Ahmad Hasanain
Dr Veton Kepuska

Speech Quefrency Transform (SQT)

A Technical Presentation

Common Questions

What are quefrencies and cepstrograms?
The quefrencies are frequencies of frequencies, and the cepstrograms are spectrograms of spectrograms.

Does the sampling rate affect the frequency resolution?
No, the length of the window affects the resolution of the frequency.

What is SQT?
SQT stands for Speech Quefrency Transform, and it is for speech features extraction, the spectrogram and the pitch, in particular.

What to expect.

How to apply the SQT procedure?

How to generate the reciprocal scale in Python?

How to recover the speech waveform from the extracted speech features?

What components are in the SQT Matrix?

What can be detected from the pitch patterns?

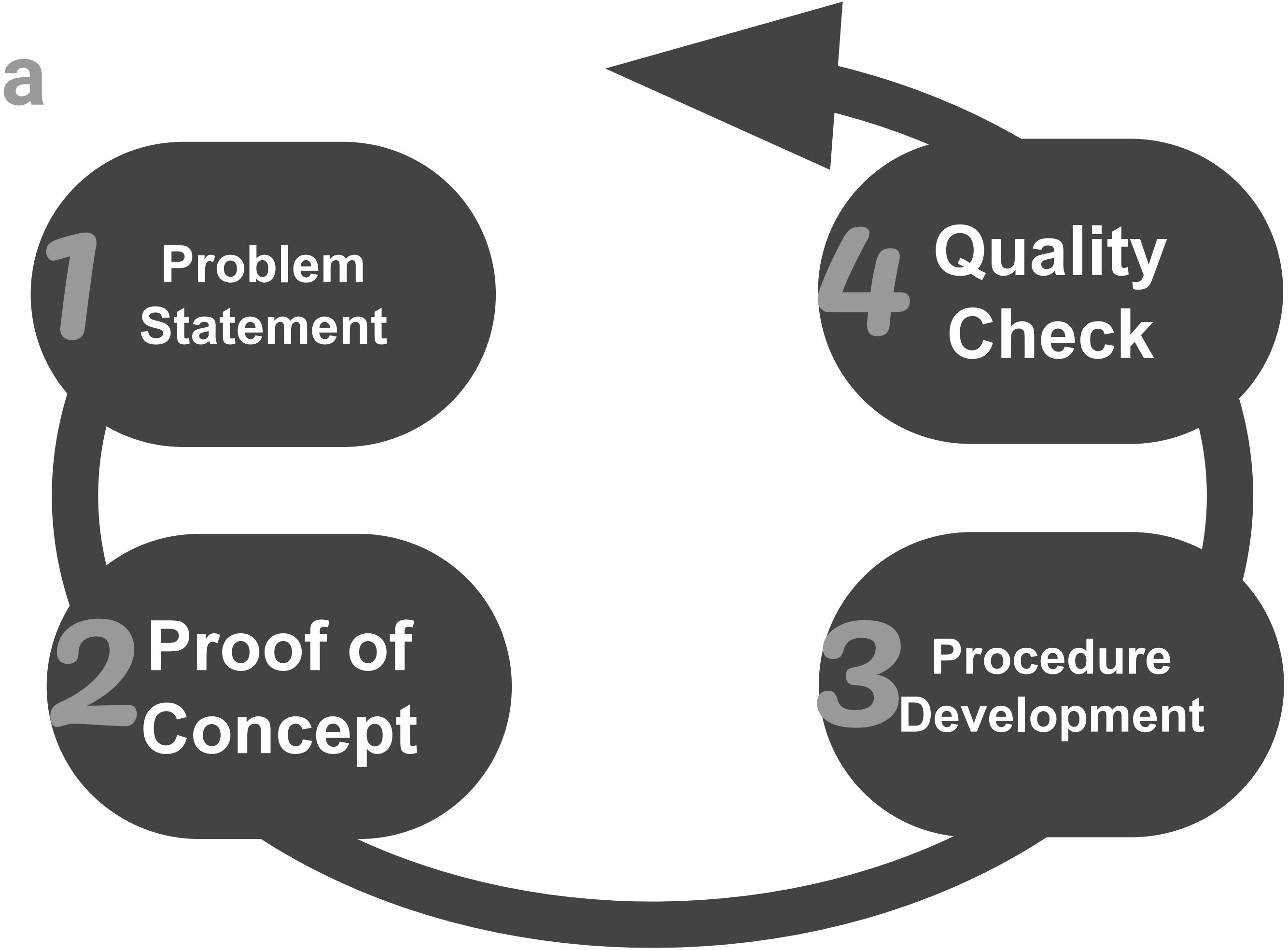
What is GPE?

Does SQT filter out wideband noise?

Does SQT filter out the over- and undertones?

Does SQT normalize spectrograms?

Agenda



1 Problem Statement

2 Proof of Concept

3 Procedure Development

4 Quality Check

The Pain Points of the Competitive Solutions

Spectrogram

Suffers from curse of dimensionality and so has to be followed by complicated learners.

Needs High Storage.

Has a Complex Speech Recovery.

MFCC

Lacks PItch Feature.

Generates Unrecoverable Features.

Doesn’t Normalize Speech Features.

Other Pitch Trackers

Lacks Speech Features.

Generates Incomplete, Unrecoverable Speech Features.

Some algorithms have to be combined, and their results sometimes were corrupted by smoothing.

SQT can DO

Streaming the Speech Features

Implementing a JavaScript Web Client

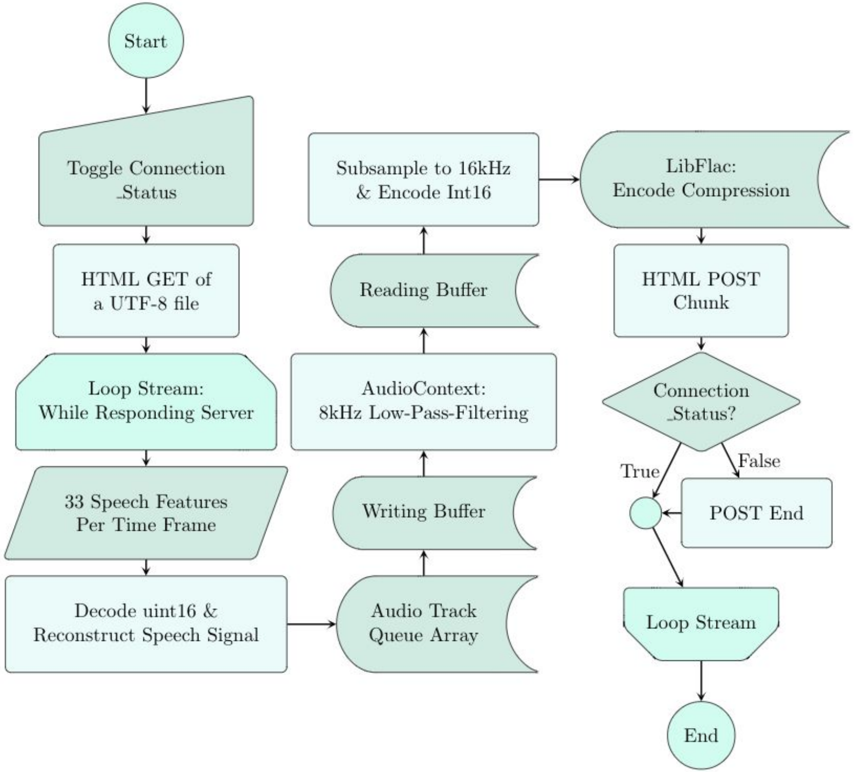

Start
Toggle Connection _Status
HTML GET of a UTF-8 file
Loop Stream: While Responding Server
33 Speech Features Per Time Frame
Decode uint16 & Reconstruct Speech Signal
Audio Track Queue Array
Writing Buffer
AudioContext: 8kHz Low-Pass-Filtering
Reading Buffer
Subsample to 16kHz & Encode Int16
LibFlac: Encode Compression
HTML POST Chunk
Connection _Status?
True
False
POST End
Loop Stream
End

Figure 6: JavaScript Web Client Flowchart

Without data compression, the features of the “wideband speech signal” requires only 0.02Mbps transmission bandwidth.

Live Demo

GitHub

Approach Introduction

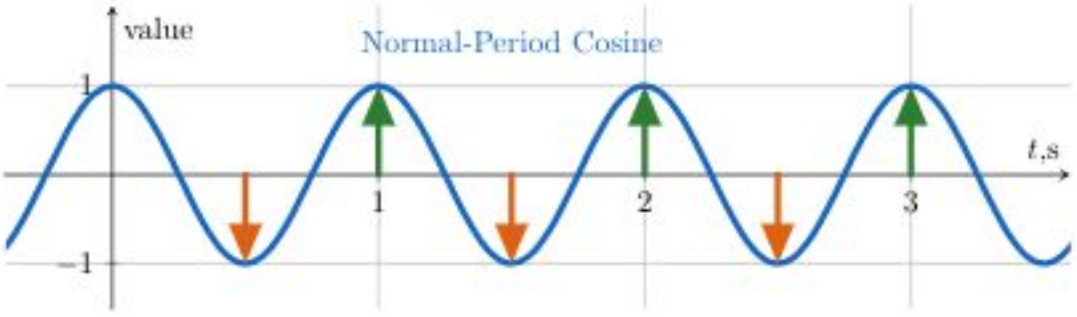


Figure 1: Frequency Detection: Fourier definition on time domain

Modeling the Frequency of Frequencies Requires Applying the Cosine Curve on the *f*-axis.

How Can We Model Quefrency?

Time-bounded signals can be approximated by sums of polynomials (Stone-Weierstrass Theorem).

Applying the Cosine Curve on the *f*-axis Can Be Obtained by a Mixture of Normal Distributions.

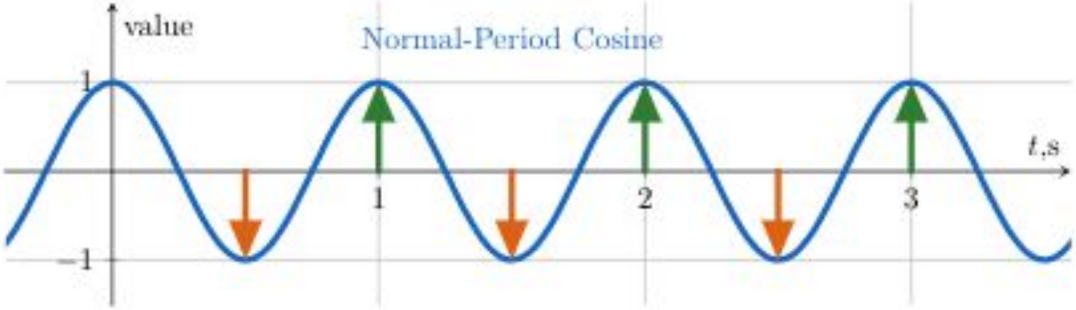


Figure 1: Frequency Detection: Fourier definition on time domain

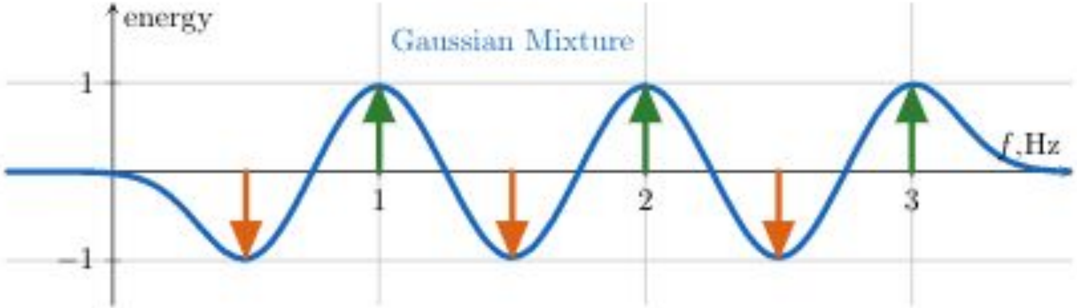


Figure 2: Quefrency Detection: Approximated result on frequency domain

Summation of Two Trains of Normal Gaussians with the Correct Variance Converges to Cosine.

$$cos(2\pi \cdot T \cdot x) \approx \sum_{m=1}^{M} e^{-(x-m\cdot T)^2/\sigma^2} - e^{-(x-(m-0.5)\cdot T)^2/\sigma^2}$$

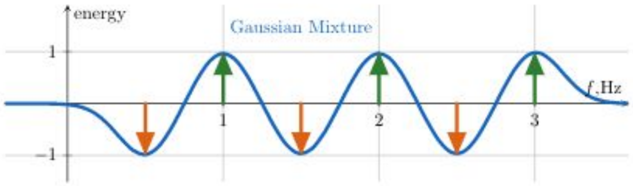


Figure 2: Quefrency Detection: Approximated result on frequency domain

The Finite Approximation Is Found Numerically

$$\text{Gaussian\_Window}(x) = e^{-x^2/\sigma^2}$$

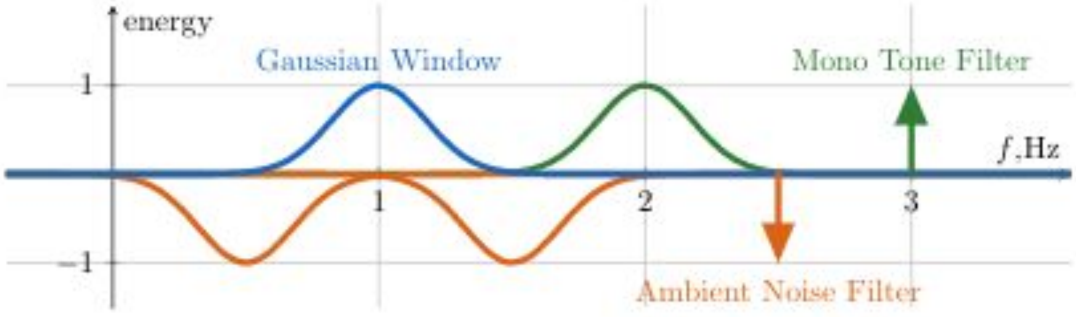
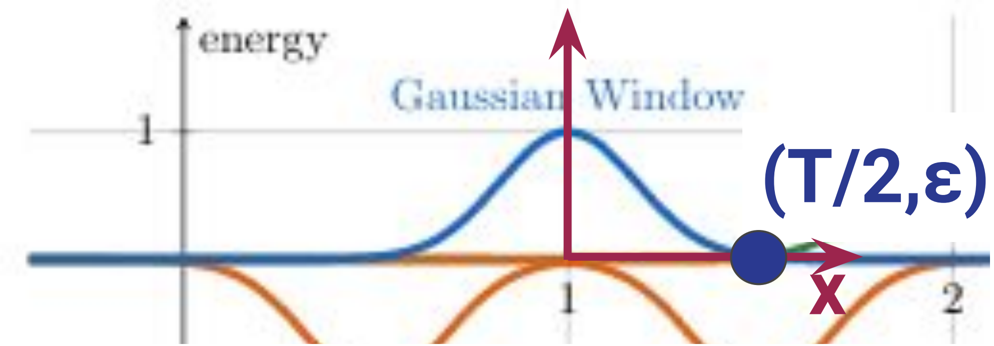


Figure 3: Quefrency Filter Banks: Components of the Convolution

Convolving two frequency signals is equivalent to multiplying their time versions (Fourier Transform Properties).

The function is realizable by a convolution between a Gaussian window and impulse trains.



At x= T/2, the value of the Gaussian_Window component should be very small, (ie, ε = 0.0001), then:

$$\sigma^2 = \frac{T^2}{-4 \cdot log_e(10^{-4})}$$

Expand the Components to Find the Correct Variance. Note the Cosine Has an Interval of T.

SQT IS LEGIT

Applying SQT

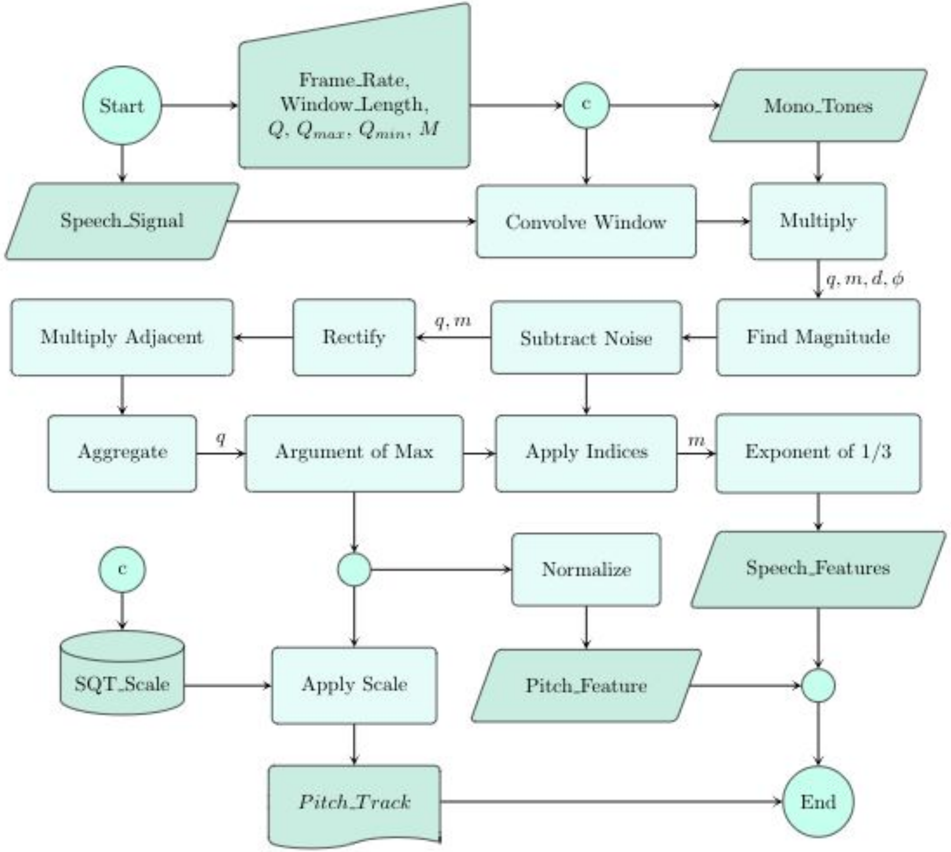
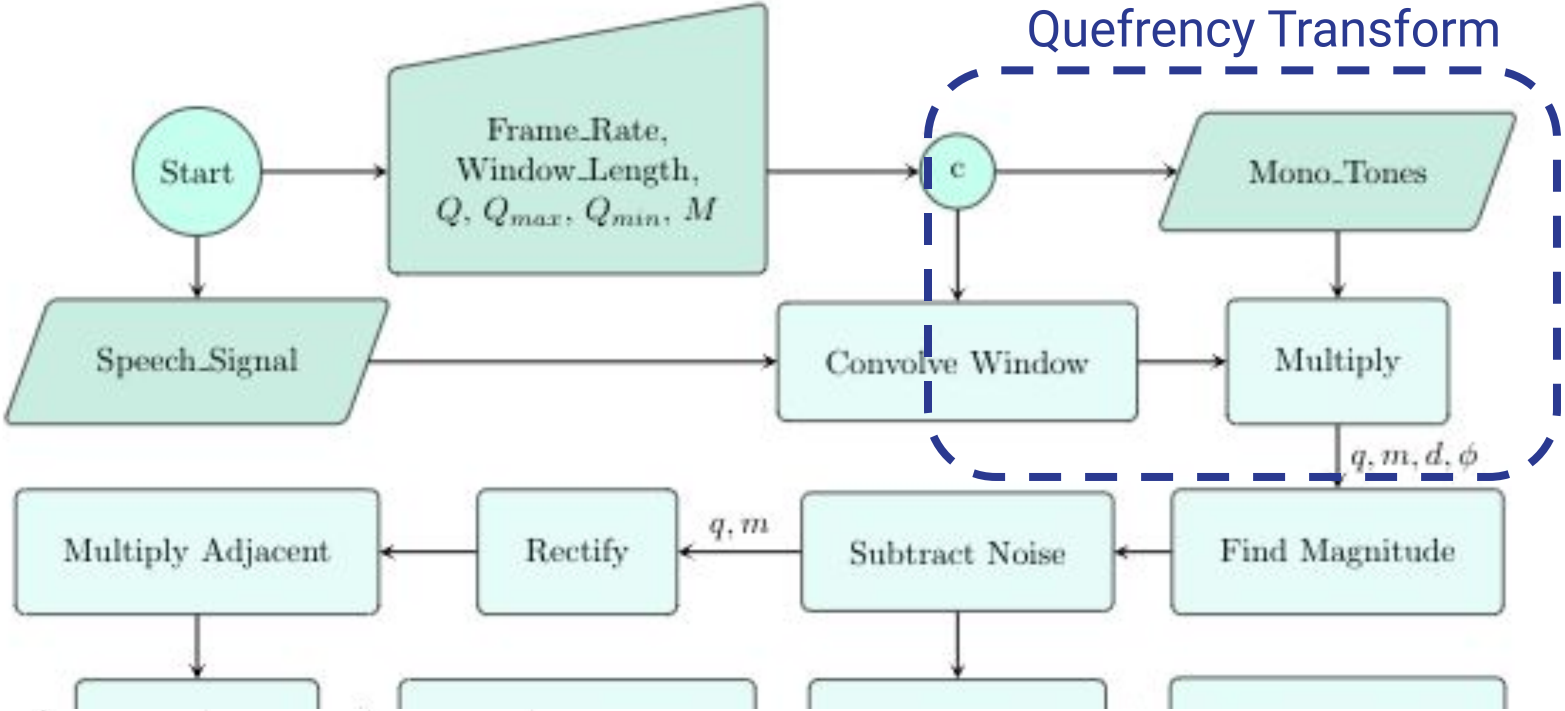


Figure 4: Procedure of Features' Extractions

The procedure of achieving the desired function and obtaining the desired features.



Quefrency Transform

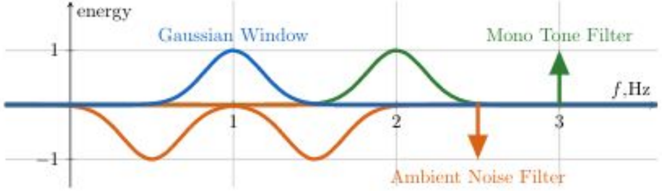
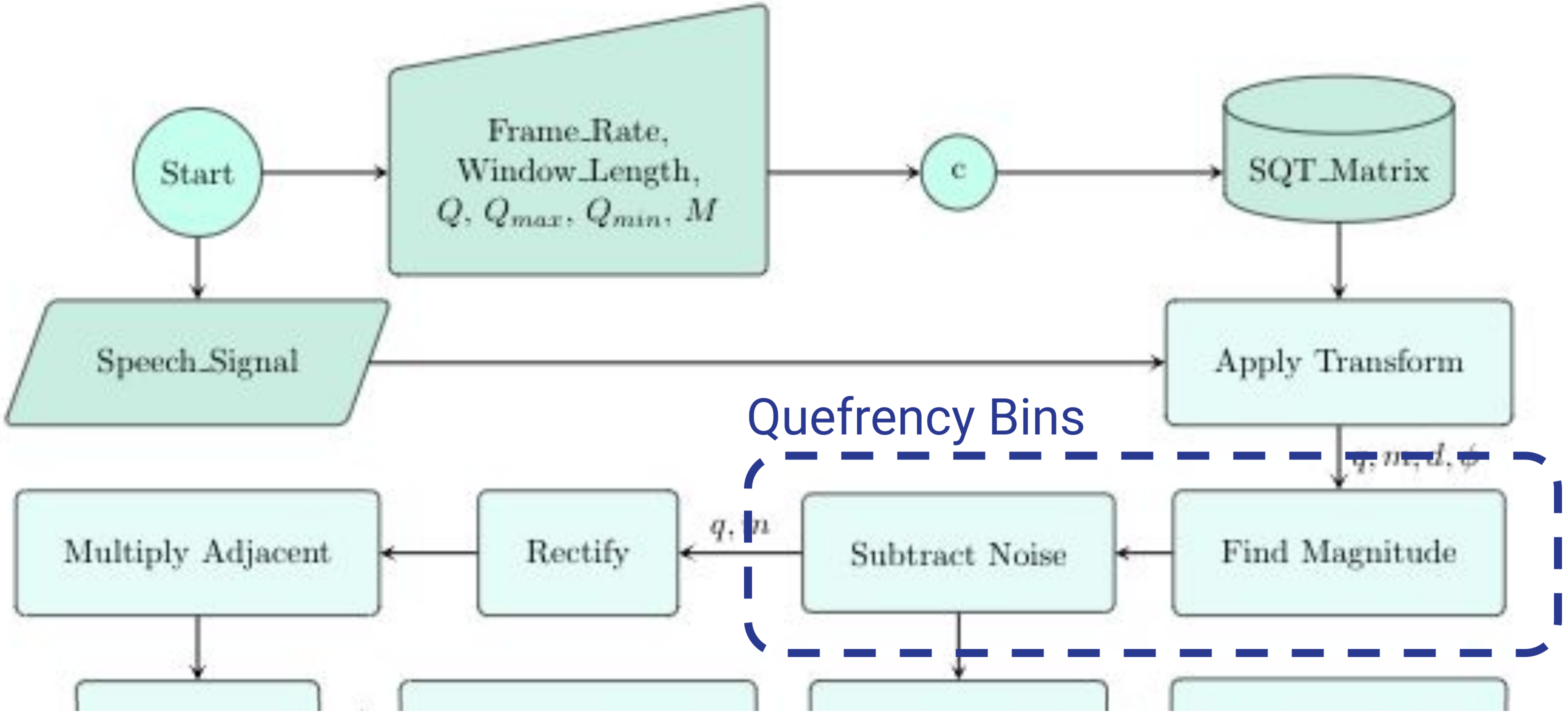
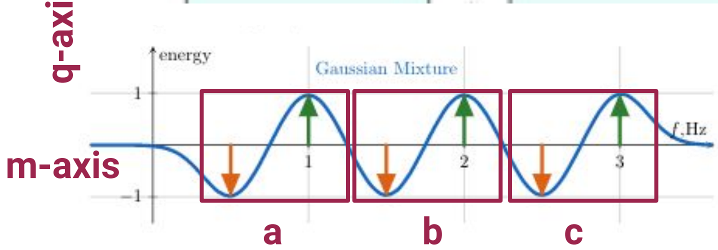


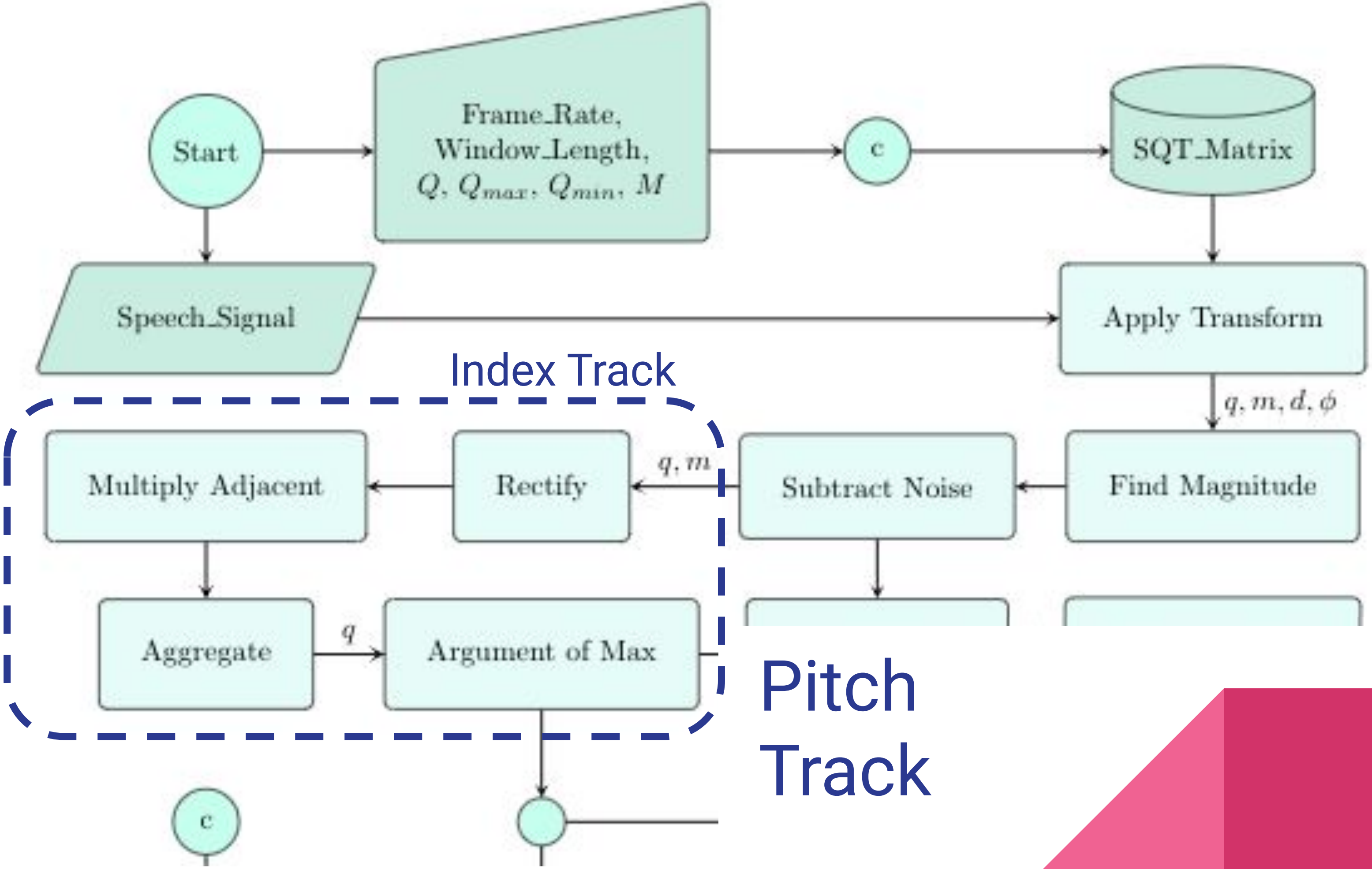
Figure 3: Quefrency Filter Banks: Components of the Convolution

Generating the Components





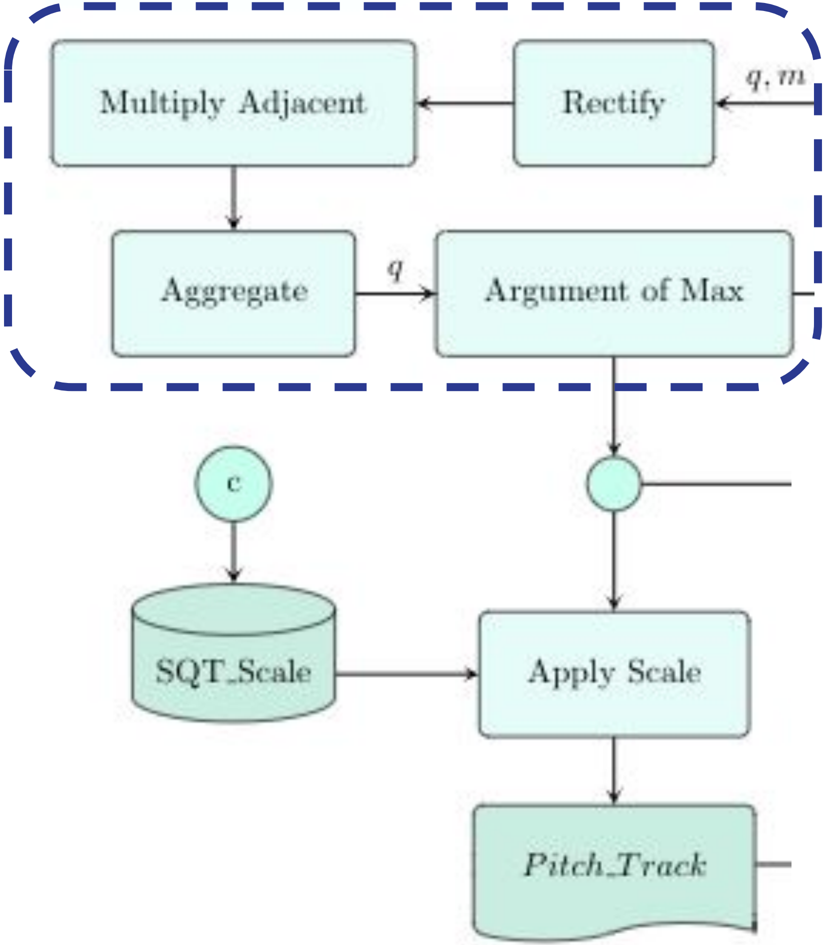
Energies of the Period Wavelets



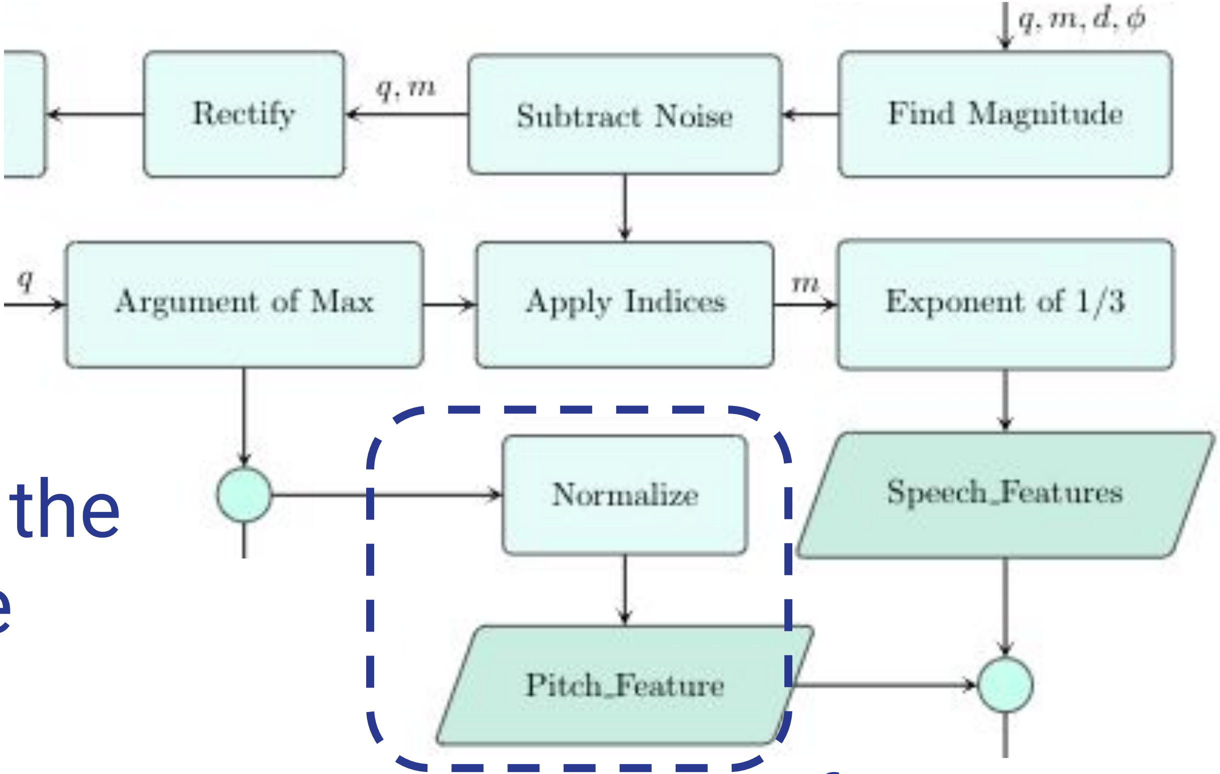
Index Track

Pitch Track

Index Track

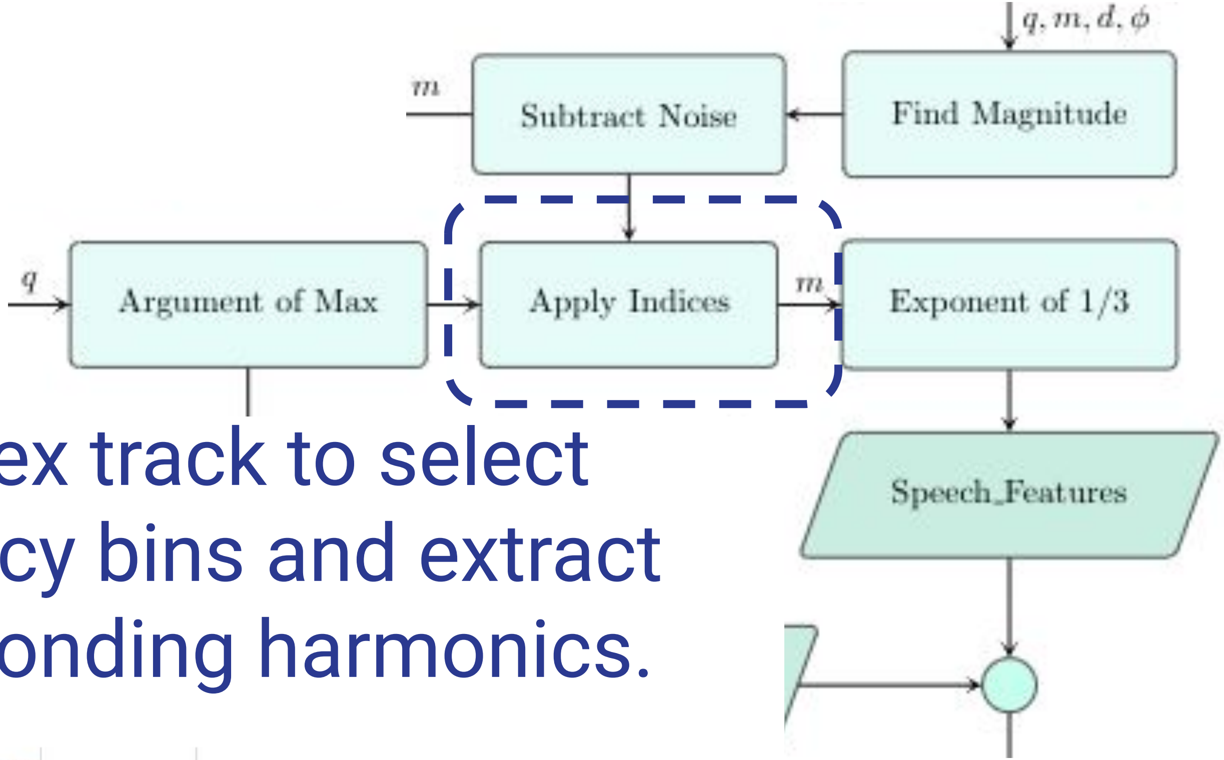


Pitch Track

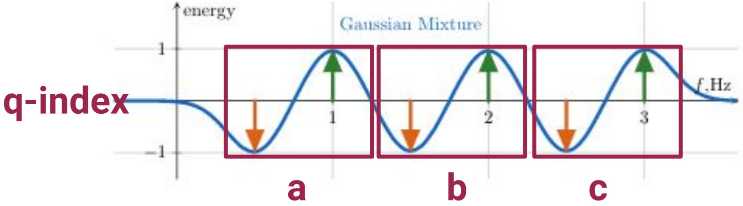


Divide the indices by the SQT_Scale length.

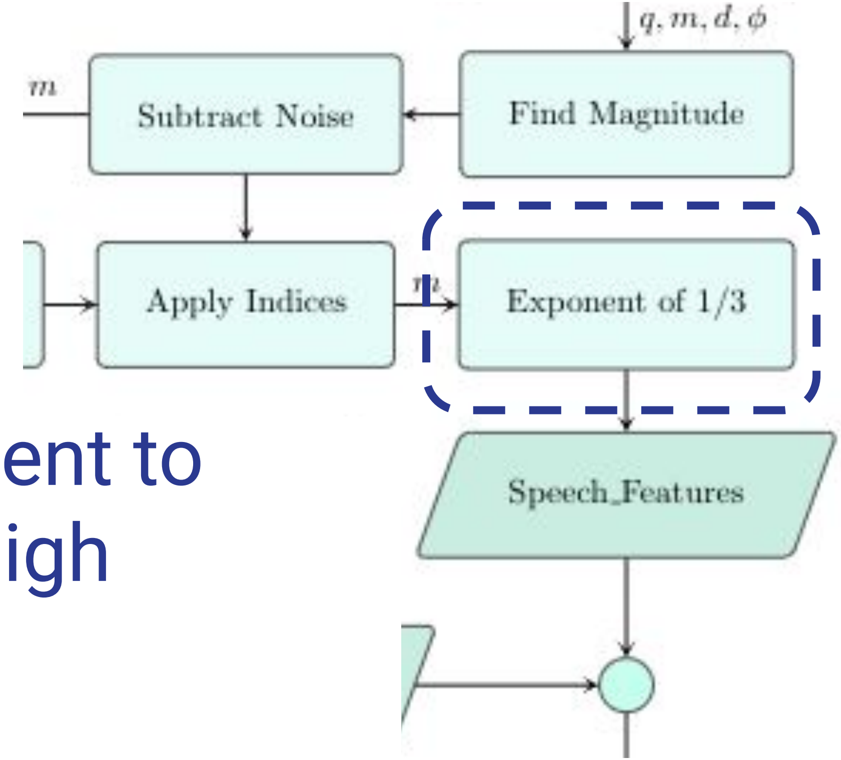
Data preparation for Neural Network



Use the index track to select the quefrency bins and extract the corresponding harmonics.



Speech Features Extraction



Apply fractional exponent to lower the gradient of high harmonic energies.

Data Preparation for Regression Learners

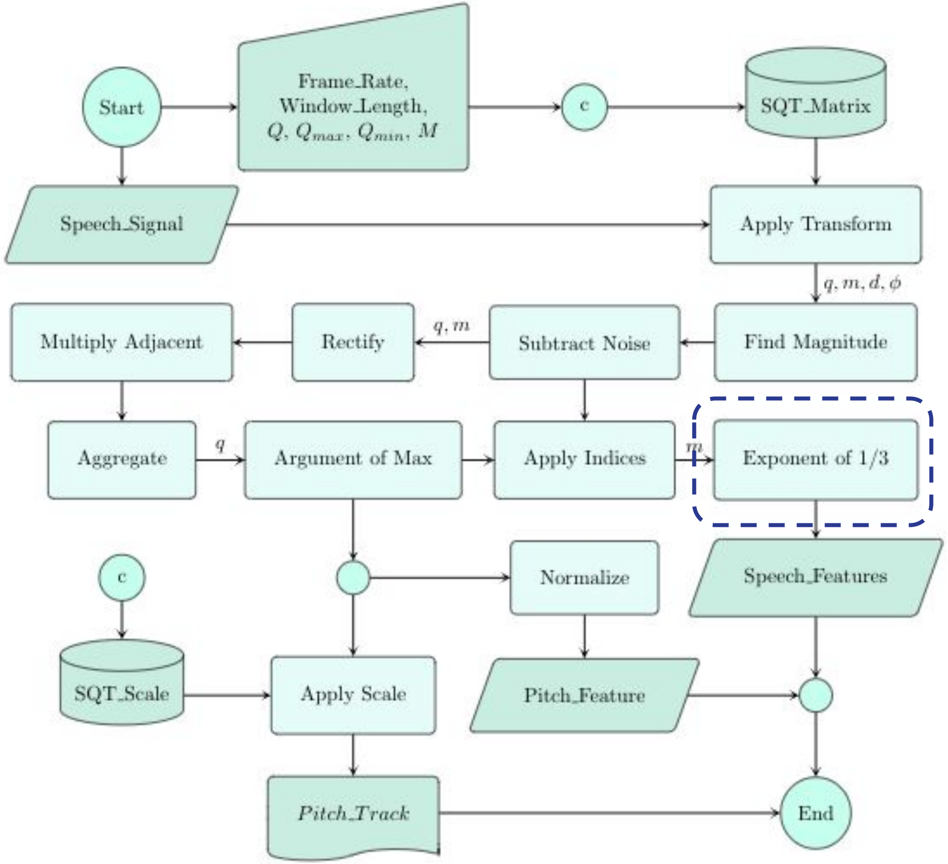


Figure 5: Procedure of Features' Extractions

That was how to extract the voice features by its quefrency speech components.

How to Hear the Speech Features

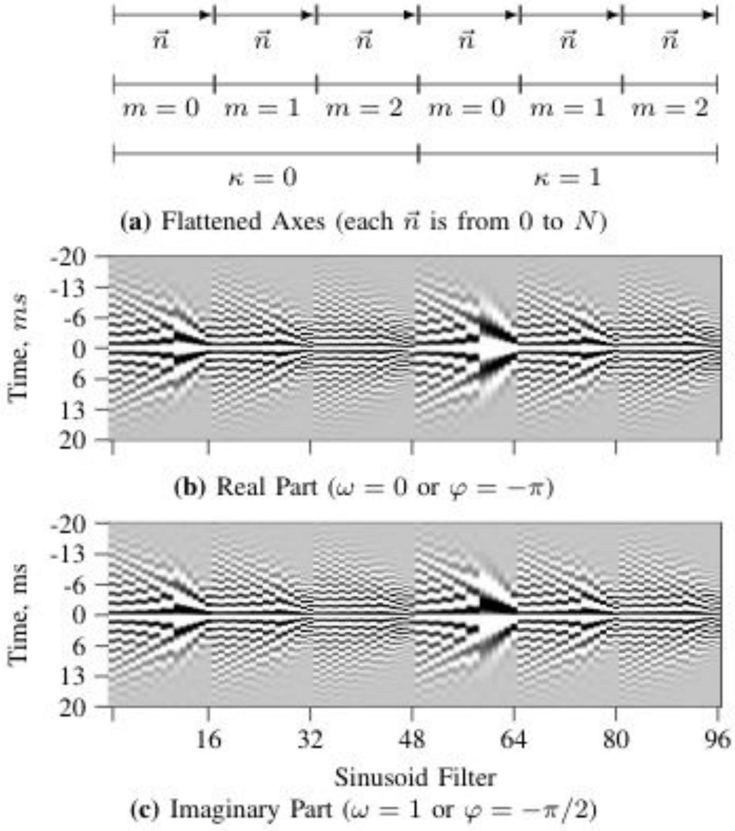
Speech Reconstruction Formula

$$\widehat{\text{Speech\_Signal}}(t) = \sum_{m} \text{Speech\_Features}^{3}[r, m] \cdot cos(2\pi \cdot t \cdot \text{Pitch\_Track}[r] \cdot m)$$

$$\begin{aligned} \text{where } t &= \text{Sample\_Number}/\text{Sample\_Rate} + t_0 \\ \text{and } r &= \lfloor t \cdot \text{Frame\_Rate} \rfloor \end{aligned}$$

The formula resembles a frequency-division multiplexing (FDM). The base signal, which conveys the vocal tract features, is distributed to the frequency impulse train. The power to the three is the reciprocal to the fractional exponent in the extraction phase. It is added since it is relevant to the gradient of a regression operation, and since it is important that the speech features be regenerated after they are machine learned.

Defining the SQT Matrix and Reciprocal Scales



(a) Flattened Axes (each $\vec{n}$ is from 0 to N)

(b) Real Part ($\omega = 0$ or $\varphi = -\pi$)

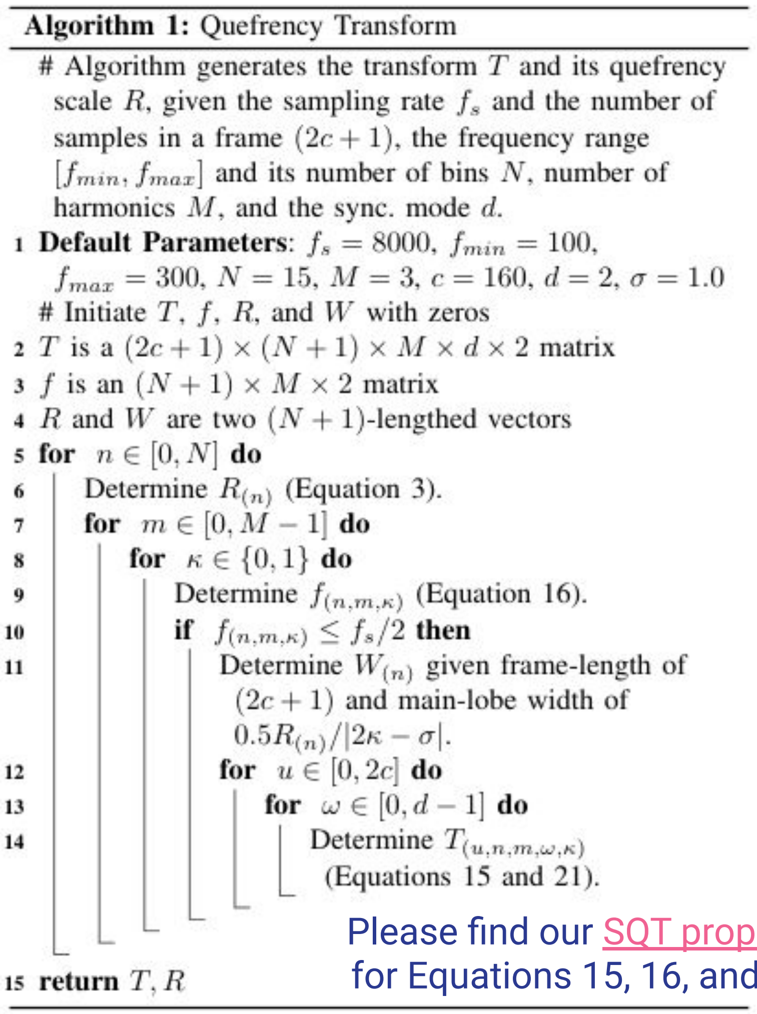
(c) Imaginary Part ($\omega = 1$ or $\varphi = -\pi/2$)

Figure 4: Windowed Transform Matrix, Accordant with the Default Parameters in Algorithm 1.

The SQT Matrix

The matrix is flattened for display, but it can be stored in a multidimensional array (i.e., numpy.ndarray).

it should be generated once at initiation or saved and loaded if you are producing high-definition cepstrograms.



```
Algorithm 1: Quefrency Transform
# Algorithm generates the transform $T$ and its quefrency
  scale $R$, given the sampling rate $f_s$ and the number of
  samples in a frame $(2c+1)$, the frequency range
  $[f_{min}, f_{max}]$ and its number of bins $N$, number of
  harmonics $M$, and the sync. mode $d$.
Default Parameters: $f_s = 8000$, $f_{min} = 100$,
  $f_{max} = 300$, $N = 15$, $M = 3$, $c = 160$, $d = 2$, $\sigma = 1.0$
# Initiate $T$, $f$, $R$, and $W$ with zeros
$T$ is a $(2c+1) \times (N+1) \times M \times d \times 2$ matrix
$f$ is an $(N+1) \times M \times 2$ matrix
$R$ and $W$ are two $(N+1)$-lengthed vectors
for $n \in [0, N]$ do
    Determine $R_{(n)}$ (Equation 3).
    for $m \in [0, M-1]$ do
        for $\kappa \in \{0, 1\}$ do
            Determine $f_{(n,m,\kappa)}$ (Equation 16).
            if $f_{(n,m,\kappa)} \leq f_s/2$ then
                Determine $W_{(n)}$ given frame-length of
                  $(2c+1)$ and main-lobe width of
                  $0.5R_{(n)}/|2\kappa - \sigma|$.
                for $u \in [0, 2c]$ do
                    for $\omega \in [0, d-1]$ do
                        Determine $T_{(u,n,m,\omega,\kappa)}$
                          (Equations 15 and 21).
return $T, R$
```

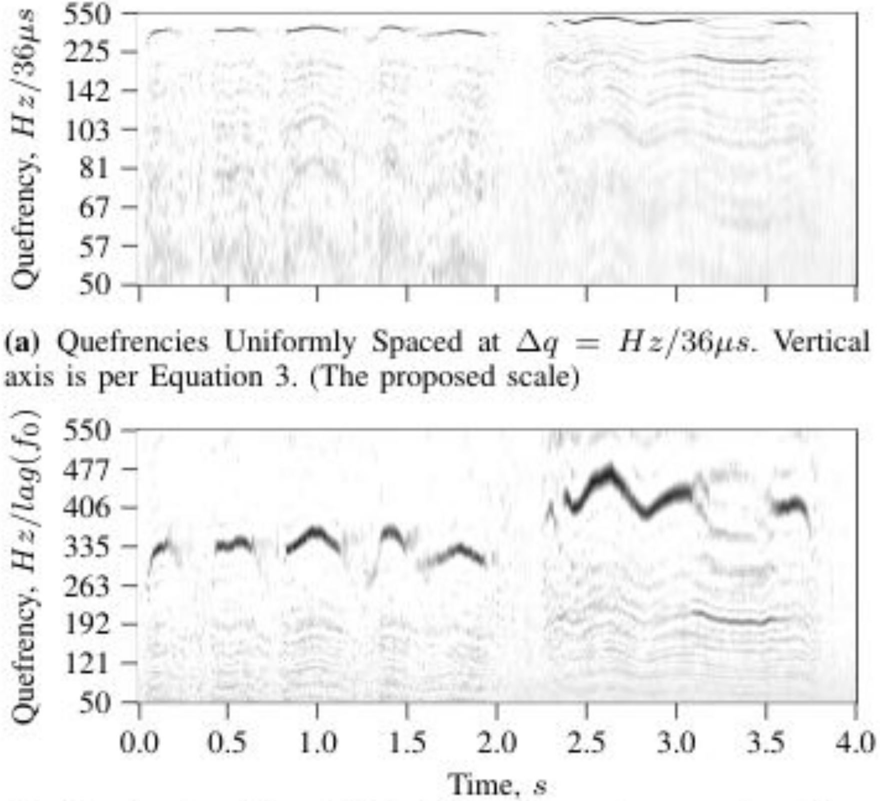
Please find our SQT proposal for Equations 15, 16, and 21.

How it is defined.

The definition is expressed in loops for clarification, but it can be prepared by parallel element-wise matrix operations (i.e., numpy.mgrid).

```
[u, q, m, k, w] =
    np.mgrid[
        -U:U+1 ,
        0:len(SQT_Scale),
        0:M,
        0:2,
        0:2 ];
```

How to Use numpy.mgrid

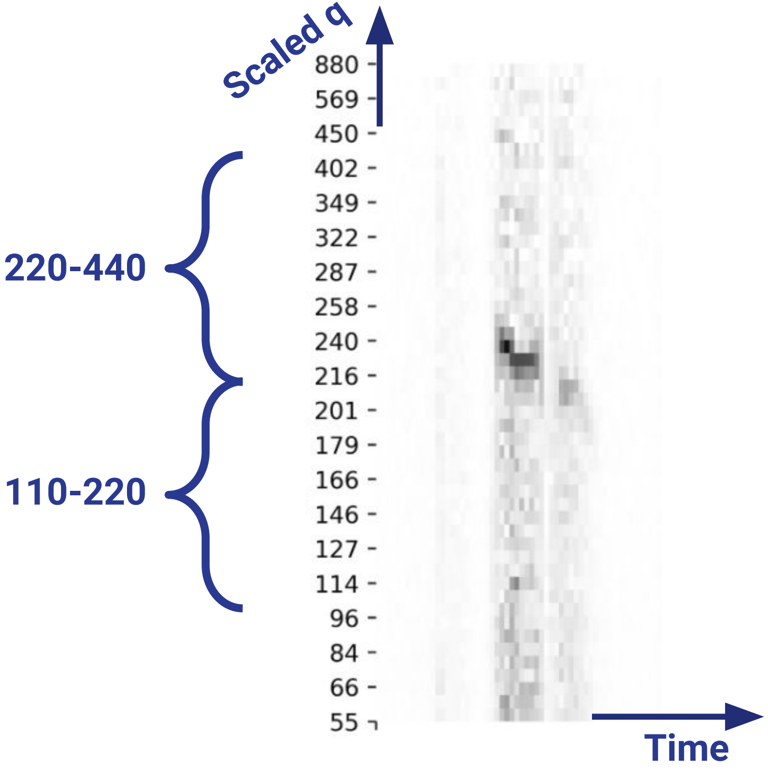


(a) Quefrencies Uniformly Spaced at $\Delta q = Hz/36\mu s$. Vertical axis is per Equation 3. (The proposed scale)

(b) Quefrencies Spaced Non-Linearly at Δq ranging from $Hz/392\mu s$ to $Hz/3\mu s$. Vertical axis is of regularly spaced frequencies. (The commonly used scale)

Reciprocal Scale (R-Scale)

Cepstrogram



Weighted Trade-Off

(70%R-Scale)

```
Q, Qmin, Qmax = [50, 55, 880]
alpha = 0.70

SQT_Scale =
    alpha/np.linspace(
    1/Qmin,
    1/Qmax,
    Q)
    + (1.0-alpha)*np.linspace(
    Qmin,
    Qmax,
    Q)
```

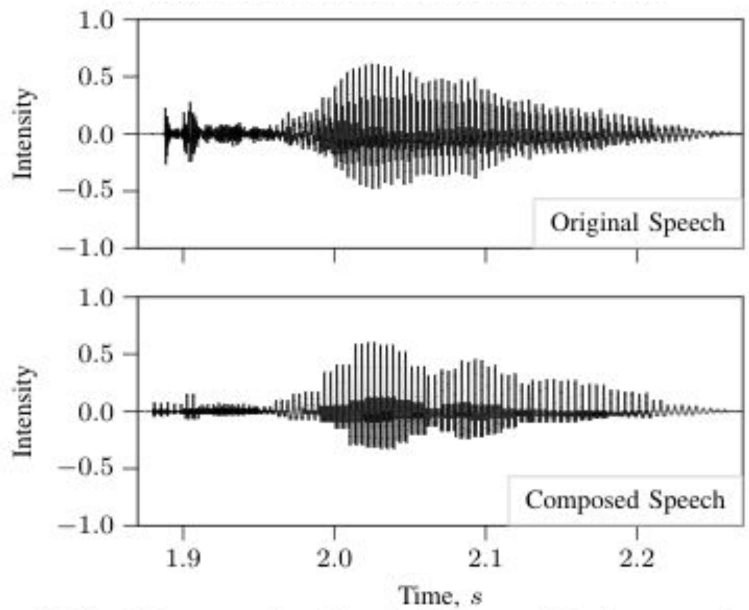
How to Generate the SQT_Scale Using Linear Space in Python & Matlab

SQT IS NOVEL

Seeing (and Hearing) Results

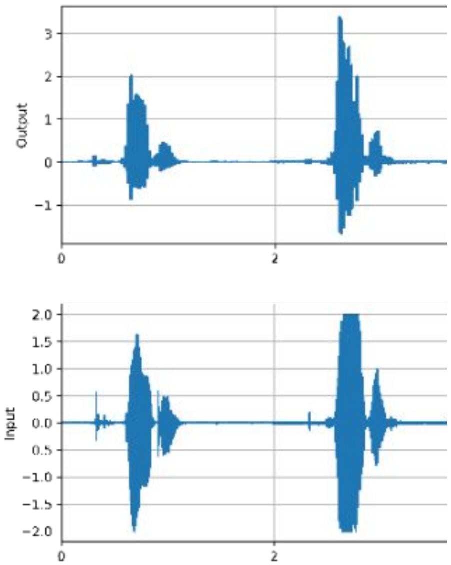
The Speech Reconstruction and Waveform

Gaussian Windows



(b) Signal Reconstruction (showing 0.4 second, "car" utterance)

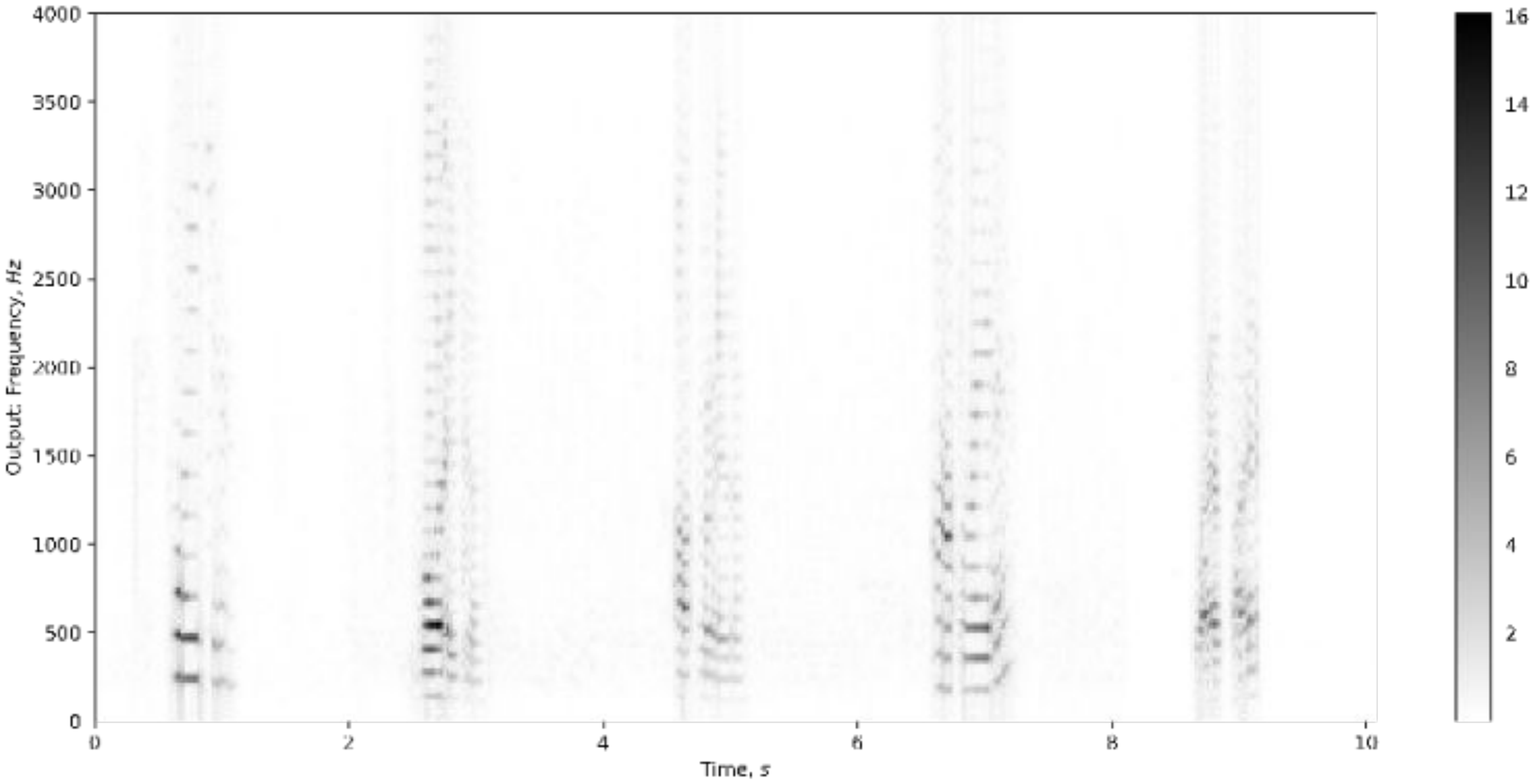
Rectangular Window



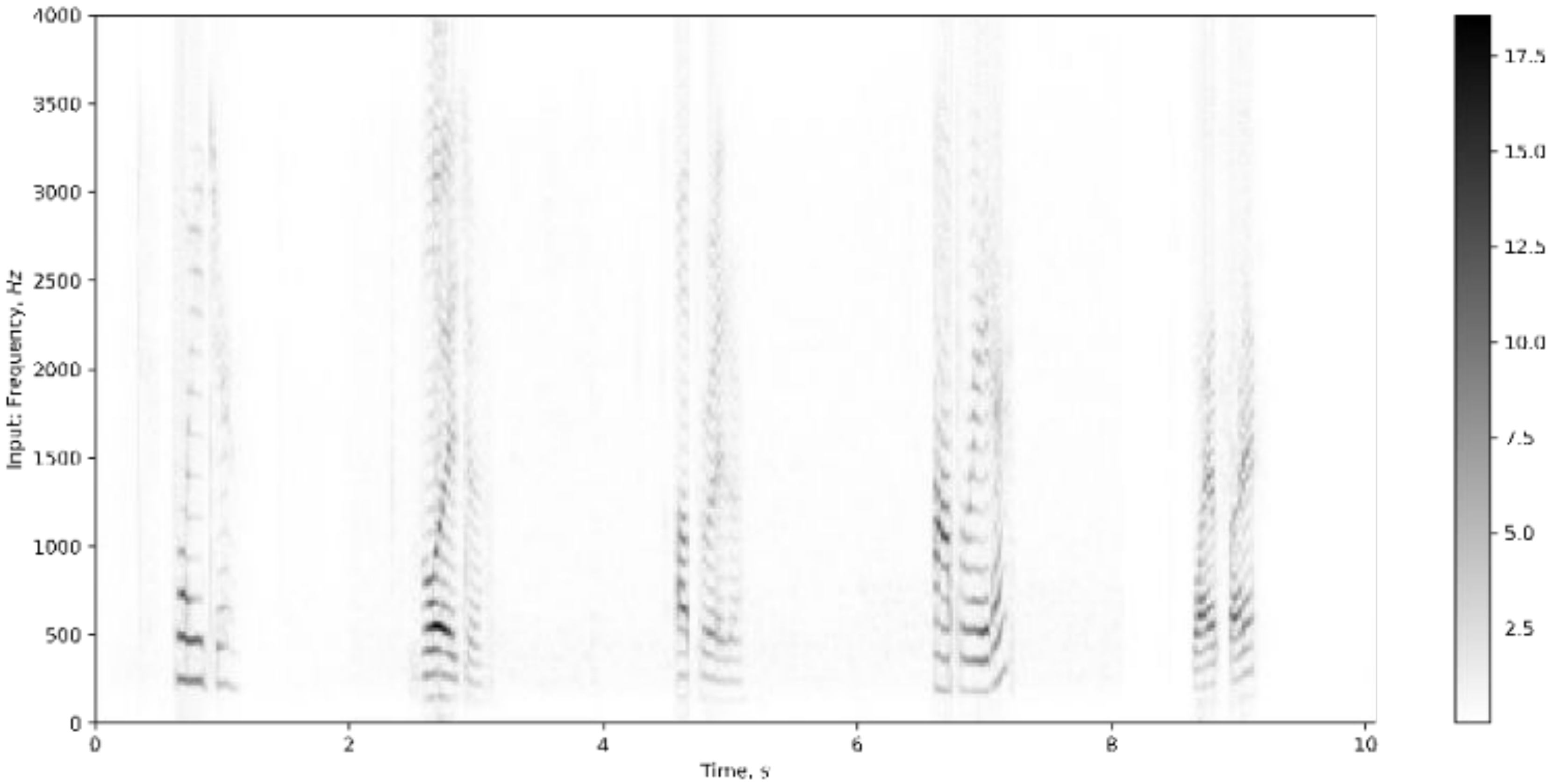
Test waveforms from the WUW Corpus.

Note: Parseval's Theorem holds true in the fixed rectangular window case.

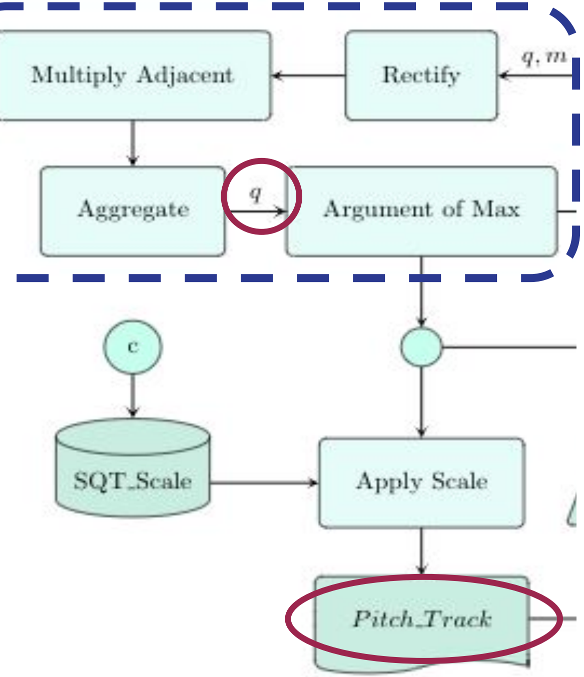
Spectrogram of the Reconstructed Signal



Spectrogram of the Input Signal

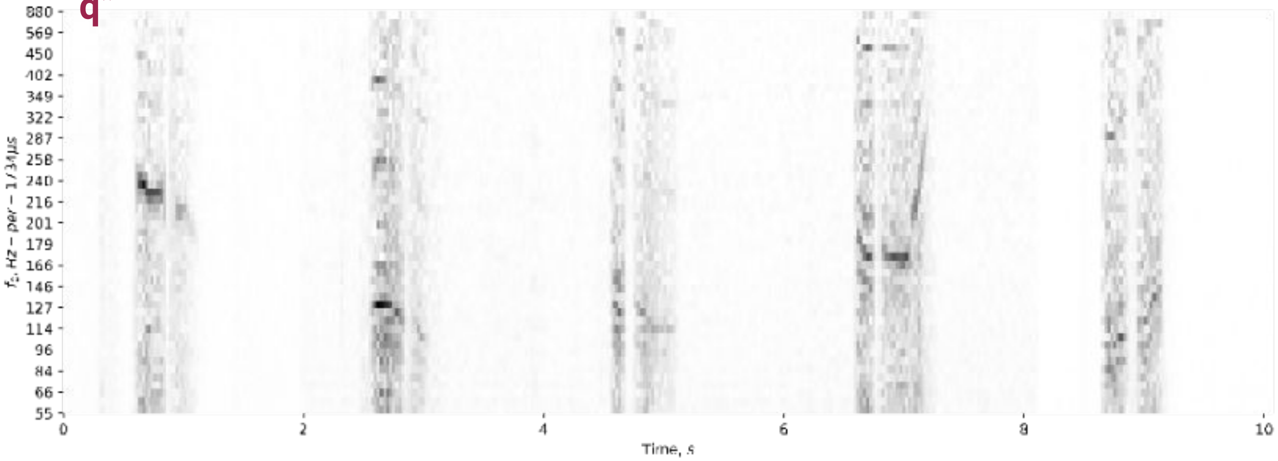


Index Track

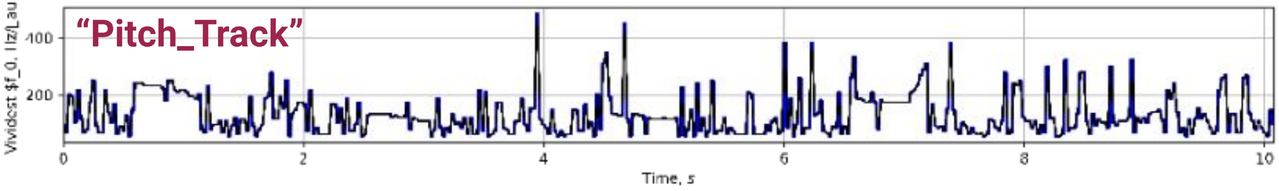


Cepstrogram

“q”

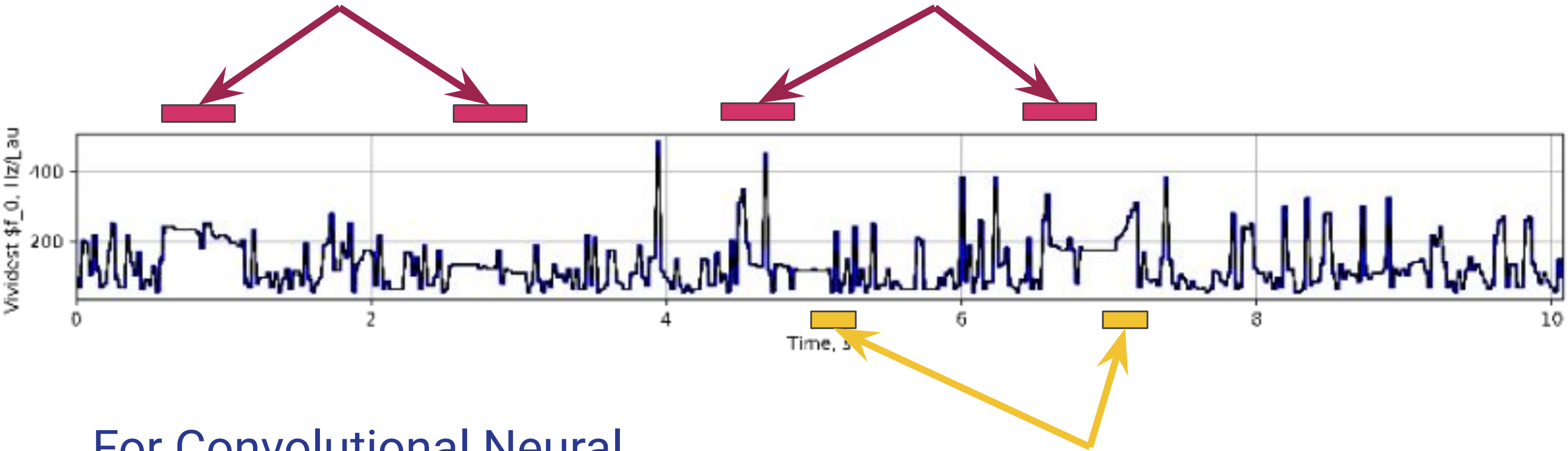


Vividest Fundamental Frequency



Speech Features to Detect in Pitch_Track

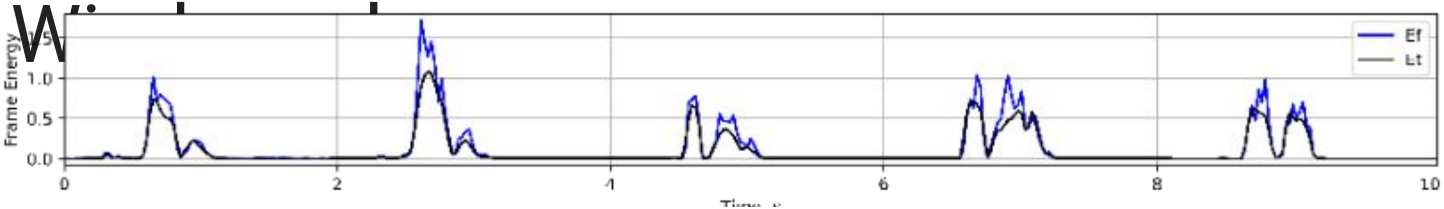
Pitch Pulses **Other Edge Patterns**



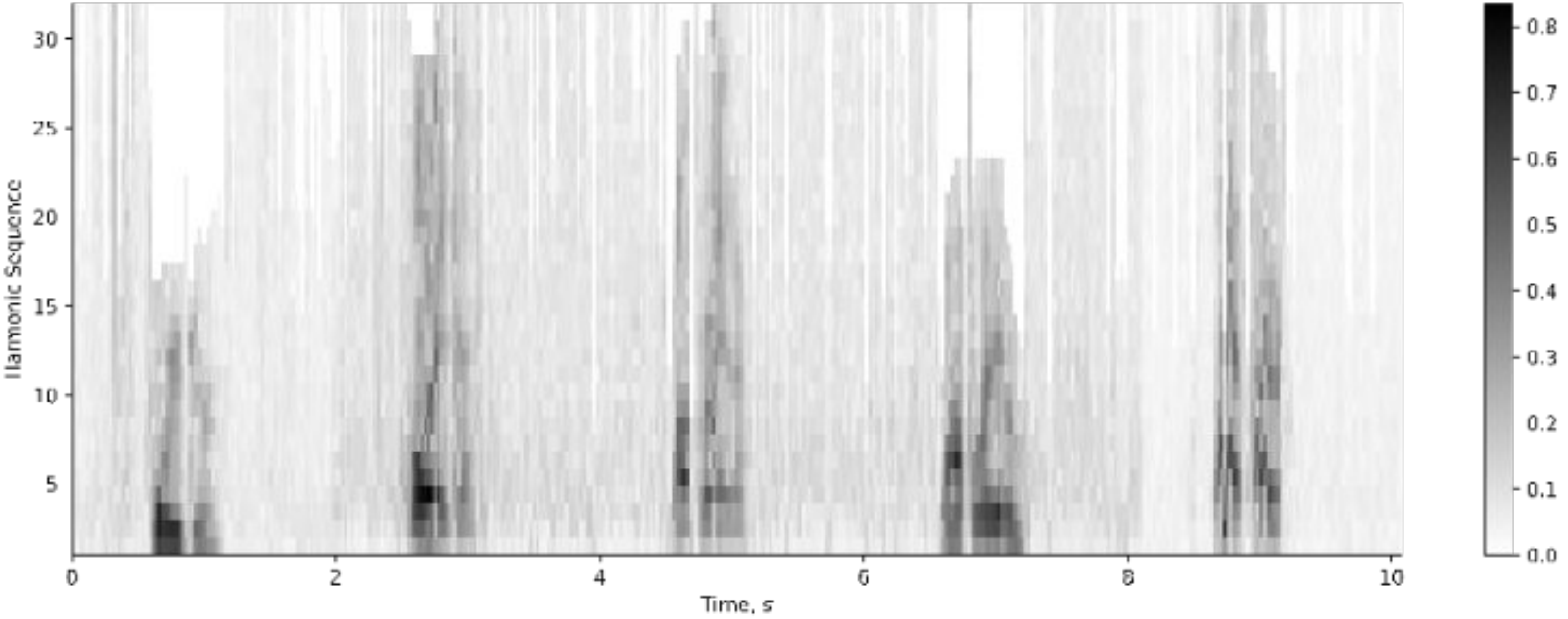
For Convolutional Neural Networks, Filter_Size should be in 50-250 ms.

Emotion Variant Pattern

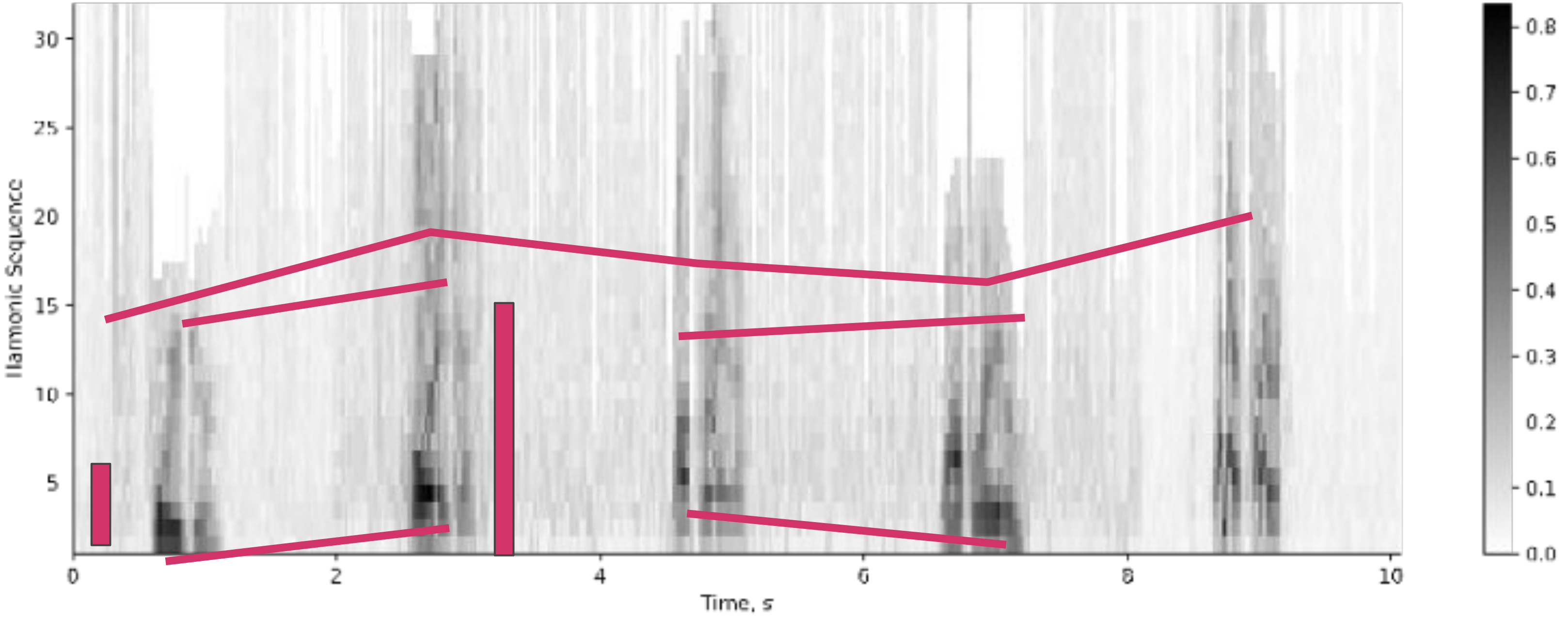
Total Energy Captured VS Total Energy W



Normalized Spectrogram (Harmonic Energies)



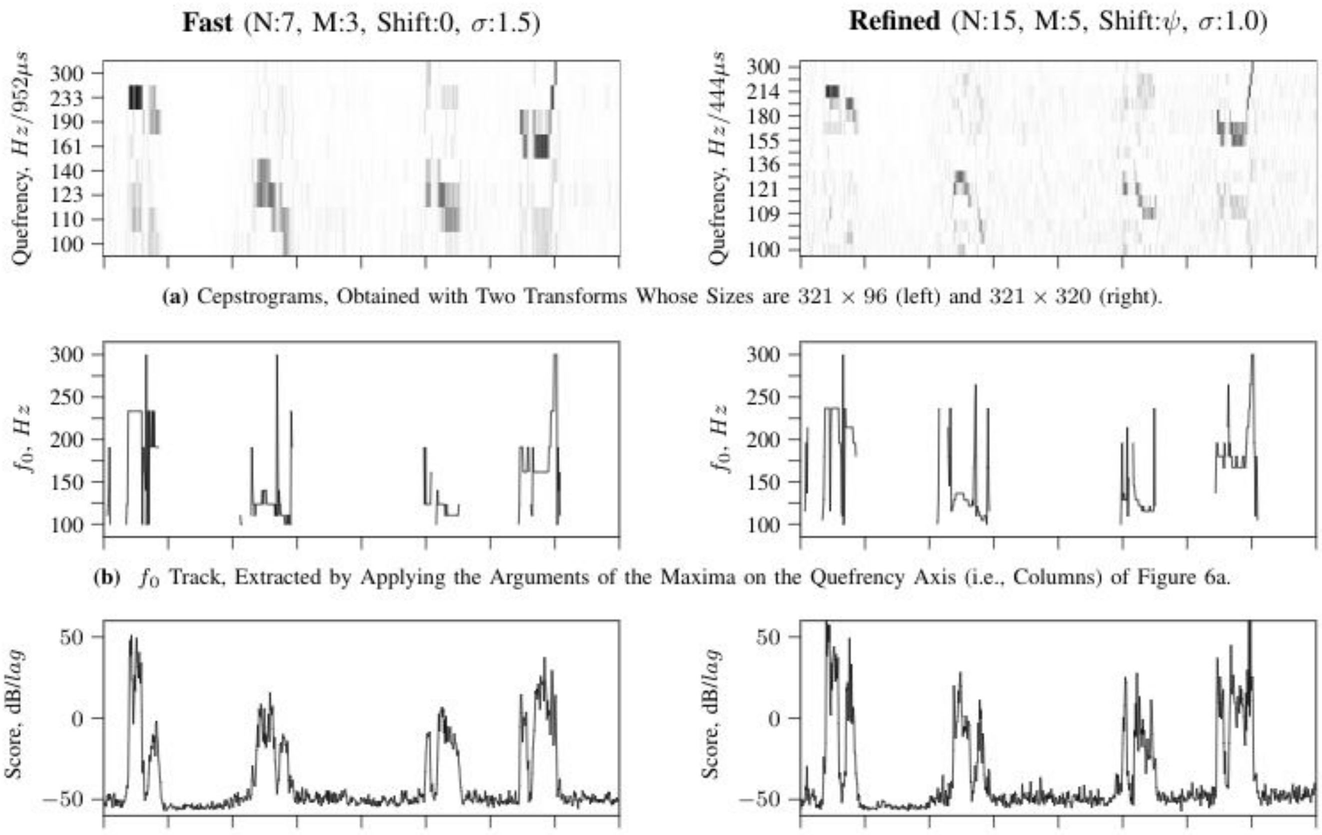
Lined Spectral Speech Features



Speaker's Characteristics Generates Harmonic Shift ⇨ For Convolutional Neural Networks, Filter_Size should be in 5-15 Harmonics.

Testing the Formula Under Various Scenarios

Robust Behavior Even When Computations Are Limited.



Fast (N:7, M:3, Shift:0, σ:1.5)

Refined (N:15, M:5, Shift:ψ, σ:1.0)

(a) Cepstrograms, Obtained with Two Transforms Whose Sizes are 321 × 96 (left) and 321 × 320 (right).

(b) f_0 Track, Extracted by Applying the Arguments of the Maxima on the Quefrency Axis (i.e., Columns) of Figure 6a.

(c) Detection Intensity Along the f_0 Track. (Global Maxima on the Columns)

It is expected to be compatible with Boosting and Dynamic Programing, because it divides the task of the speech feature extraction into smaller tasks.

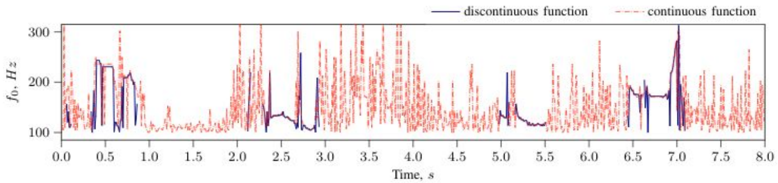
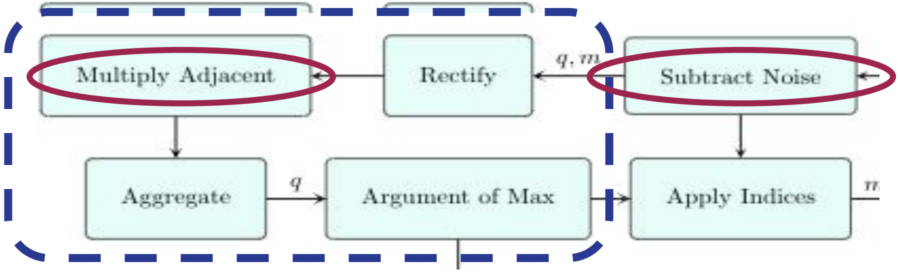
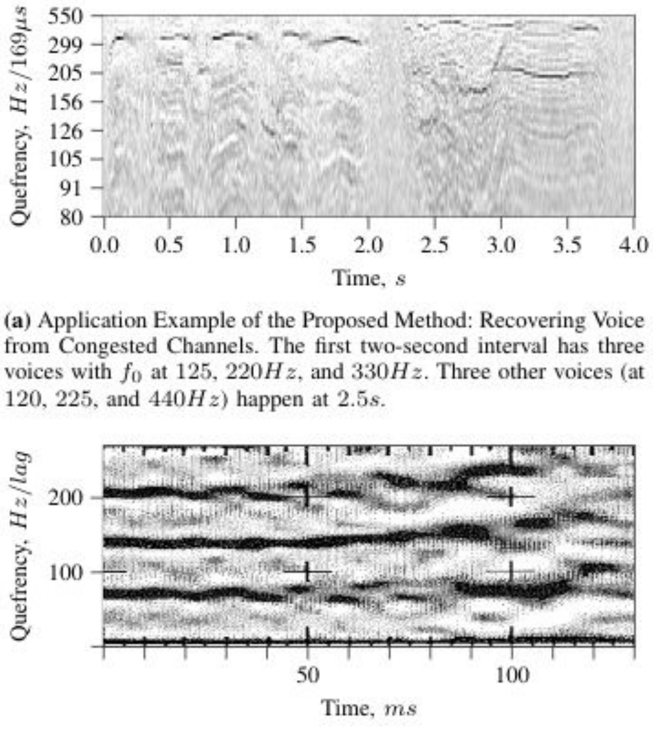


Figure 8: Comparison between the f_0 readings of the configurations in Figures 7a (discontinuous) and 7c (continuous).

SQT addresses the perplexity challenge of the over- and undertones by subtracting the adjacent noise and multiplying the adjacent harmonics.

Index Track



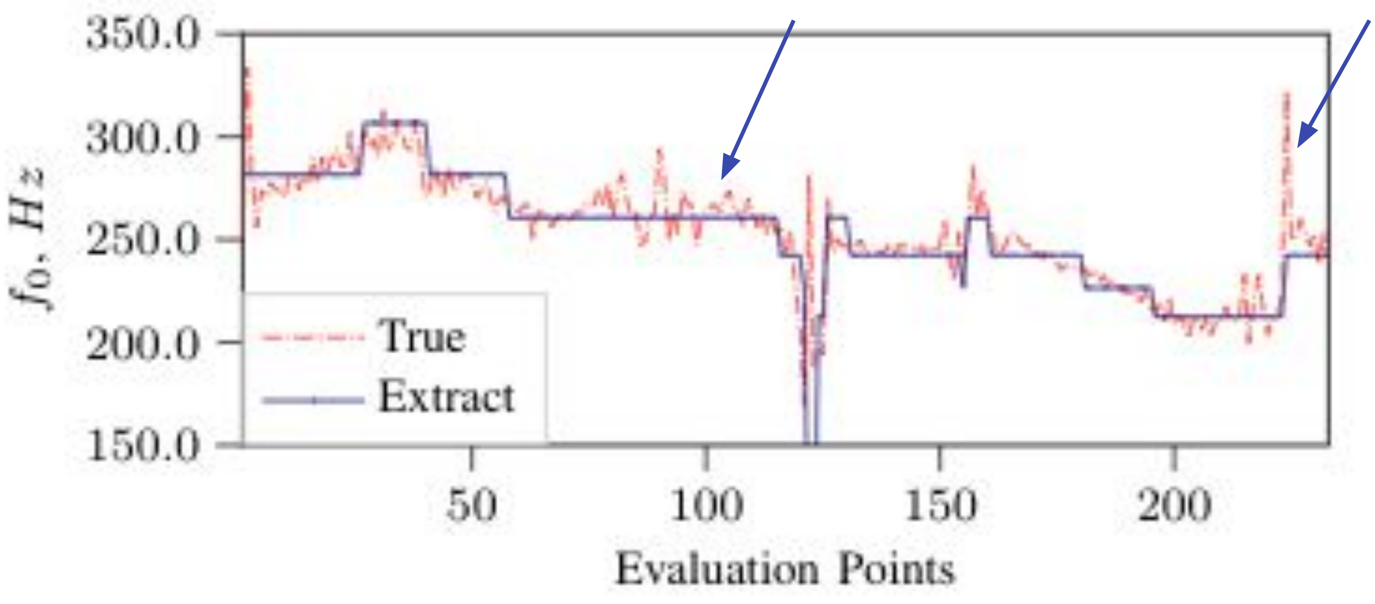


(a) Application Example of the Proposed Method: Recovering Voice from Congested Channels. The first two-second interval has three voices with f_0 at 125, $220Hz$, and $330Hz$. Three other voices (at 120, 225, and $440Hz$) happen at $2.5s$.

(b) f_0 Perplexity in Widespread Methods. An example [10] shows f_0 of a voice at $70Hz$, $140Hz$, or $210Hz$.

Figure 2: Cepstrogram Comparison

Visualizing the Pitch Extraction with the True Human Labeled FDA Corpus At Selected Time Intervals.



Quantization Error

(Re-) Sampling Error

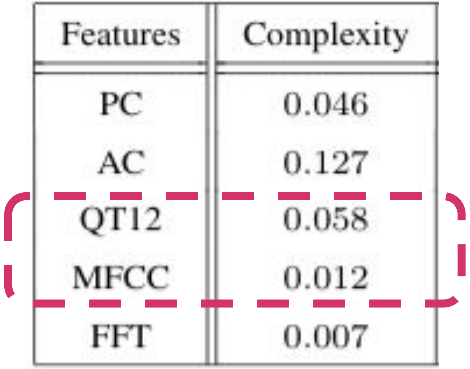
(a) Pitch Evaluation (showing the first 233 points)

Statistical Results

Three Methods:

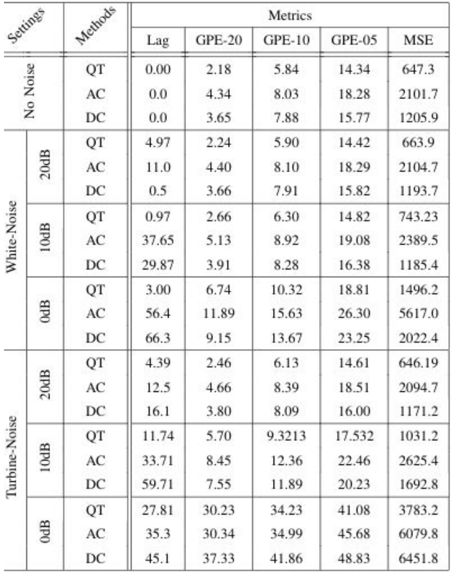
- One of our Matlab Implementations of the Quefrency Transform (QT) based method.
- Formal Matlab Implementation of an Amplitude Compression (AC) based method proposed by Gonzalez, Sira and Brookes, Mike in “A Pitch Estimation Filter Robust to High Levels of Noise (PEFAC).”
- Formal Matlab Implementation of a Pitch Contours (PC) based method proposed by Atal, Bishnu Saroop in “Automatic Speaker Recognition Based on Pitch Contours.”

Time Costs



Features	Complexity
PC	0.046
AC	0.127
QT12	0.058
MFCC	0.012
FFT	0.007

Performance of Currently Utilized Methods on the Pitch FDA Corpus and Under Several Ambient Noise Settings.



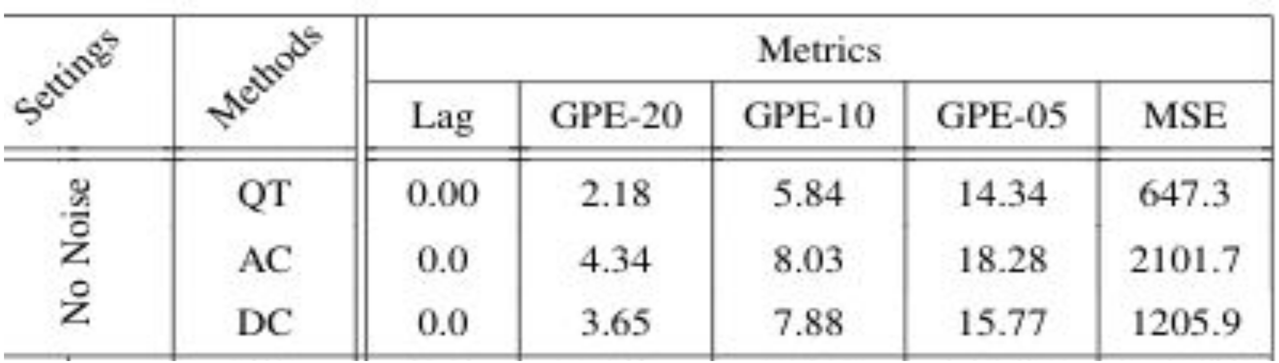
Settings		Methods	Metrics				
			Lag	GPE-20	GPE-10	GPE-05	MSE
No Noise		QT	0.00	2.18	5.84	14.34	647.3
		AC	0.0	4.34	8.03	18.28	2101.7
		DC	0.0	3.65	7.88	15.77	1205.9
White-Noise	20dB	QT	4.97	2.24	5.90	14.42	663.9
		AC	11.0	4.40	8.10	18.29	2104.7
		DC	0.5	3.66	7.91	15.82	1193.7
	10dB	QT	0.97	2.66	6.30	14.82	743.23
		AC	37.65	5.13	8.92	19.08	2389.5
		DC	29.87	3.91	8.28	16.38	1185.4
	0dB	QT	3.00	6.74	10.32	18.81	1496.2
		AC	56.4	11.89	15.63	26.30	5617.0
		DC	66.3	9.15	13.67	23.25	2022.4
Turbine-Noise	20dB	QT	4.39	2.46	6.13	14.61	646.19
		AC	12.5	4.66	8.39	18.51	2094.7
		DC	16.1	3.80	8.09	16.00	1171.2
	10dB	QT	11.74	5.70	9.3213	17.532	1031.2
		AC	33.71	8.45	12.36	22.46	2625.4
		DC	59.71	7.55	11.89	20.23	1692.8
	0dB	QT	27.81	30.23	34.23	41.08	3783.2
		AC	35.3	30.34	34.99	45.68	6079.8
		DC	45.1	37.33	41.86	48.83	6451.8

Table 2: FDA Evaluation

FDA Corpus **As It Is**

Three Performance Metrics:

- Lag of the best fitting calibration. Smaller interval is better.
- GPE (Gross Pitch Error) is a common speech metric of pitch performance. It accommodates an acceptable margin of absolute distance error, either 20, 10, or 5% of the target value. It’s the probability that certain error rate is exceeded. Smaller GPE value correlates with better methods.
- MSE (Mean Squared Error) is a common metric in signal processing. The smaller the error, the better.



Settings	Methods	Metrics				
		Lag	GPE-20	GPE-10	GPE-05	MSE
No Noise	QT	0.00	2.18	5.84	14.34	647.3
	AC	0.0	4.34	8.03	18.28	2101.7
	DC	0.0	3.65	7.88	15.77	1205.9

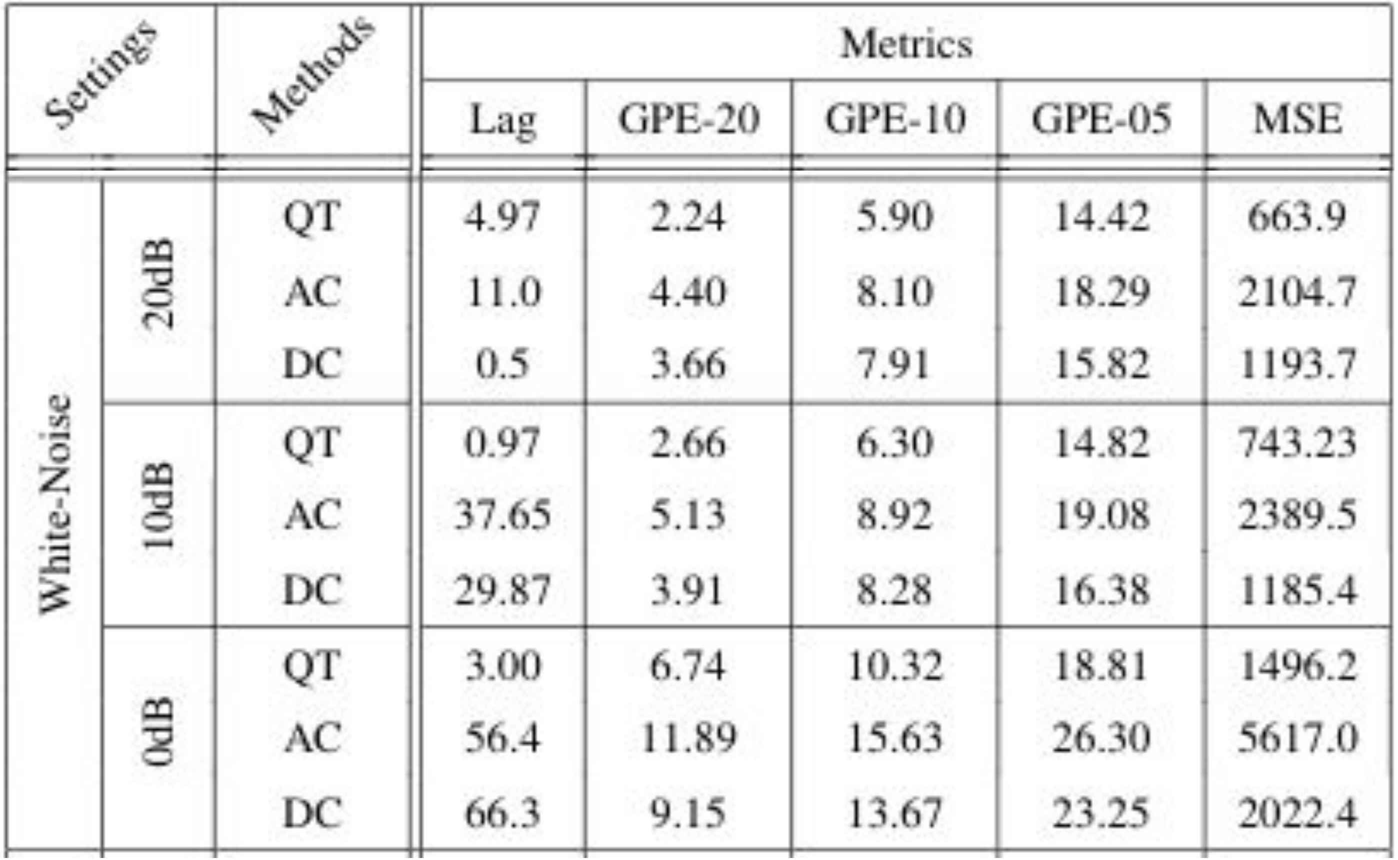
FDA Corpus With **Additive White Noise**

- The White Noise in dB is the higher, the better in terms of communication **channel**.

0dB means the speech signal and the noise have the same energy.

10dB is when the signal’s energy is ten times the noise’s.

20dB is when the signal’s is 100 times the noise’s.



Settings		Methods	Metrics				
			Lag	GPE-20	GPE-10	GPE-05	MSE
White-Noise	20dB	QT	4.97	2.24	5.90	14.42	663.9
		AC	11.0	4.40	8.10	18.29	2104.7
		DC	0.5	3.66	7.91	15.82	1193.7
	10dB	QT	0.97	2.66	6.30	14.82	743.23
		AC	37.65	5.13	8.92	19.08	2389.5
		DC	29.87	3.91	8.28	16.38	1185.4
	0dB	QT	3.00	6.74	10.32	18.81	1496.2
		AC	56.4	11.89	15.63	26.30	5617.0
		DC	66.3	9.15	13.67	23.25	2022.4

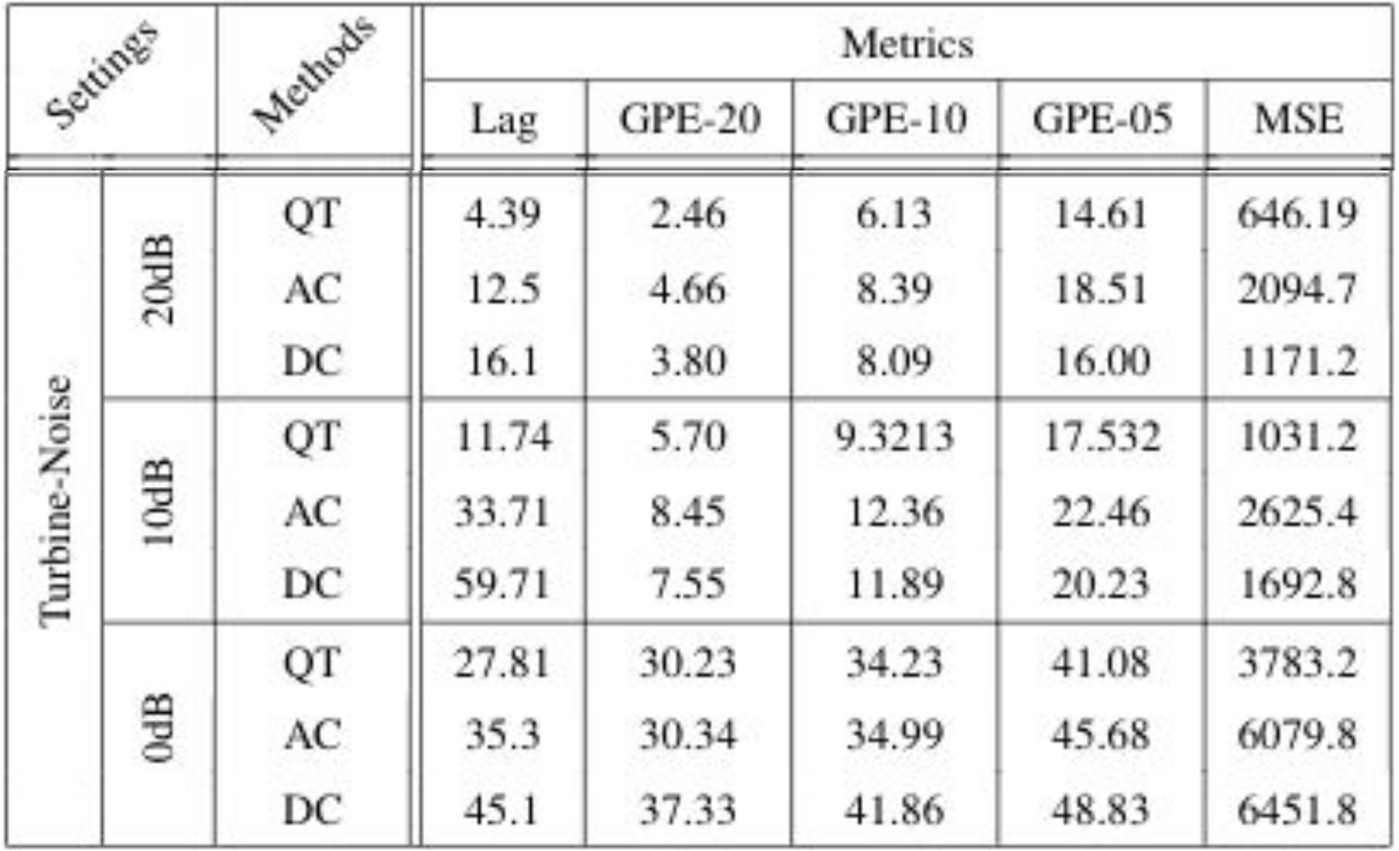
FDA Corpus With **Additive Turbine** Noise

- The Turbine-Noise in dB is the higher, the better in terms of the source **environment**.

0dB means the speech signal and the noise have the same energy.

10dB is when the signal’s energy is ten times the noise’s.

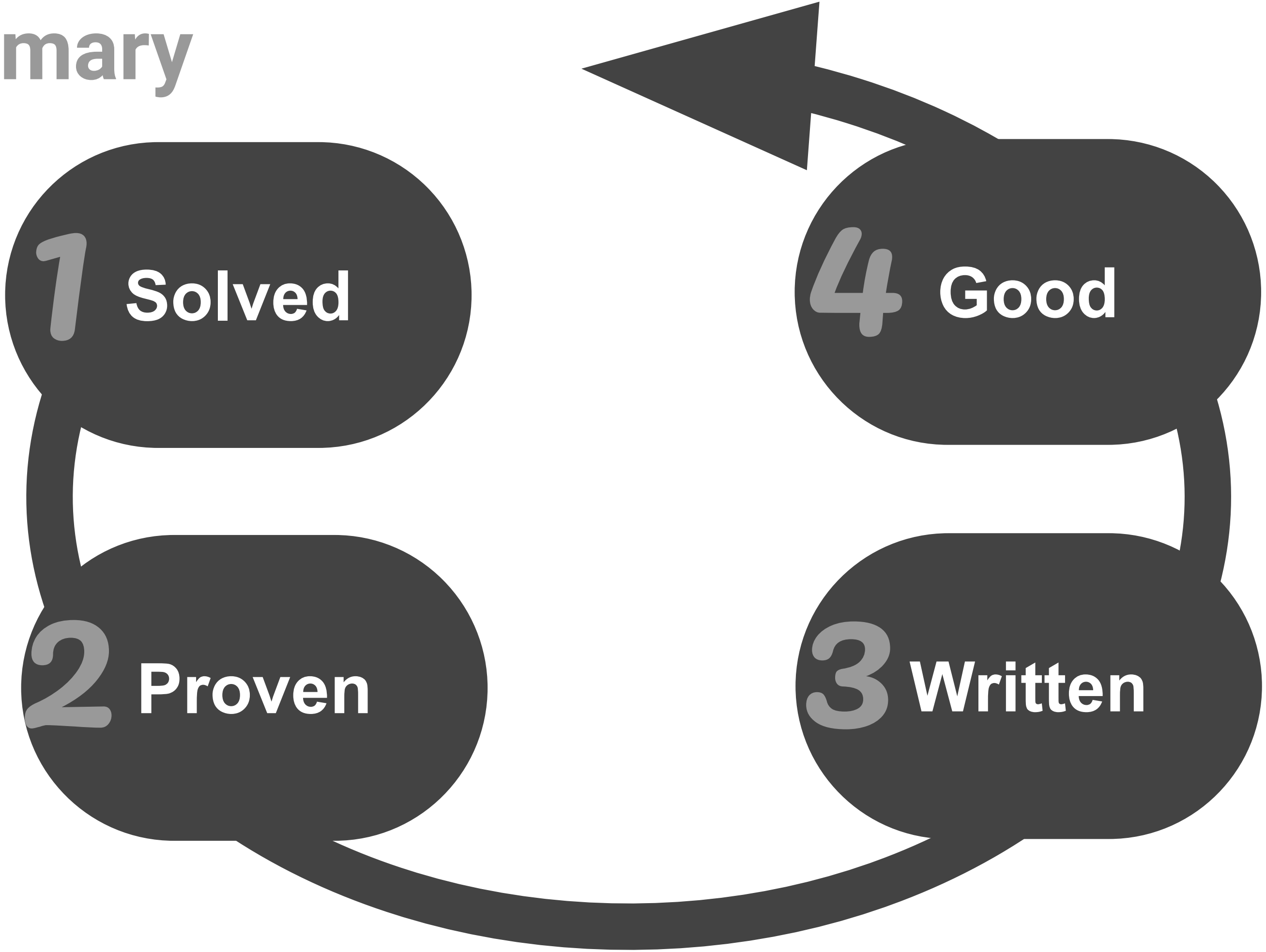
20dB is when the signal’s is 100 times the noise’s.



Settings		Methods	Metrics				
			Lag	GPE-20	GPE-10	GPE-05	MSE
Turbine-Noise	20dB	QT	4.39	2.46	6.13	14.61	646.19
		AC	12.5	4.66	8.39	18.51	2094.7
		DC	16.1	3.80	8.09	16.00	1171.2
	10dB	QT	11.74	5.70	9.3213	17.532	1031.2
		AC	33.71	8.45	12.36	22.46	2625.4
		DC	59.71	7.55	11.89	20.23	1692.8
	0dB	QT	27.81	30.23	34.23	41.08	3783.2
		AC	35.3	30.34	34.99	45.68	6079.8
		DC	45.1	37.33	41.86	48.83	6451.8

SQT IS ROBUST

Summary



1 Solved

2 Proven

3 Written

4 Good

Q&As

Thank You For Your Feedback.