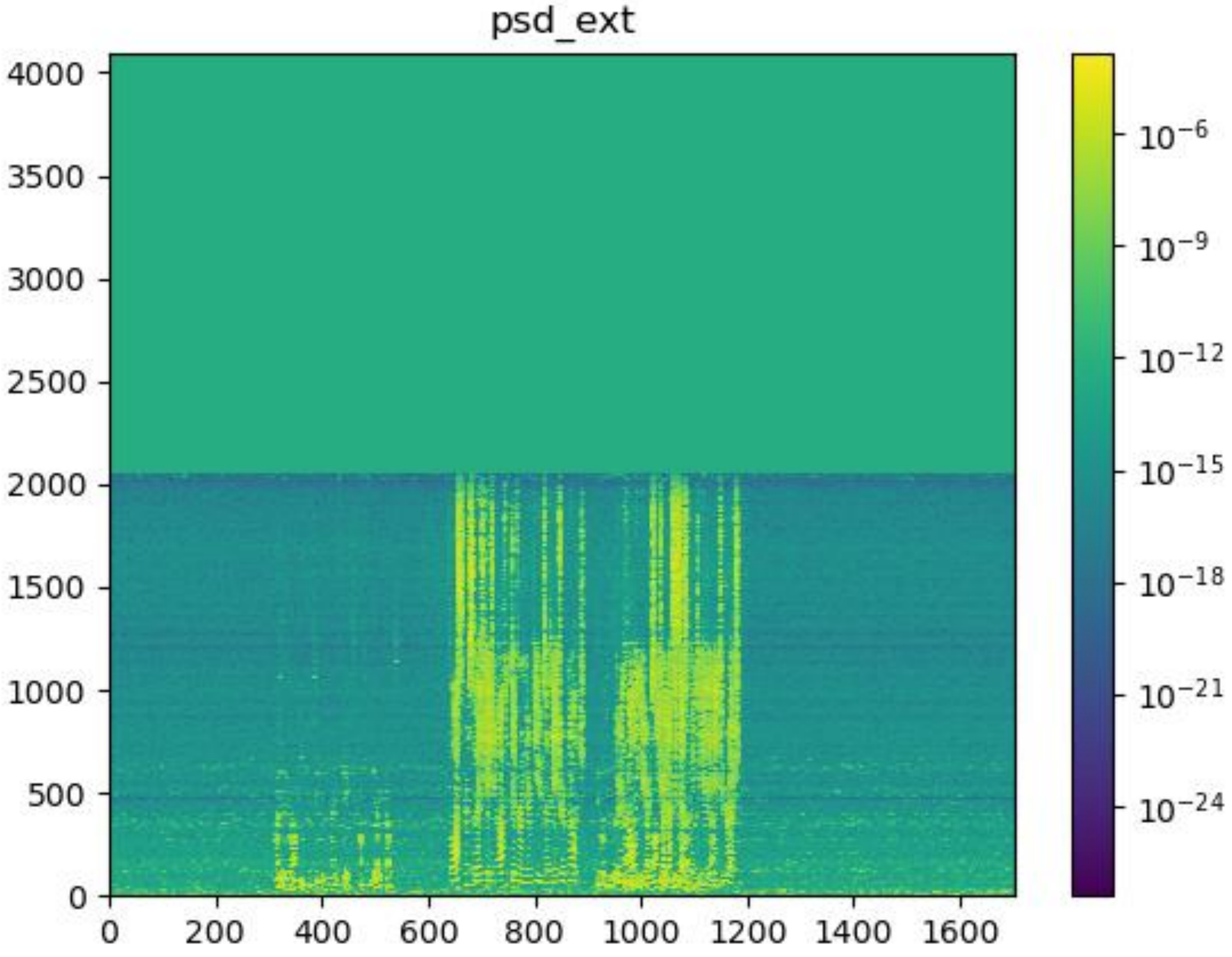
Overview

[A posterior SNR PSD]

PSD extended

Extend the PSD for better resolution.

Filled with noise floor

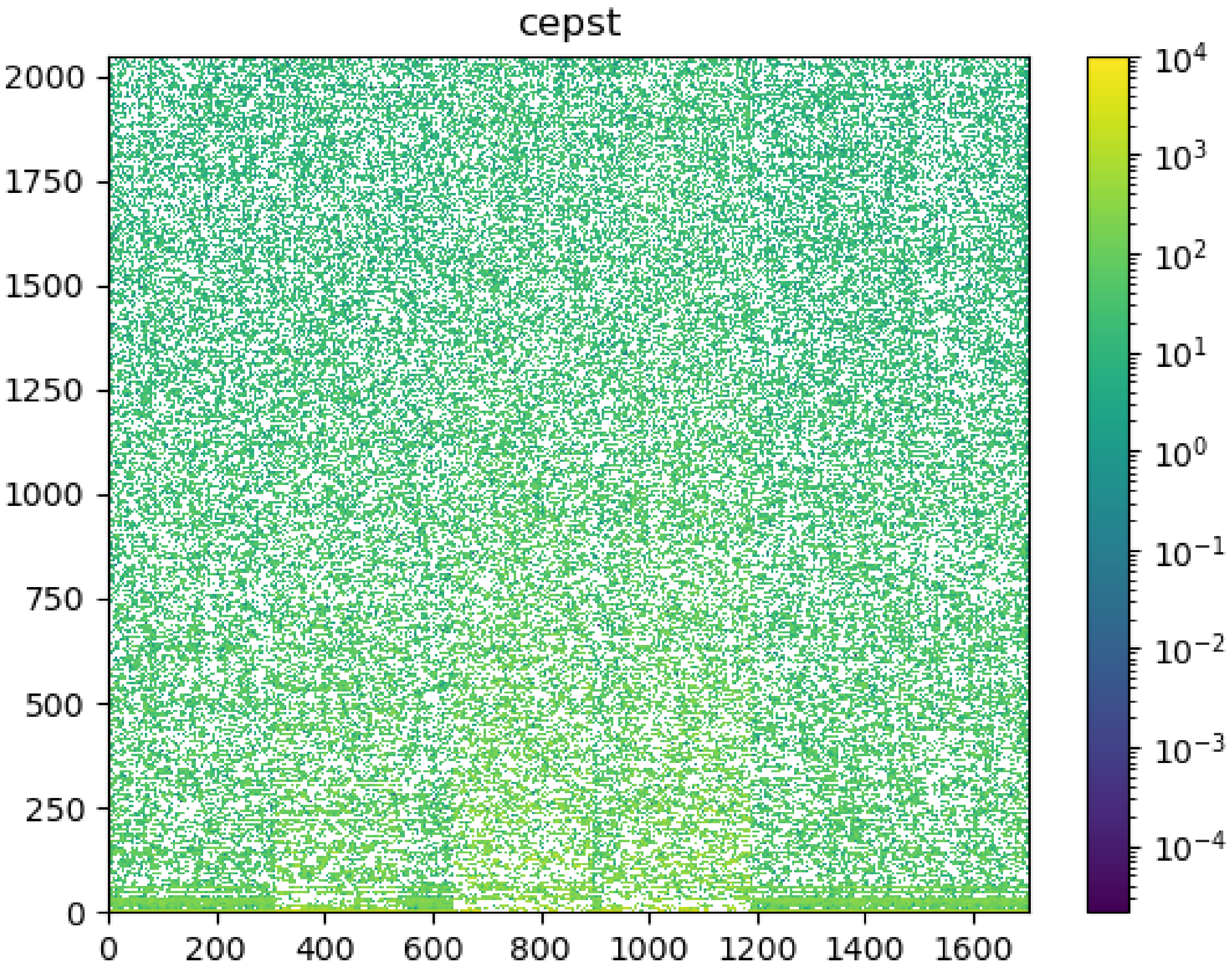


Overview

Cepstrum

Sometimes it is also defined as:[2]

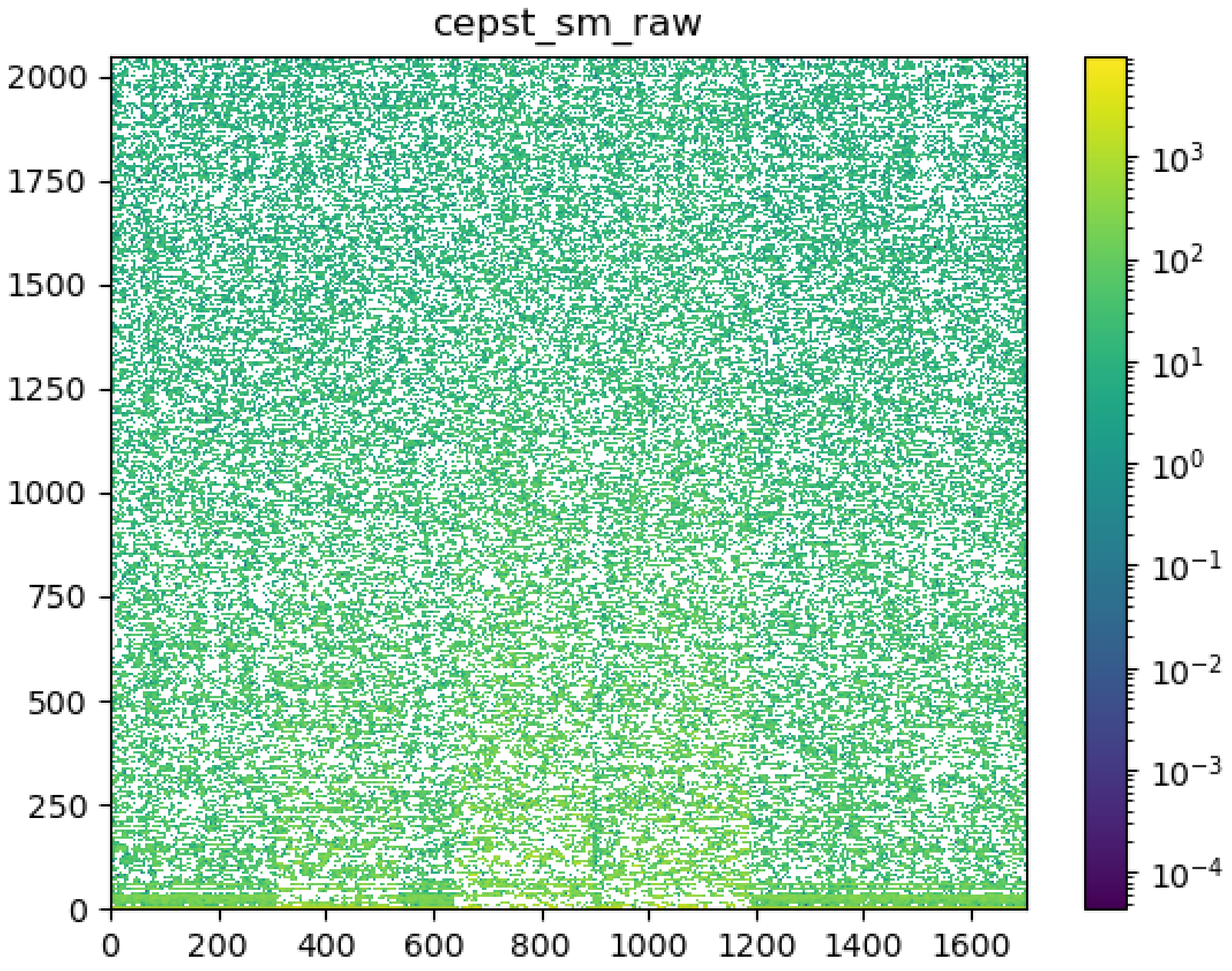
$$C_p = \left|\mathcal{F}\left\{\log\left(|\mathcal{F}\{f(t)\}|^2\right)\right\}\right|^2$$



Overview

https://en.wikipedia.org/wiki/Cepstrum

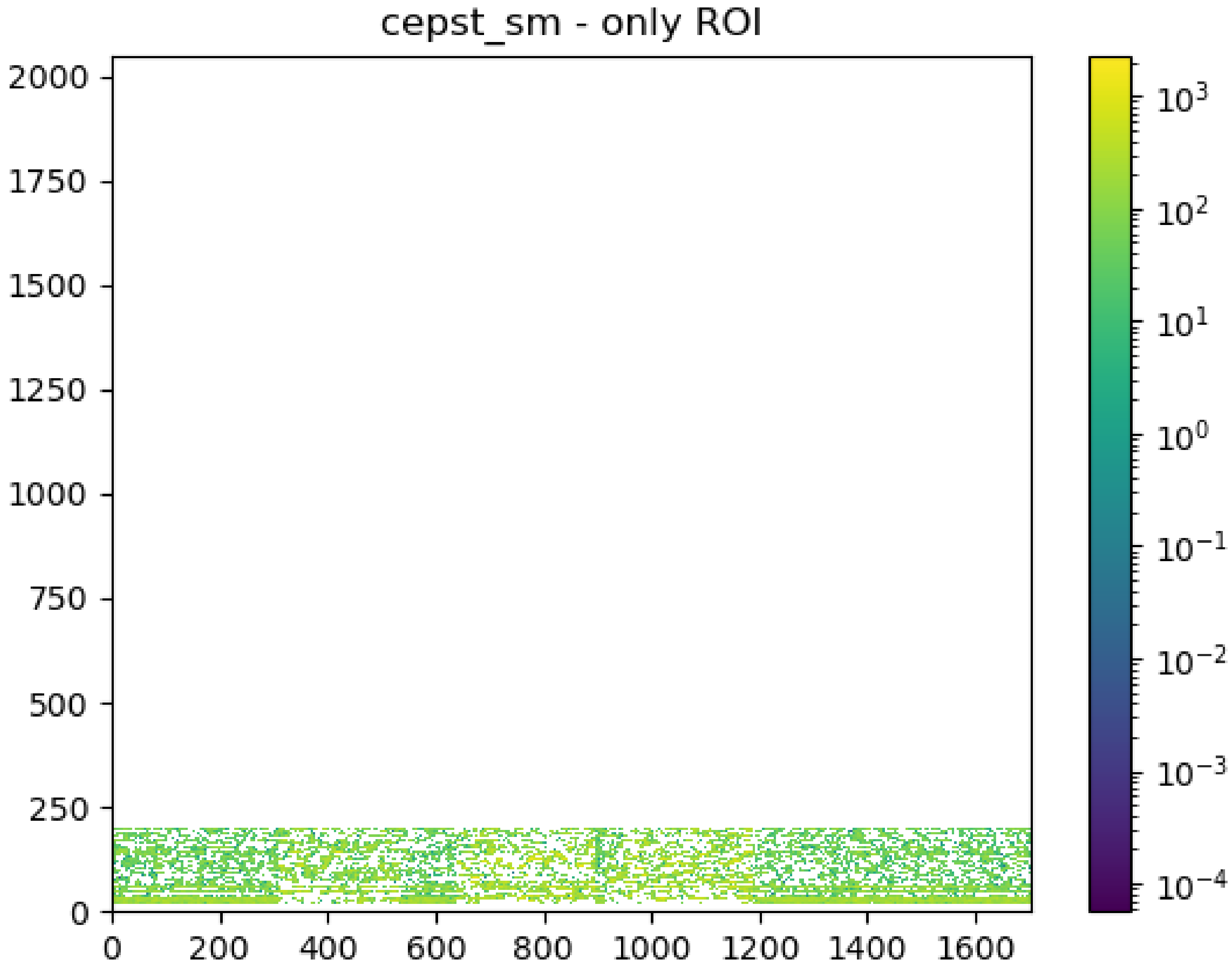
Smoothed Cepstrum



Overview

Cepstrum ROI

We limit the search to only few low frequency bins.

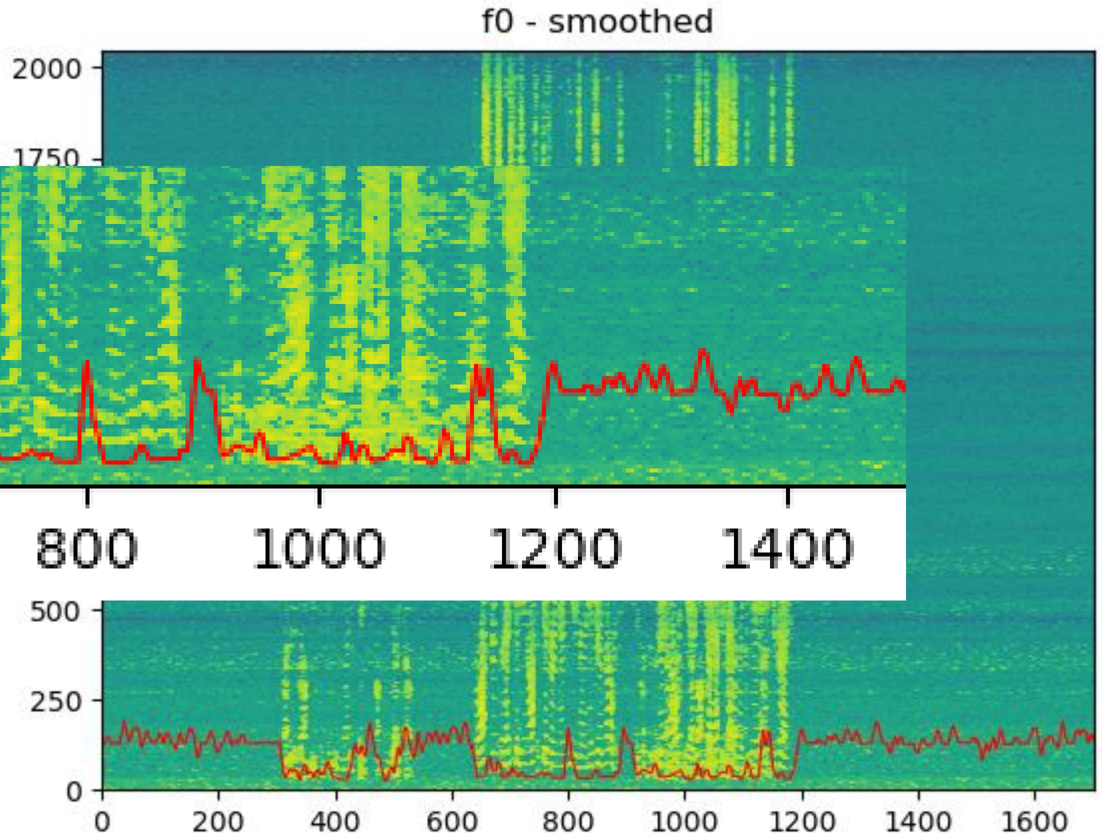


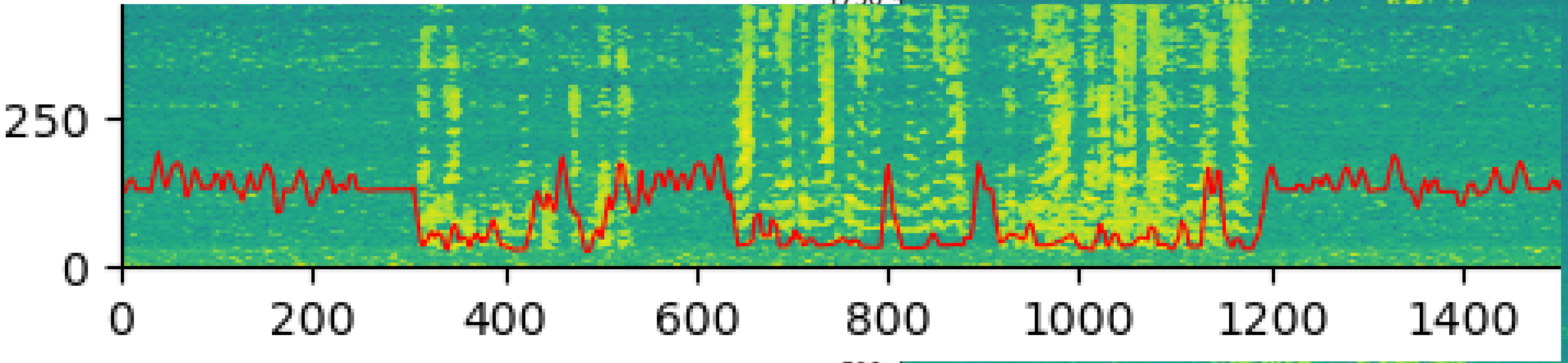
Overview

Smoothed f_0

Overview

Good detection of f_0 (for voiced parts)





The Cepstrum's independent variable are called '*quefrency*', and has a time scale

$$q_0 = quef\left[argmax\left(\mathcal{R}eal(ceptrum)\right)\right]$$
$$f_0 = 1/q_0$$

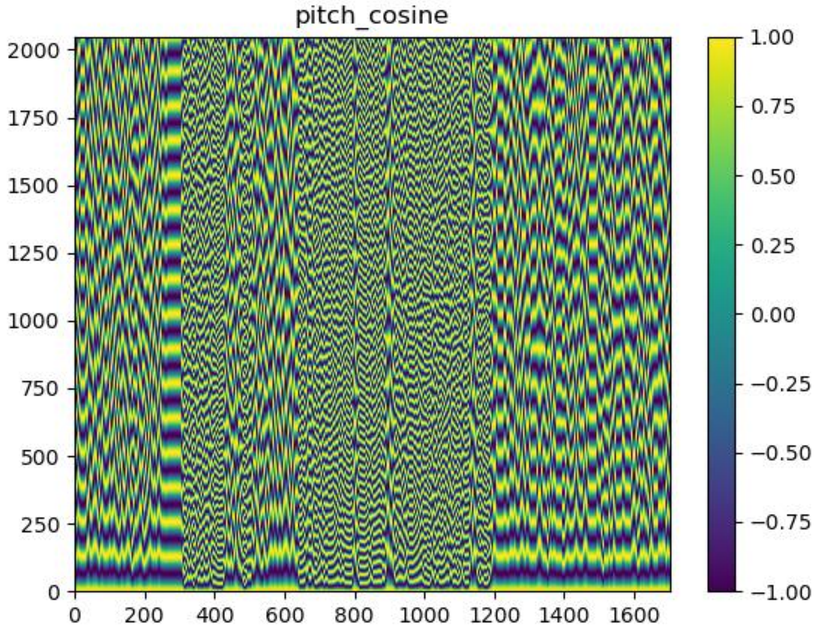
Harmonies Signal

Overview

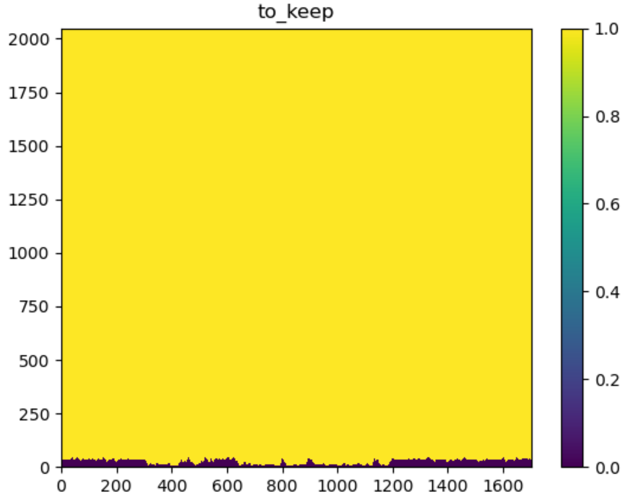
Create a signal built from the harmonies of f_0:
For each time bin:

$$\cos\left(2\pi\cdot\frac{f}{f_0}\right), f\in[0, f_s/2]$$

(this is a cosine with period correspond to the detected f_0 in each time-bin)



Harmonies Cleanup: Remove all bellow $f_0/4$



Harmonies masked

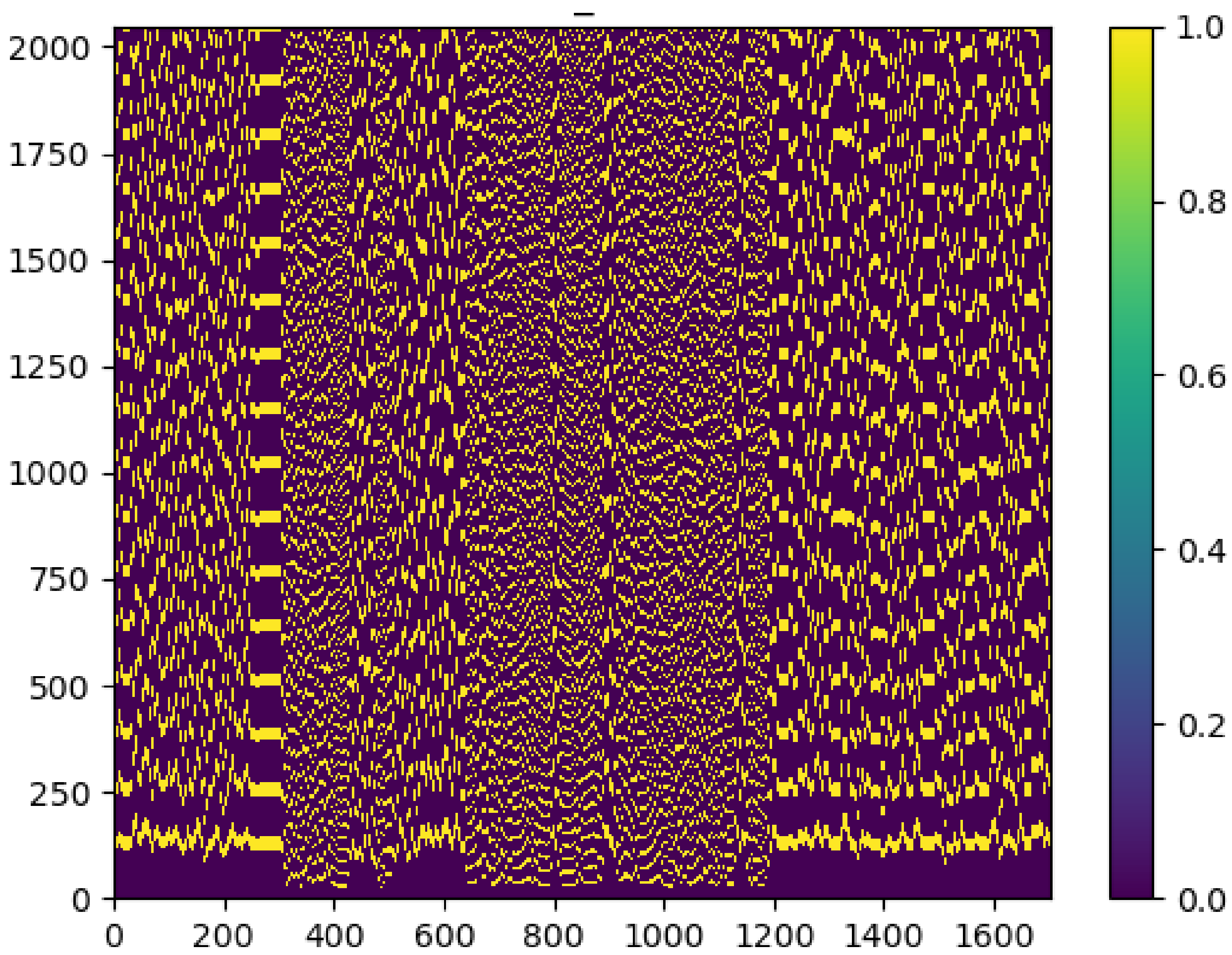
Overview

Define:
1. Contrast
2. Threshold

Apply:

$$exp\left(cont' \cdot (Harm.Sig\ - 1)\right) > \mathrm{Thr}$$

This result in a $[0,1]$ mask, which we multiply by the Harmonies signal.

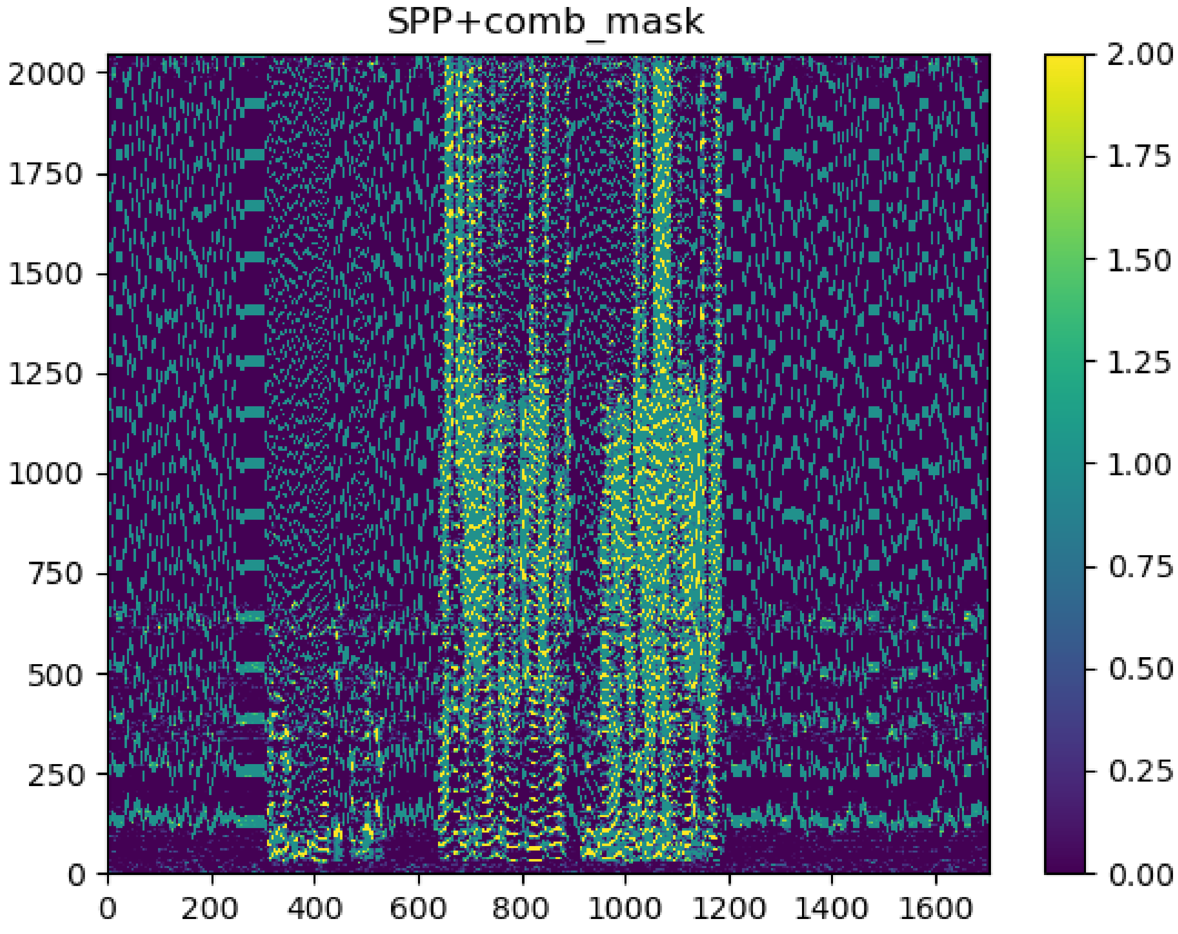


SPP + Harmonies mask

Overview

The combined SSP and Harmonies mask can point out most of the speech signal.

We apply the 'Hysteresis Thresholding' for this signal



First SPP

Processed SPP

Hysteresis Thresholding

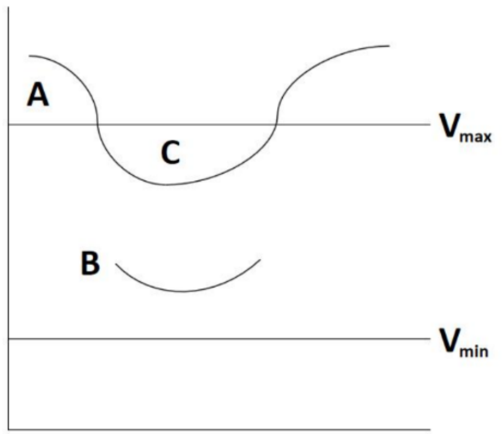
Overview

Hysteresis thresholding, Originally from image processing:

- High threshold: Pixels above this are definitely edges.
- Low threshold: Pixels below this are definitely not edges.
- In-between: Pixels between the thresholds are potential edges.

Hysteresis considers a pixel a true edge only if it's above the low threshold AND connected to a pixel above the high threshold.

This means weak edges only get included if they're part of a stronger edge, reducing noise and creating cleaner results.



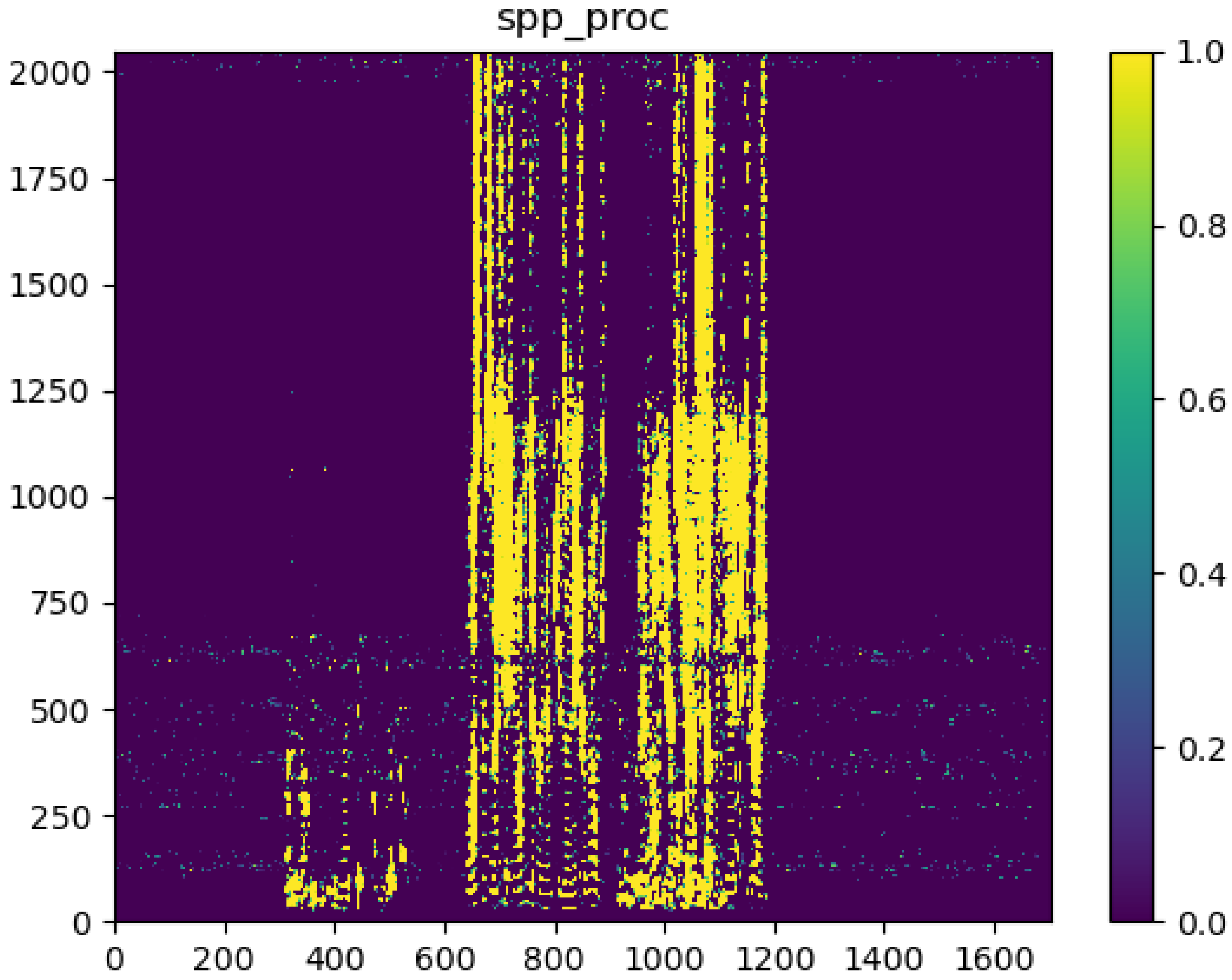
The edge A is above the Vmax, so considered as “sure-edge”. Although edge C is below Vmax, it is connected to edge A, so that also considered as a valid edge and we get that full curve.

But edge B, although it is above minVal and is in the same region as that of edge C, it is not connected to any “sure-edge”, so it is discarded

[p.12: https://www.researchgate.net/publication/338145885_Towards_a_Digital_Diatom_image_processing_and_deep_learning_analysis_of_Bacillaria_paradoxa_dynamic_morphology#pf13](https://www.researchgate.net/publication/338145885_Towards_a_Digital_Diatom_image_processing_and_deep_learning_analysis_of_Bacillaria_paradoxa_dynamic_morphology#pf13)

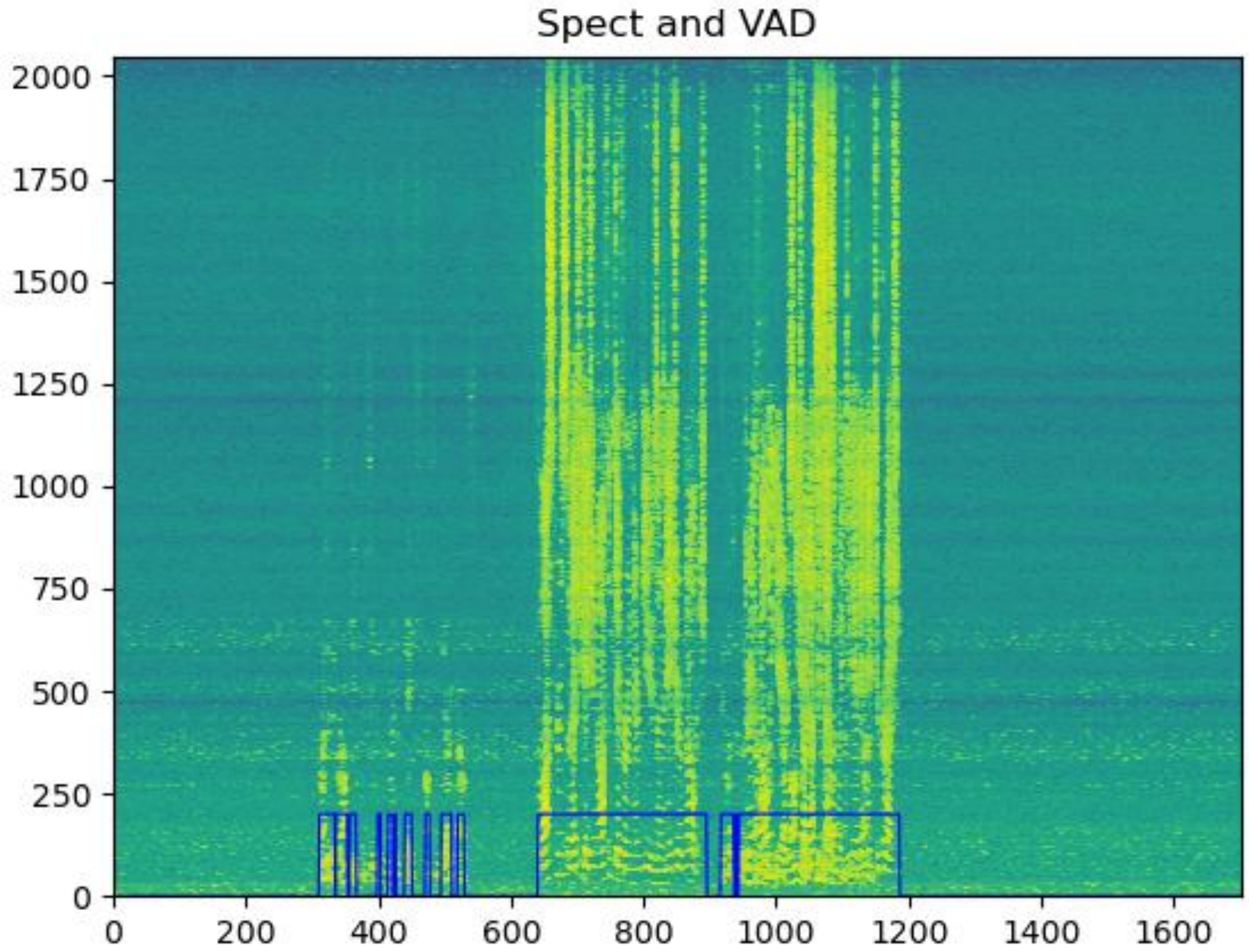
Processed SPP

1. There is less noise
2. Clearer speech harmonies
3. Extended bands



VAD

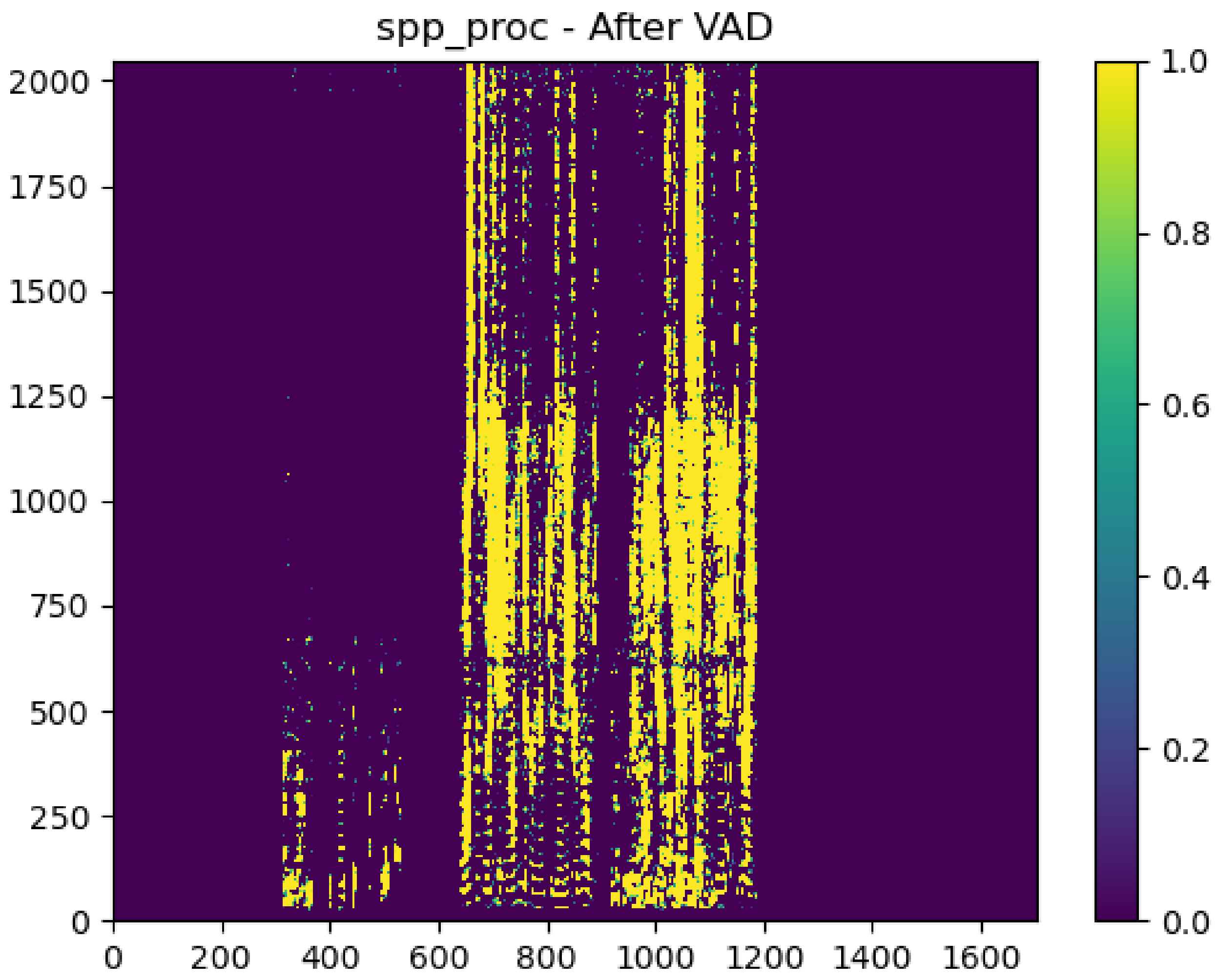
1. Set threshold: 0.015-0.03
2. VAD is given by:
   $mean(SPP) > Thr$



Overview

SPP after VAD

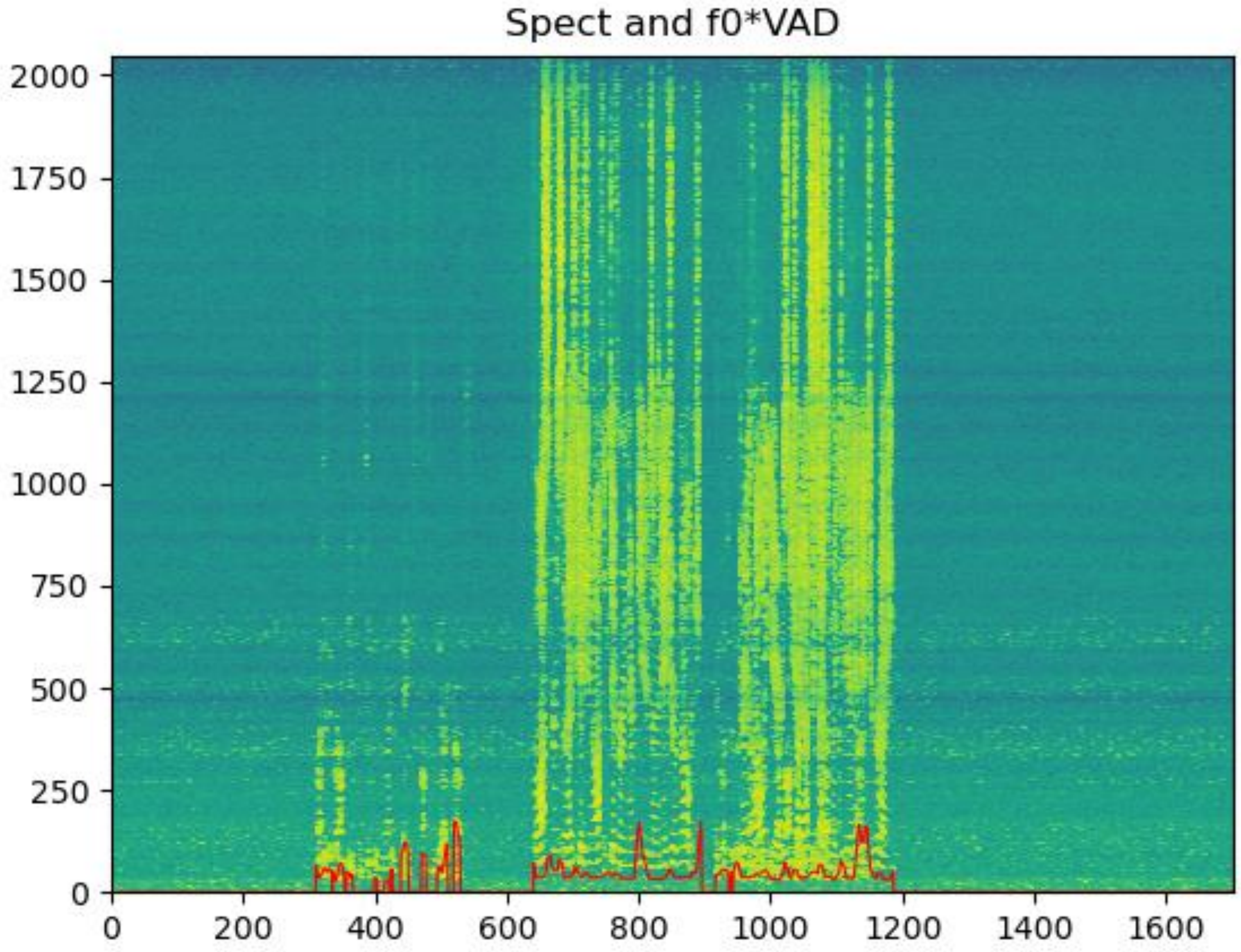
Overview



See [next slide](#) -
for Spectrogram with f_0 **and VAD**

Processed SPP

Spectrogram, f_0 and VAD



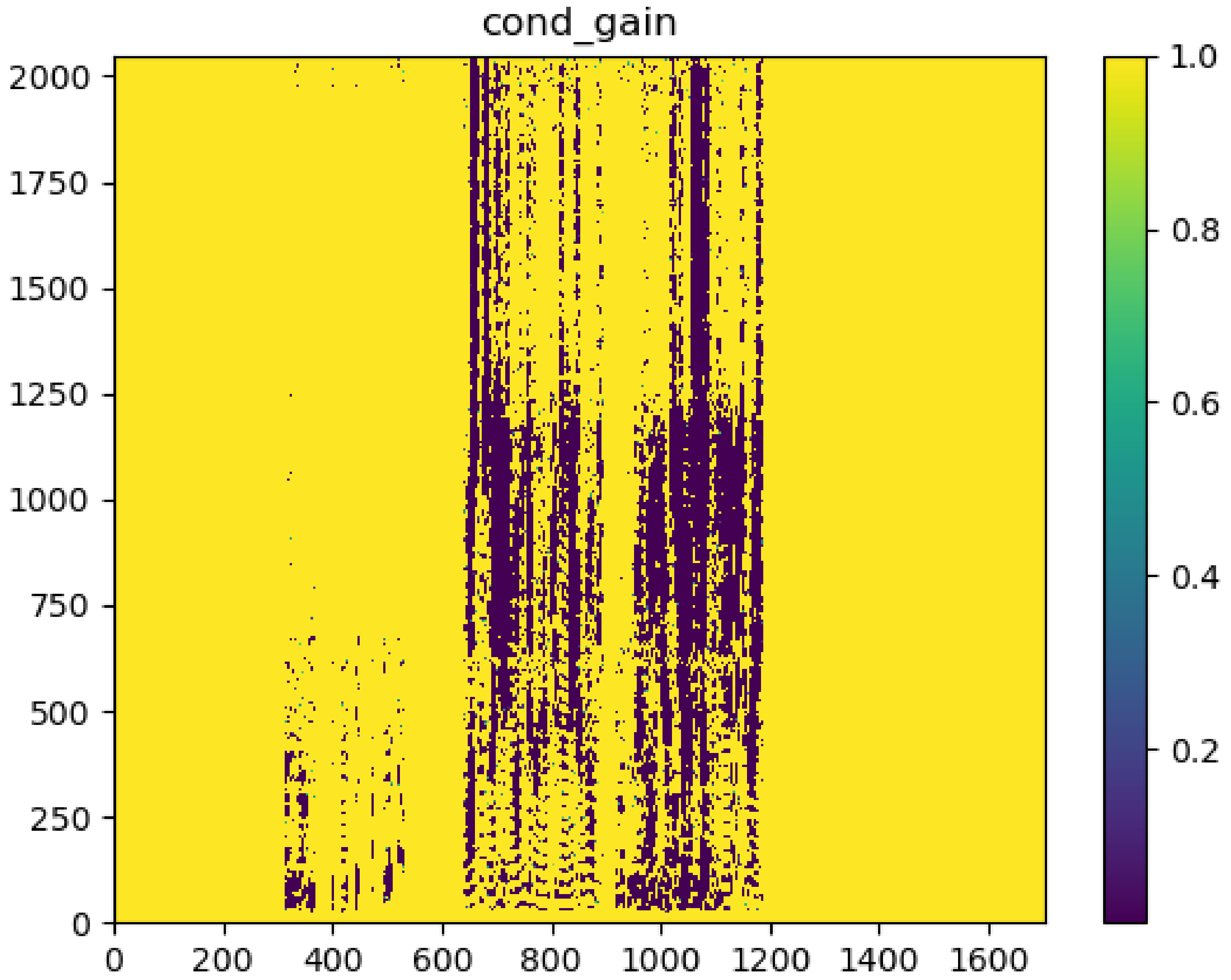
Overview

f_0

Gain

Gain mask:

$$G_S{}^{\wedge Spp} * G_N{}^{\wedge(1-Spp)}$$

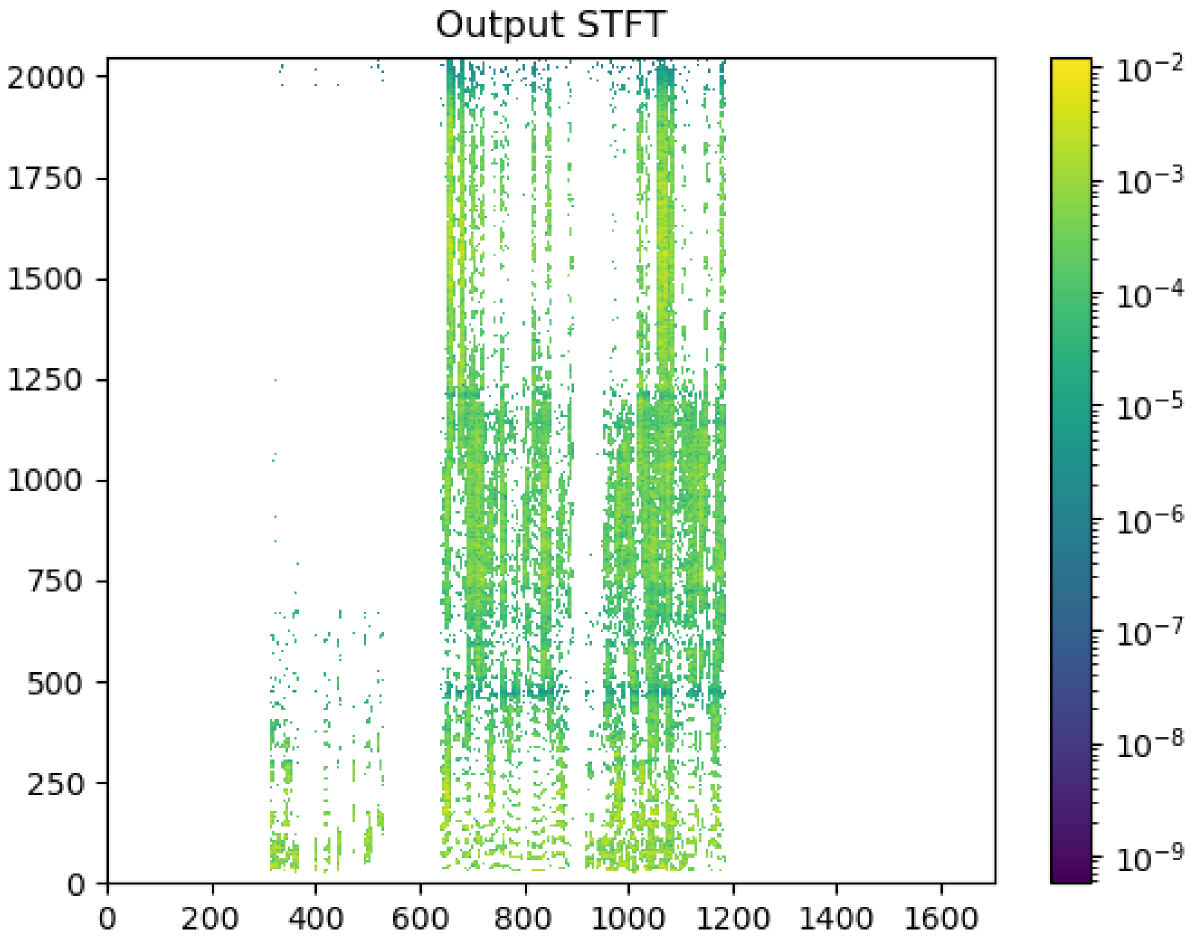


Overview

Output - Cleaned Spectrogram

Out STFT=

$$input\ STFT * \sqrt{Gain}$$



Overview

Or, Noise extraction - [Next Slide]

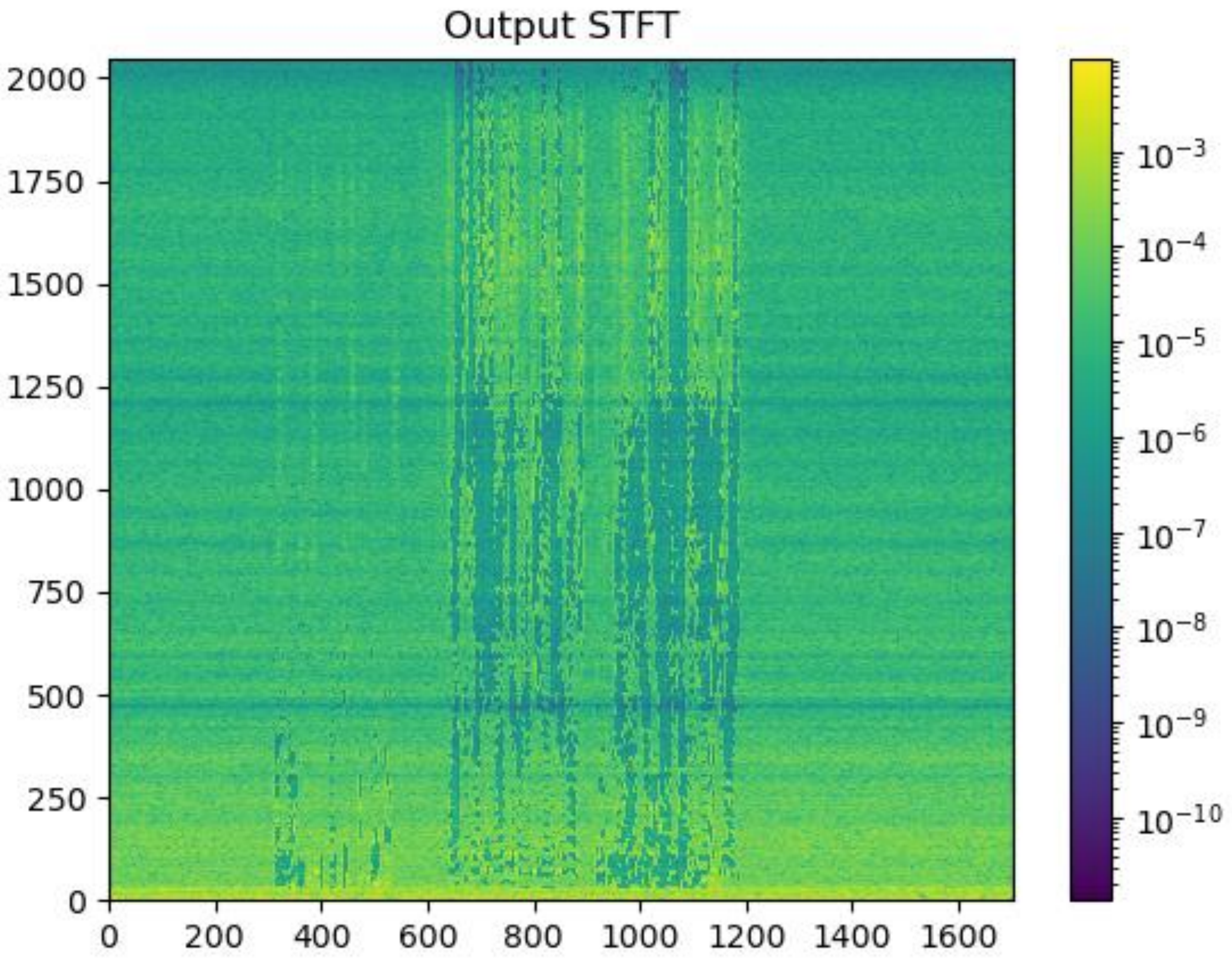
Output – Noise Extraction

Out STFT=

$$input\ STFT * \sqrt{1 - Gain}$$

Or

$$input\ STFT / \sqrt{Gain}$$



Overview

Demos

Now listen to 3 demos