

BENJAMIN VILLALONGA CORREA

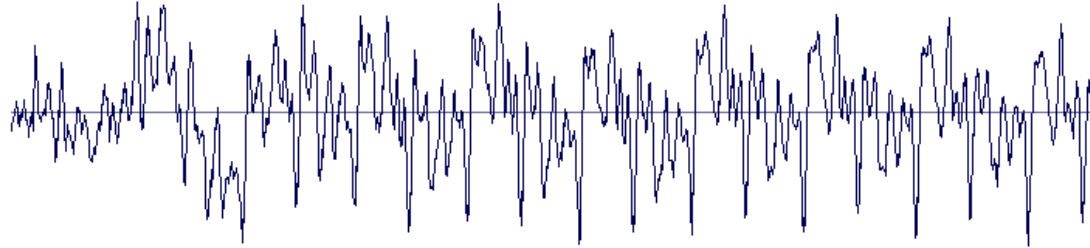
---

# PITCH CORRECTION

### THE PROBLEM

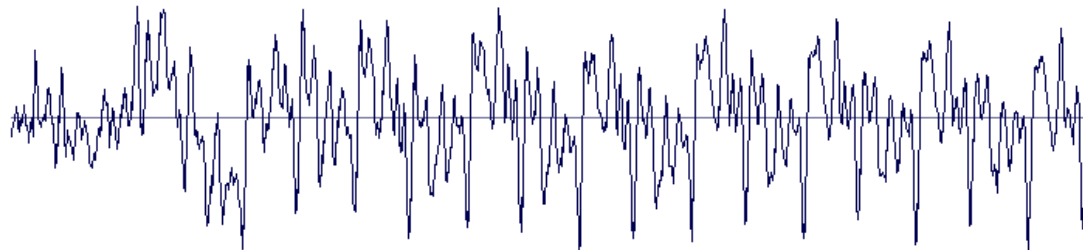
- ▶ Typically, musical systems reduce the continuous spectrum to a small set of “allowed” frequencies.
- ▶ Equal temperament: log of frequencies are equally spaced.
- ▶ Sound from a single source: combination of  $f$ ,  $2f$ ,  $3f$ , ...
- ▶ Fundamental belongs to the set of “allowed” frequencies.
- ▶ The problem being... not everybody can!!

# NAÏVE ALGORITHM

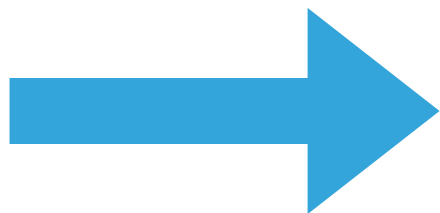


1. For each  $t$  get  $f$
2. Shift it to closest "allowed"  $f$
3. Go to step 1 for  $t+dt$

# PROBLEM WITH NAÏVE ALGORITHM



- ▶ We can't work with continuous variables
- ▶ Frequency is not a local observable
- ▶ How do we change frequencies without affecting time scales?



*Phase Vocoder* (Voice Decoder)

# PHASE VOCODER

Original vocoder (1930s) was intended for voice analysis and bandwidth reduction in communications.

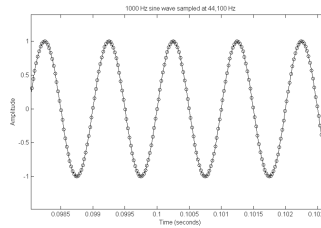


The Phase Vocoder is a multipurpose algorithm that allows time and frequency manipulations that are independent from each other.

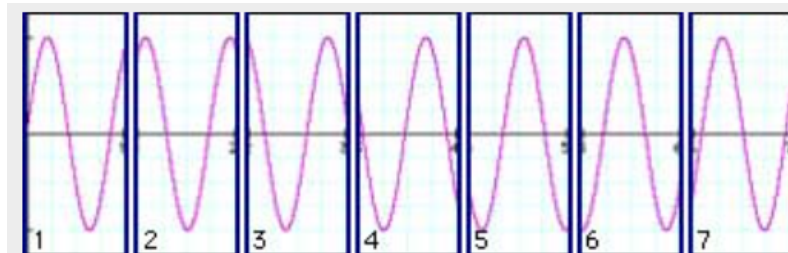
Can be used for pitch correction, time stretching, general pitch shifting...

## PHASE VOCODER (1)

Signal as array of amplitudes( $t$ ) (for example .wav file)



The signal is split in (small) frames. Each frame is analyzed separately to get its spectrum.

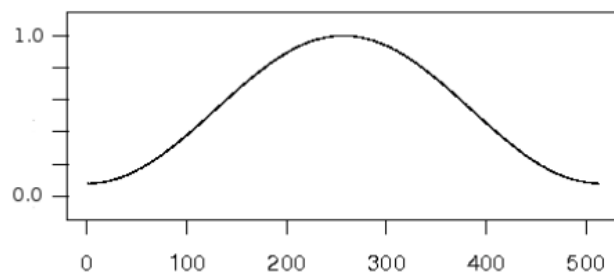


Discrete FT implies discretized set of frequencies. Create discontinuities that reflect on a wrong spectrum. New strategy!

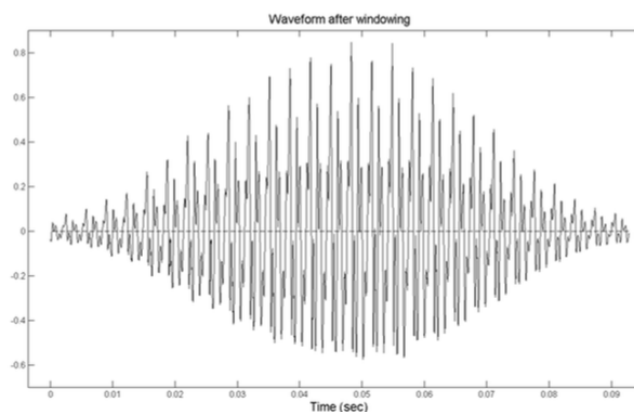
## PHASE VOCODER (2)

Each frame is windowed:

### Hanning

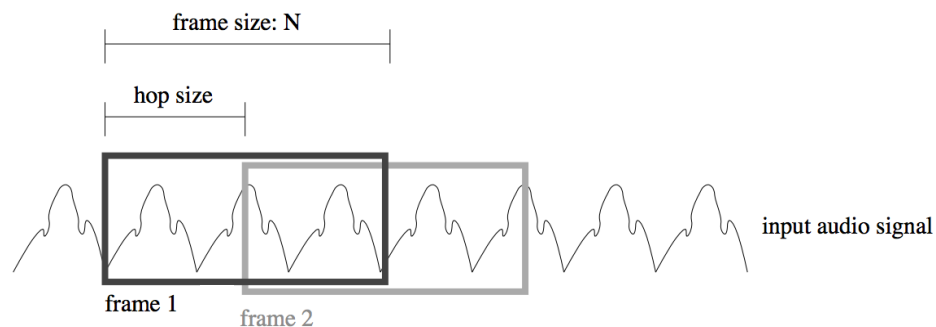


$$w(n) = 0.54 - 0.46\cos(2\pi n/N)$$



Optimized for getting better spectra (avoiding discontinuities).

Furthermore, frames overlap heavily:



## PHASE VOCODER (3)

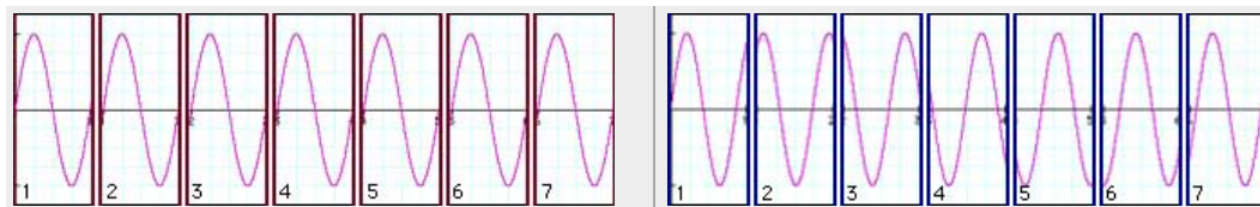
For each frame:

$$X(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-i2\pi nm/N} \omega(n)$$

where:

$$m = 0, 1, \dots, N - 1$$

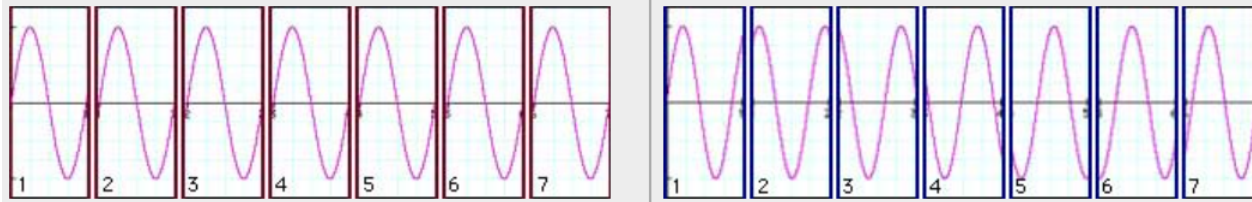
But  $\mathbf{x(n)}$  is real, so only necessary  $\mathbf{m} < \mathbf{N/2}$  (Nyquist frequency). Frequencies not in the discrete set of the FT will spread their amplitude among others. Can we do better?



Non matching frequencies can be recovered through the phase difference of  $\mathbf{X(m)}$ 's.

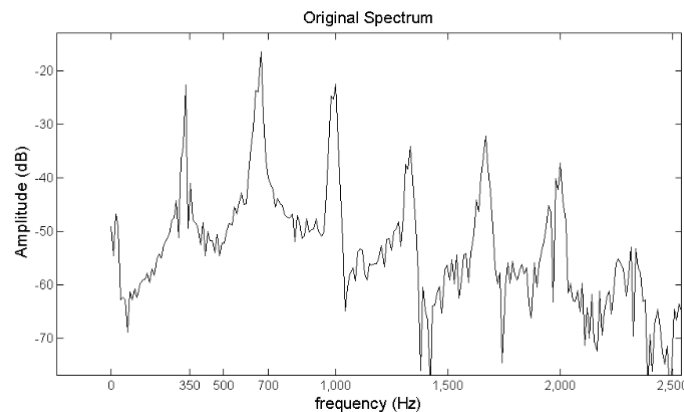


## PHASE VOCODER (4)

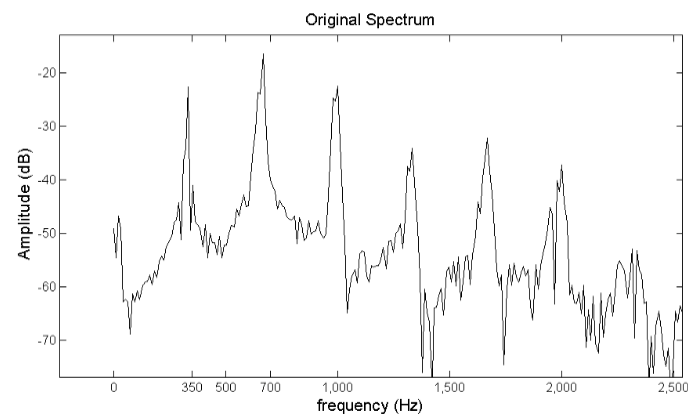


$$f_{real} = f_{discrete}(m) + [\theta(X(n)) - \theta(X(n-1))]_{-\pi}^{\pi} \cdot \frac{\text{sampling rate}}{2\pi N}$$

Store, for each frame ( $n$ ), an array with **real frequency** and **amplitude**, instead of **discrete frequency**, **amplitude** and **phase**.



# PHASE VOCODER (5)



I finally have a spectrum that I can shift as I want:

1. Multiply frequencies by factor and fit it again in closest frequency from discrete set
2. Drop all new frequencies that lay out of my discrete set (it is broad enough to capture audible frequencies)

Once I shift each frame's spectrum, I reconstruct the time-domain signal:

1. Adjust phase of each frequency to be smooth with previous frame (at that frequency)
2. Invert Discrete Fourier Transform
3. To avoid discontinuities coming from varying amplitudes of the same frequency, interpolate all frames that contribute to one point in  $\mathbf{t}$

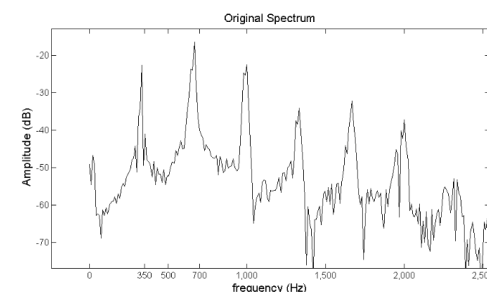
# APPLICATIONS

The applications will make use of different frequency shifts to achieve different effects:

- ▶ Pitch shift
- ▶ Pitch correction (auto-tune)
- ▶ Time stretch

For **Pitch correction** we need to identify what the fundamental frequency of each frame is. There are several algorithms for that. In the frequency domain, one simple implementation is...

# IDENTIFYING THE FUNDAMENTAL FREQUENCY OF A FRAME



1. Find maxima (peaks) of the spectrum
2. For a set of maxima, compute maximum common denominator
3. If the MCD belongs to the set, that is the fundamental. If not, go to step 4
4. The fundamental is the lowest maximum that is "sufficiently peaked"

After identifying the fundamental, we find the closest "allowed frequency" the shifting factor will be:

$$\text{shifting factor} = \frac{f_{\text{allowed}}}{f_{\text{sang}}}$$

### EXAMPLE

Original auto-tune exploited time-domain algorithms for real time applications (1997).

<https://www.youtube.com/watch?v=6fTh0WRJoX4>

# REFERENCES

- ▶ A good intuitive explanation: <http://blogs.zynaptiq.com/bernsee/pitch-shifting-using-the-ft/>
- ▶ Short accessible paper: [http://dave.ucsc.edu/physics195/thesis\\_2009/m\\_peimani.pdf](http://dave.ucsc.edu/physics195/thesis_2009/m_peimani.pdf)
- ▶ More rigorous, still accessible paper: [http://music.informatics.indiana.edu/media/students/kyung/kyung\\_paper.pdf](http://music.informatics.indiana.edu/media/students/kyung/kyung_paper.pdf)
- ▶ Long and complete scientific paper: [http://chamilo2.grenet.fr/inp/courses/PHELMAA35PMSPAR0/document/Projet\\_TFN/MoulinesLaroche1995.pdf](http://chamilo2.grenet.fr/inp/courses/PHELMAA35PMSPAR0/document/Projet_TFN/MoulinesLaroche1995.pdf)