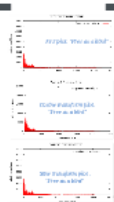
Nyoka Sound Library

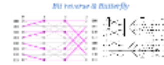
Digital signal processing is the mathematical manipulation of an information signal to modify or improve it in some way.



Cosine transform

Fast Fourier Transform

Sine transform

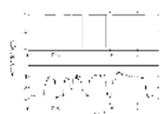


Bit reverse & Butterfly

SHUKRANI

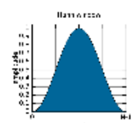
Pitch filter

Developed by Prof. Arturo Camacho. Ph.D



1. Select a window size

2. Weight the samples in neighborhood



$$w_N(n'-n) = 1 - \cos\left(\frac{\pi(n'-n)}{N+1}\right)$$

3. Apply K-means

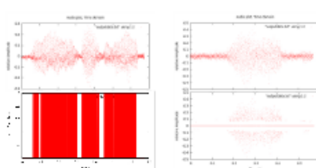
$$c_1(n,\mu,N) = \frac{\sum_{n'=-N}^{N} \mu(n'+n)\, s(n'+n)\, w_N(n')}{\sum_{n'=-N}^{N} \mu(n'+n)\, w_N(n')} \qquad c_0(n,\mu,N) = \frac{\sum_{n'=-N}^{N} [1-\mu(n'+n)]\, s(n'+n)\, w_N(n')}{\sum_{n'=-N}^{N} [1-\mu(n'+n)]\, w_N(n')}$$

4. Increase window size geometrically and repeat from 1

K-means algorithm

$$C = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Examples



5. Select the windows size that maximize distance between centroids

6. determine the class membership

$$m(n) = \left[\frac{s(n) - c_0(n, \mu^*(n), N^*(n))}{c_1(n, \mu^*(n), N^*(n)) - c_0(n, \mu^*(n), N^*(n))} > 0.5\right]$$

Arturo Agüí Chinchilla
Guillermo Cornejo Suárez
José Jobel Rodríguez Pineda

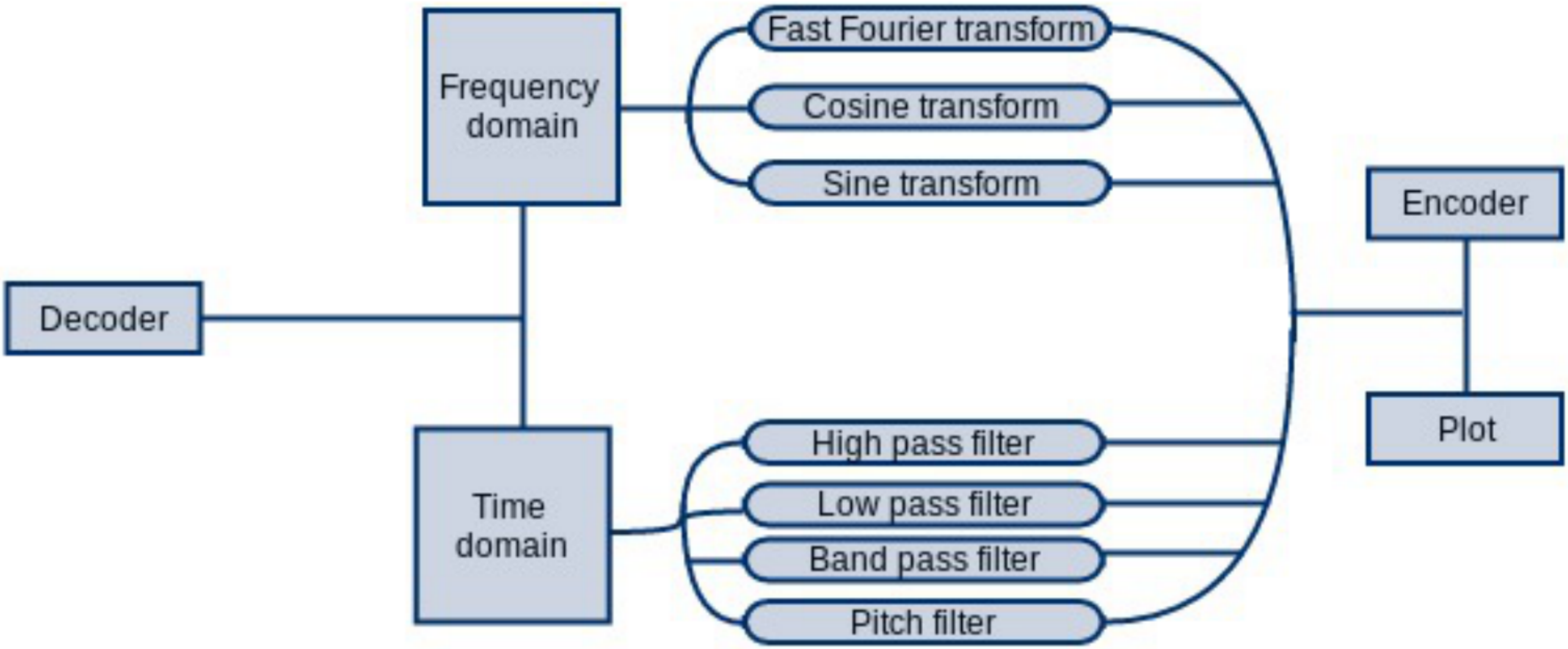
Nyoka Sound Library

Arturo Apú Chinchilla
Guillermo Cornejo Suárez
José Johel Rodríguez Pineda

Digital signal processing is the mathematical manipulation of an information signal to modify or improve it in some way.

Benefits

- Exact reproduction
- Easy manipulation


Frequency domain
Fast Fourier transform
Cosine transform
Sine transform
Decoder
Encoder
Plot
Time domain
High pass filter
Low pass filter
Band pass filter
Pitch filter

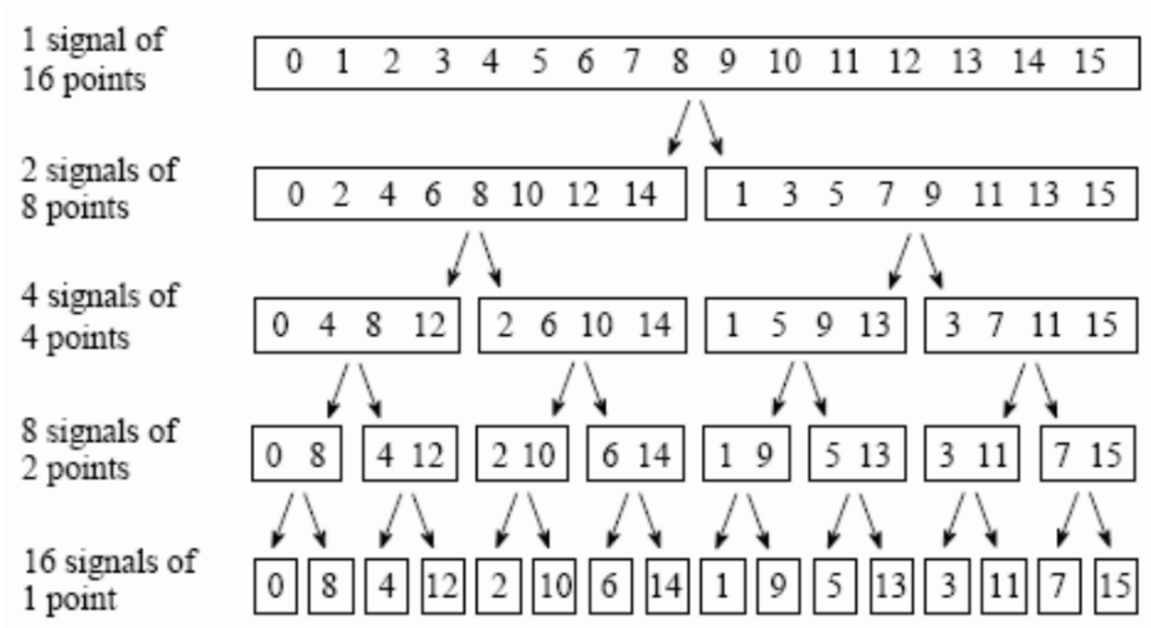
Fast Fourier Transform

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N}$$

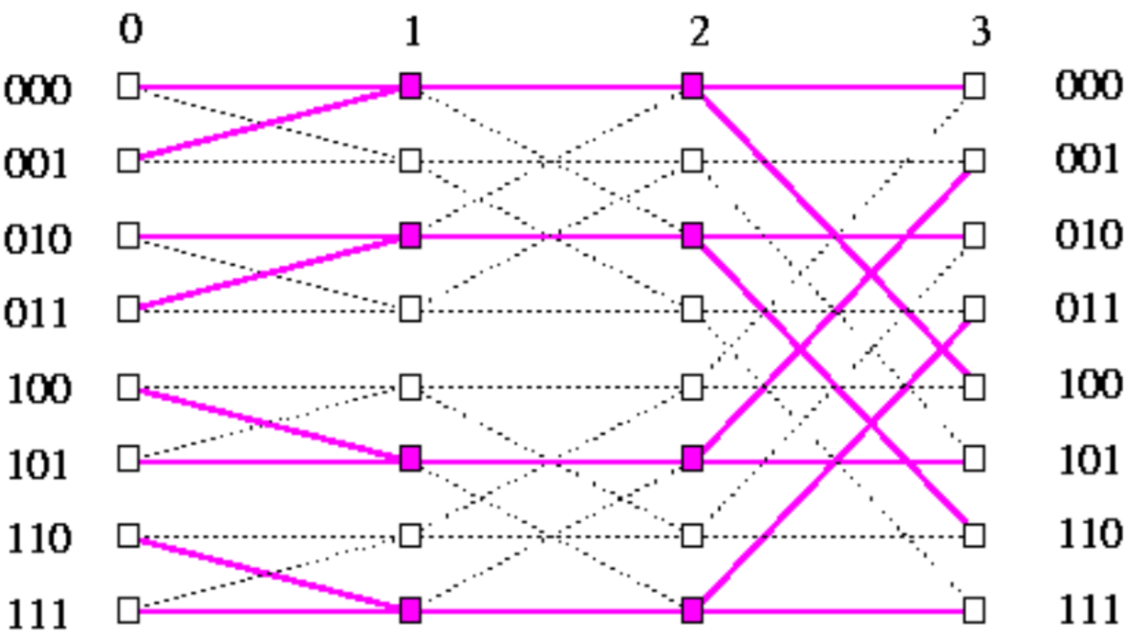
Danielson-Lanzczon lemma

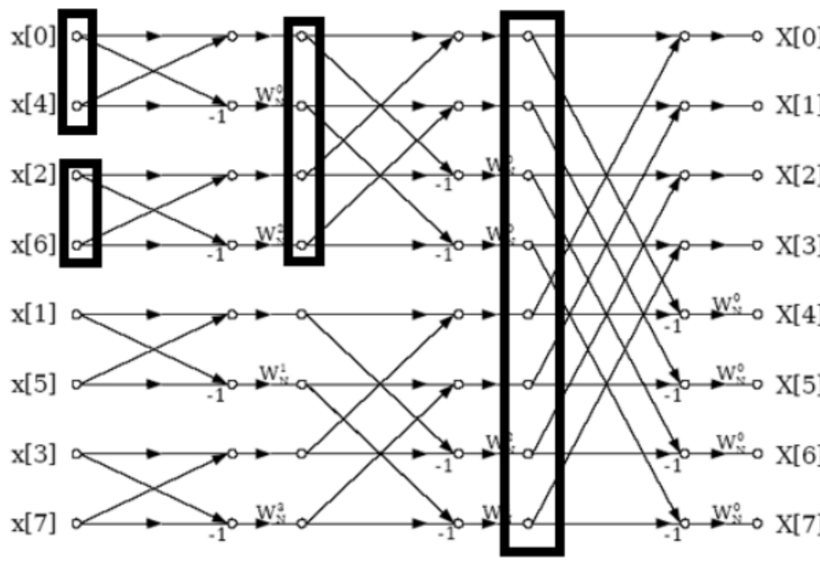
$$
\begin{aligned}
F_k &= \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \\
&= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \\
&= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j+1} \\
&= F_k^e + W^k \ F_k^o
\end{aligned}
$$

Sample numbers in normal order		Sample numbers after bit reversal	
Decimal	*Binary*	*Decimal*	*Binary*
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0100
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011
14	1110	7	0111
15	1111	15	1111



Bit reverse & Butterfly





$$F_k = \frac{1}{2}(f_0 + (-1)^k f_N) + \sum_{j=1}^{N-1} f_j \cos(\pi j k / N)$$

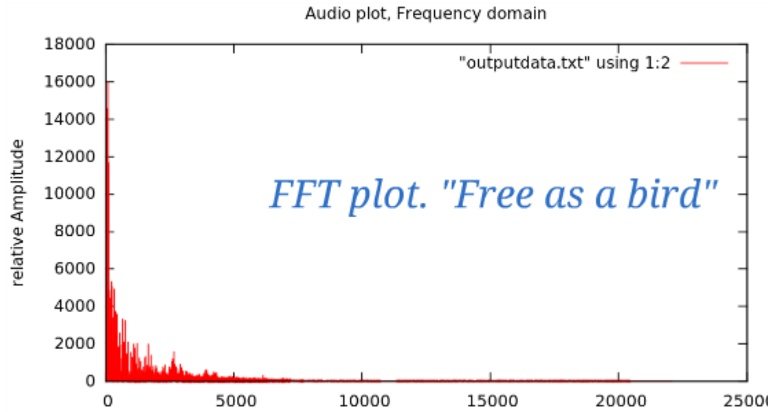
Cosine transform

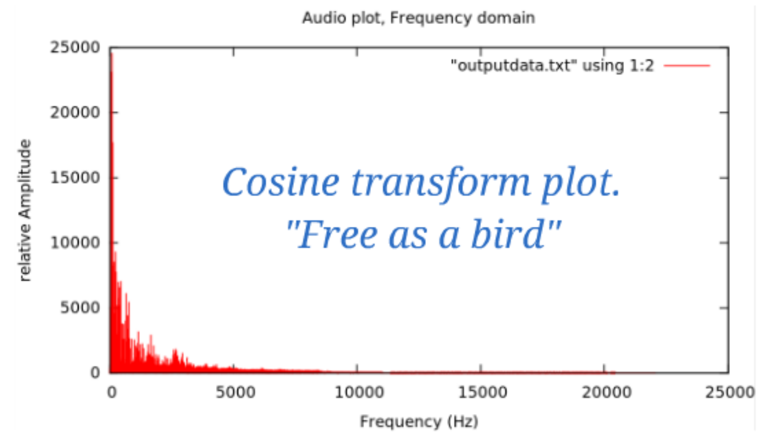
$$F_{2k} = R_k \qquad F_{2k+1} = F_{2k-1} + I_k$$

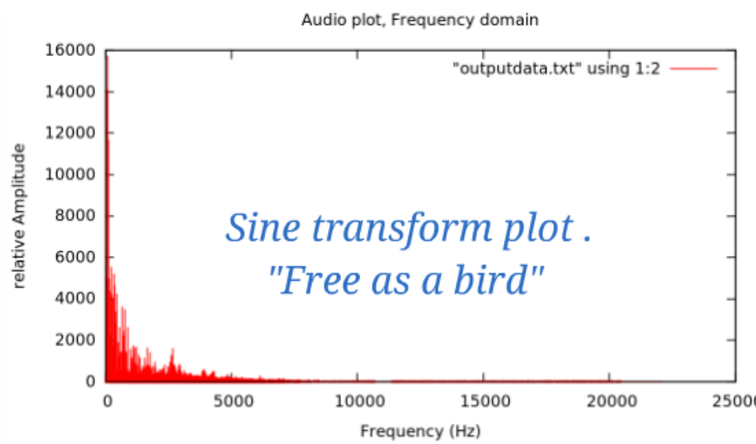
$$F_k = \sum_{j=0}^{2N-1} f_j e^{2\pi i jk/2N} = 2i \sum_{j=1}^{N-1} f_j \sin(\pi jk/N)$$

Sine transform

$$F_{2k} = I_k \qquad F_{2k+1} = F_{2k-1} + R_k$$

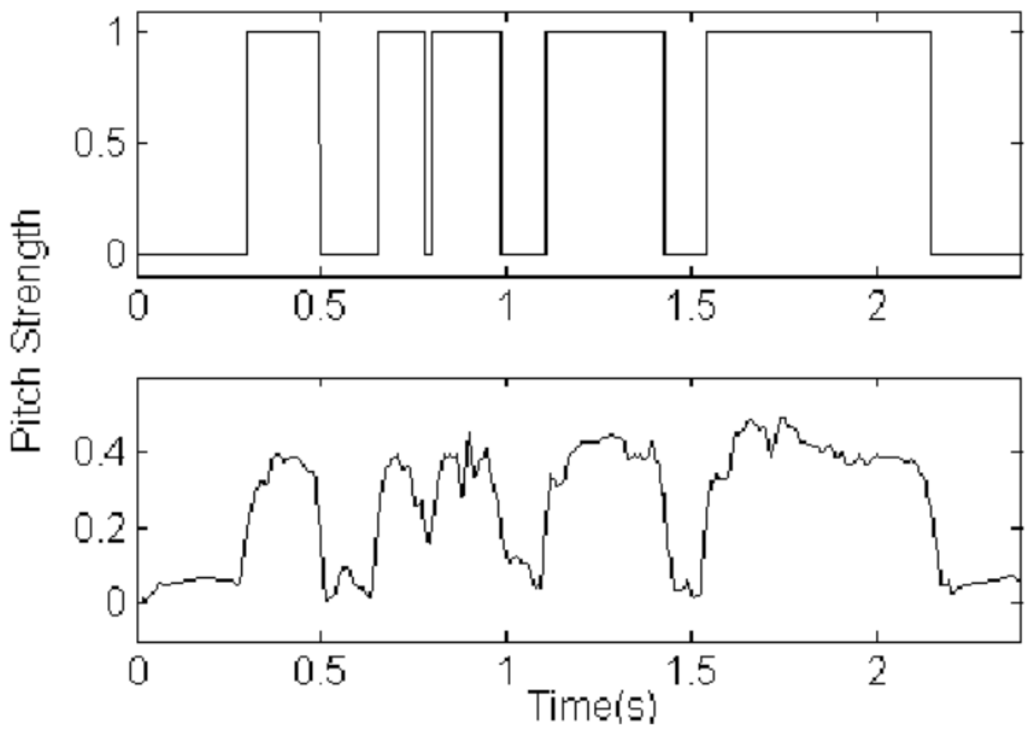

Audio plot, Frequency domain
"outputdata.txt" using 1:2
FFT plot. "Free as a bird"
relative Amplitude
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
5000
10000
15000
20000
25000


Audio plot, Frequency domain
"outputdata.txt" using 1:2
Cosine transform plot.
"Free as a bird"
relative Amplitude
25000
20000
15000
10000
5000
0
0
5000
10000
15000
20000
25000
Frequency (Hz)


Audio plot, Frequency domain
"outputdata.txt" using 1:2
Sine transform plot .
"Free as a bird"
relative Amplitude
16000
14000
12000
10000
8000
6000
4000
2000
0
0
5000
10000
15000
20000
25000
Frequency (Hz)

Pitch filter

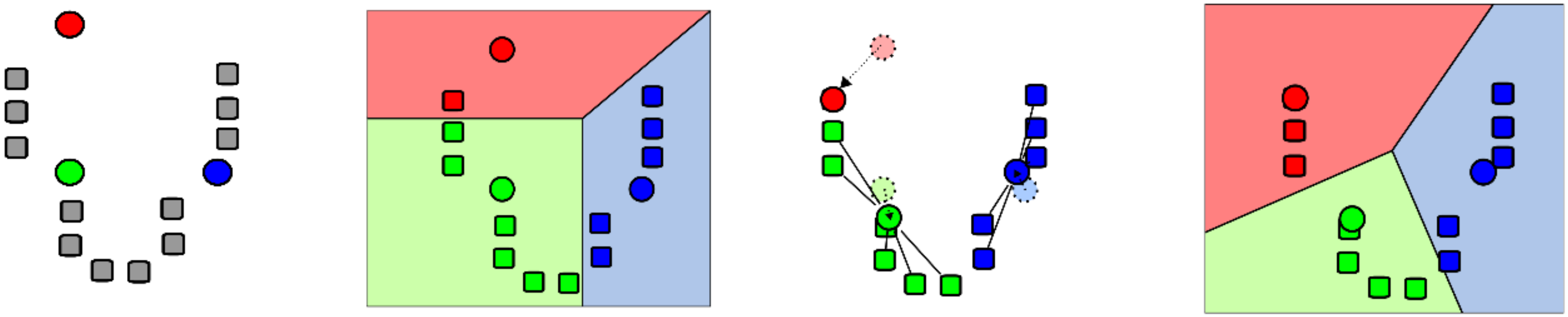
Developed by Prof. Arturo Camacho. Ph.D



PREZI

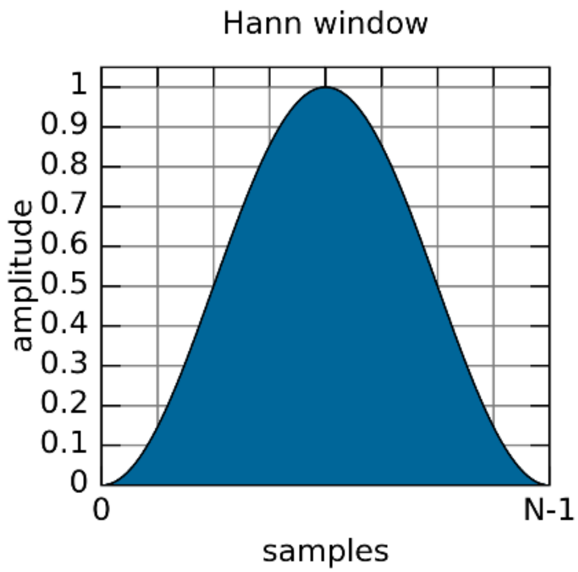
K-means algorithm

$$\mathbf{C} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k}{k}$$



1. Select a window size

2. Weight the samples in neighborhood



$$w_N(n'-n) = 1 + \cos\left(\frac{\pi(n'-n)}{N+1}\right)$$

3. Apply K-means

$$c_1(n,\mu,N)=\frac{\sum_{n'=-N}^{N}\mu(n'+n)s(n'+n)w_N(n')}{\sum_{n'=-N}^{N}\mu(n'+n)w_N(n')}$$

$$c_0(n,\mu,N)=\frac{\sum_{n'=-N}^{N}\left[1-\mu(n'+n)\right]\,s(n'+n)w_N(n')}{\sum_{n'=-N}^{N}\left[1-\mu(n'+n)\right]\,w_N(n')}$$

4. Increase window size geometrically and repeat from 1

5. Select the windows size that maximize distance between centroids

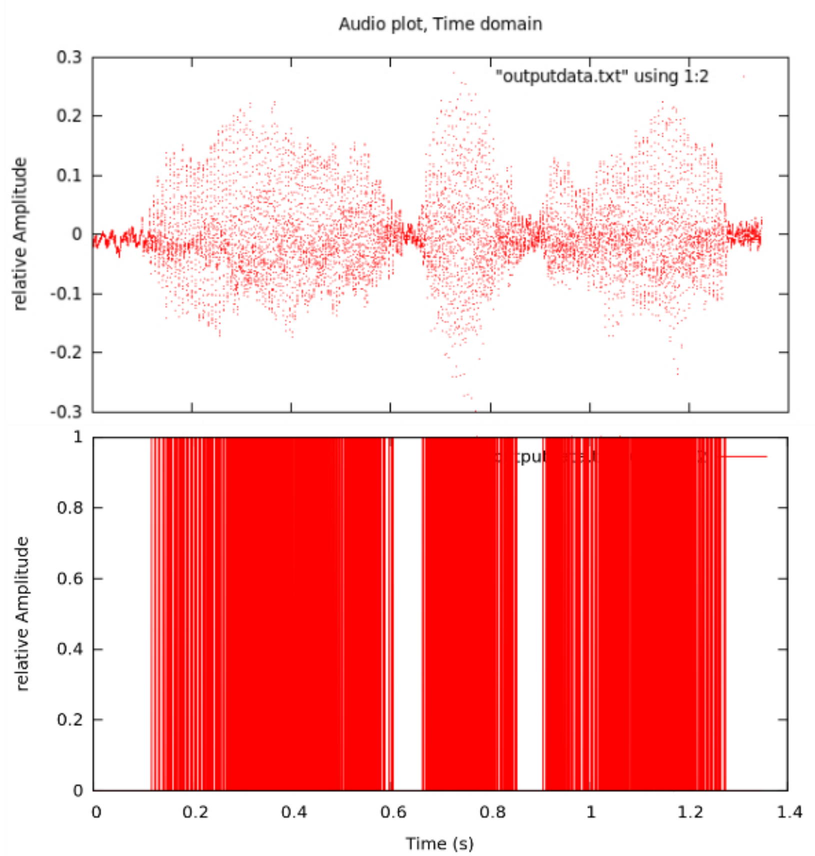
4. Increase window size geometrically and repeat from 1

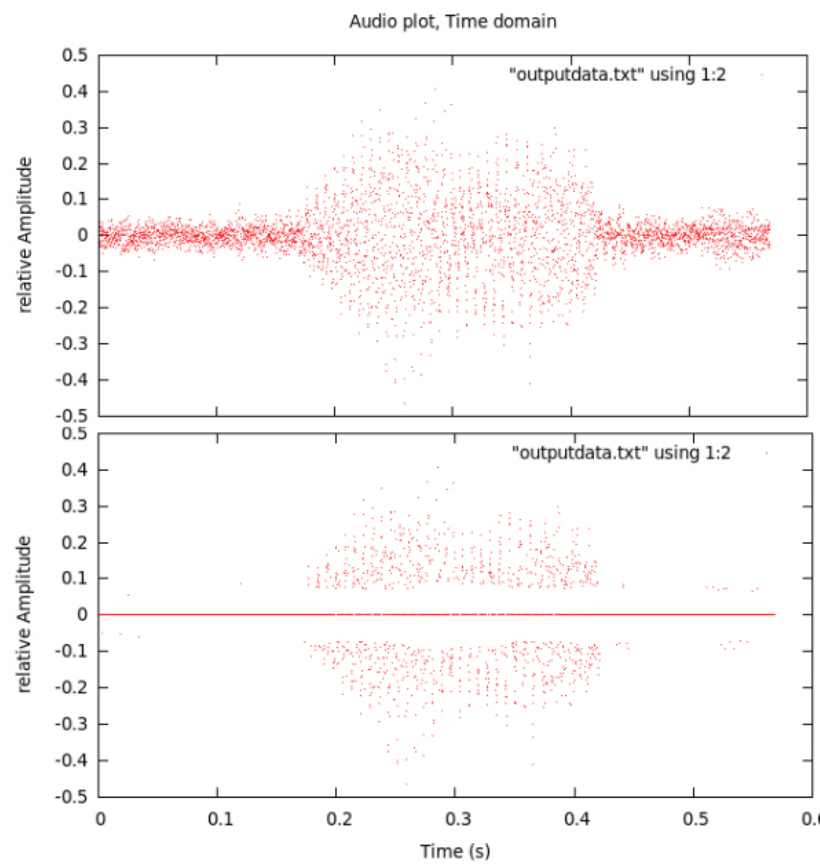
5. Select the windows size that maximize distance between centroids

6. determine the class membership

$$m(n) = \left[\frac{s(n) - c_0(n, \mu^*(n), N^*(n))}{c_1(n, \mu^*(n), N^*(n)) - c_0(n, \mu^*(n), N^*(n))} > 0.5 \right]$$

Examples





Prezi

SHUKRANI