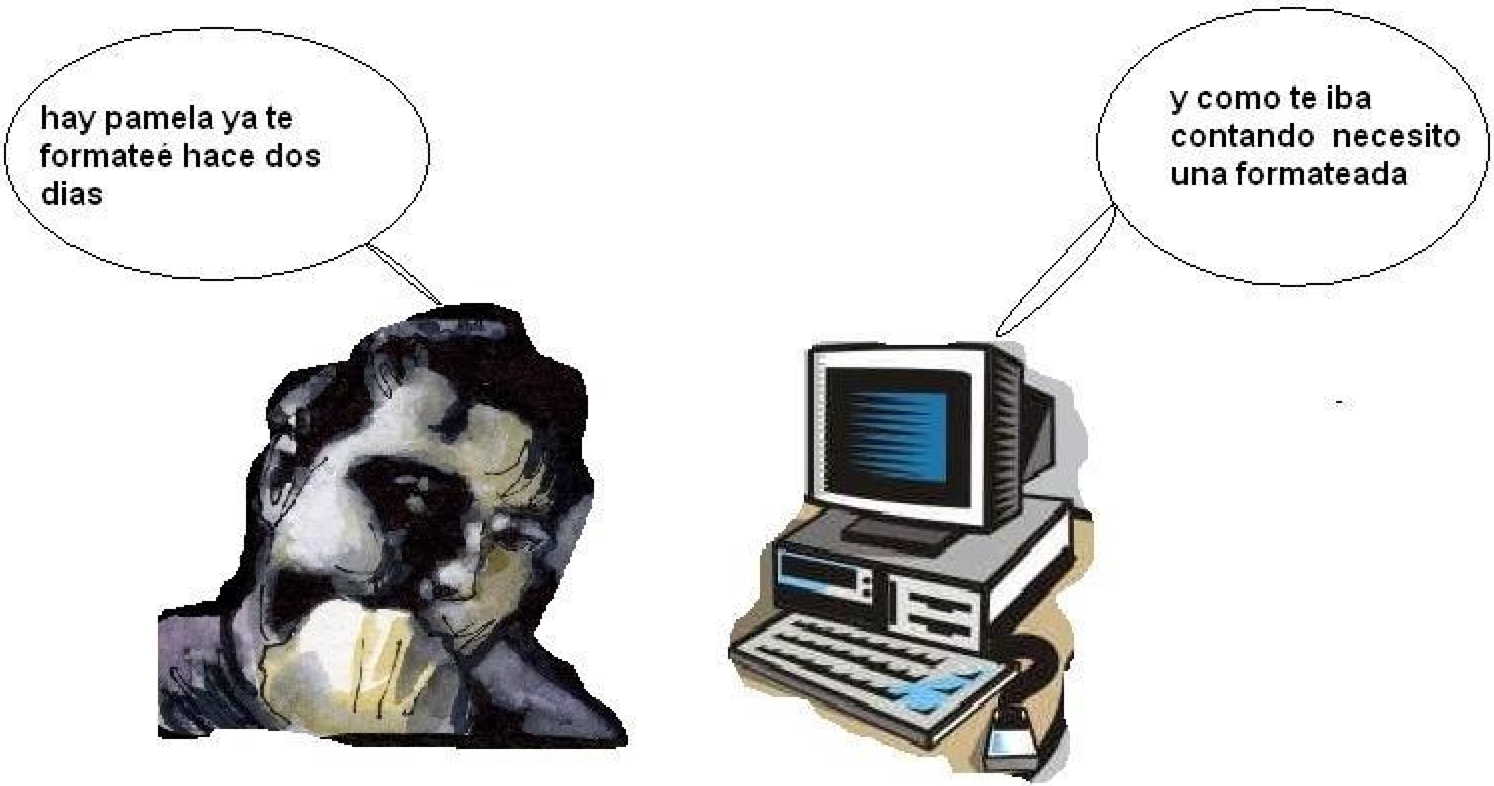
Coeficientes Cepstrales en Frecuencia Mel y Dynamic Time Warping para Reconocimiento Automatico del Habla

Jorge Luis Guevara Diaz

Semana ciencia de la Computación
Escuela de Informática
Universidad Nacional de Trujillo-Perú

Podriamos conversar con las maquinas como lo hacemos con los humanos?



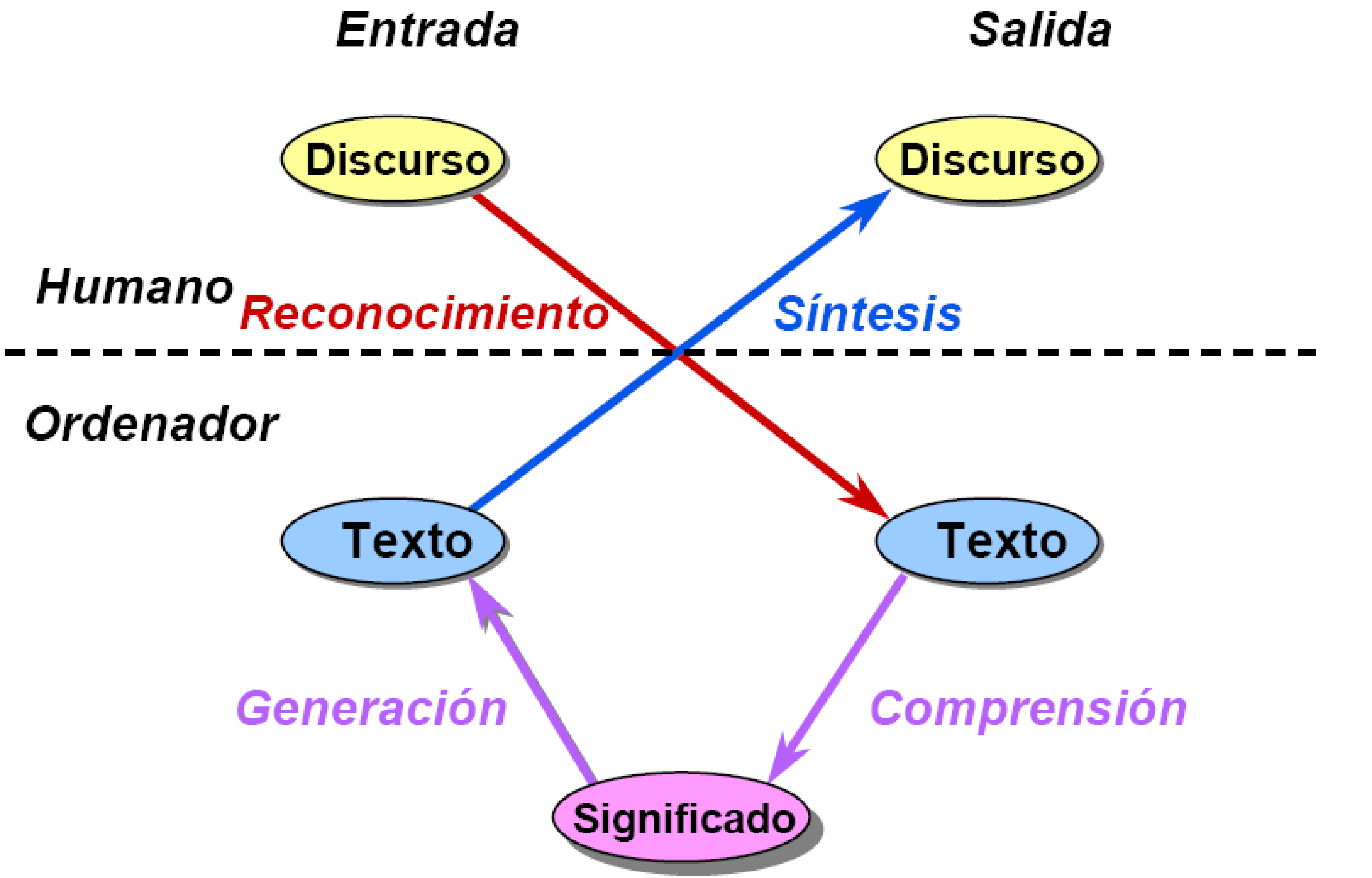
Sistemas informáticos del lenguaje hablado

1. Modulo de reconocimiento del hablante (speaker recognition)

2. Modulo de reconocimiento reconocimiento automatico del habla (speech recognition)

3 Modulo de entendimiento del lenguaje
Spoken Language Understanding

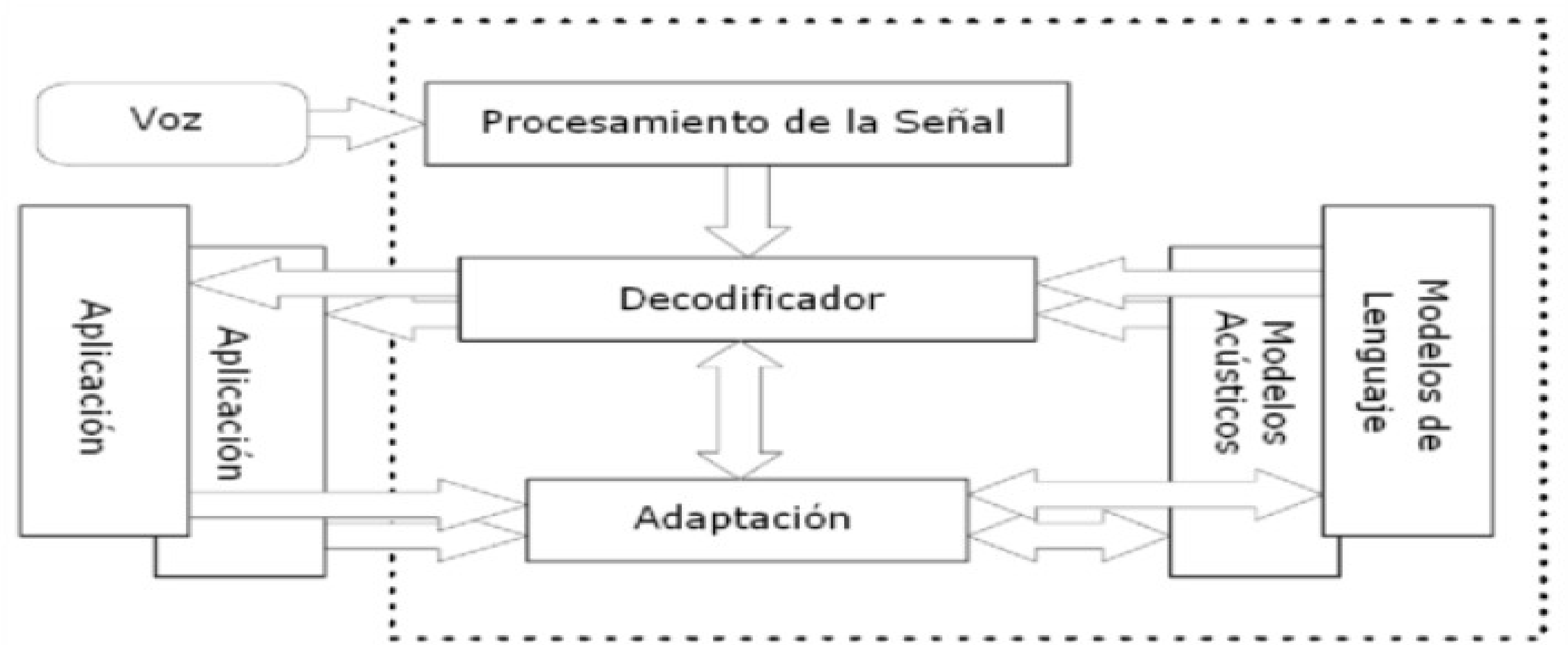
4 Modulo de sintesis
(Text-to-Speech Conversion)



Entrada
Salida
Discurso
Discurso
Humano
Reconocimiento
Síntesis
Ordenador
Texto
Texto
Generación
Comprensión
Significado

Modulo de reconocimiento

- Reconocimiento Automatico del Habla (Speech recognition)



Que necesito?

- Hacer un procesamiento de la señal de voz en la computadora , con algoritmos óptimos menor complejidad computacional

Procesamiento digital de señales

Algoritmos, matematicas, tecnicas para señales digitalizadas

Machine learnig

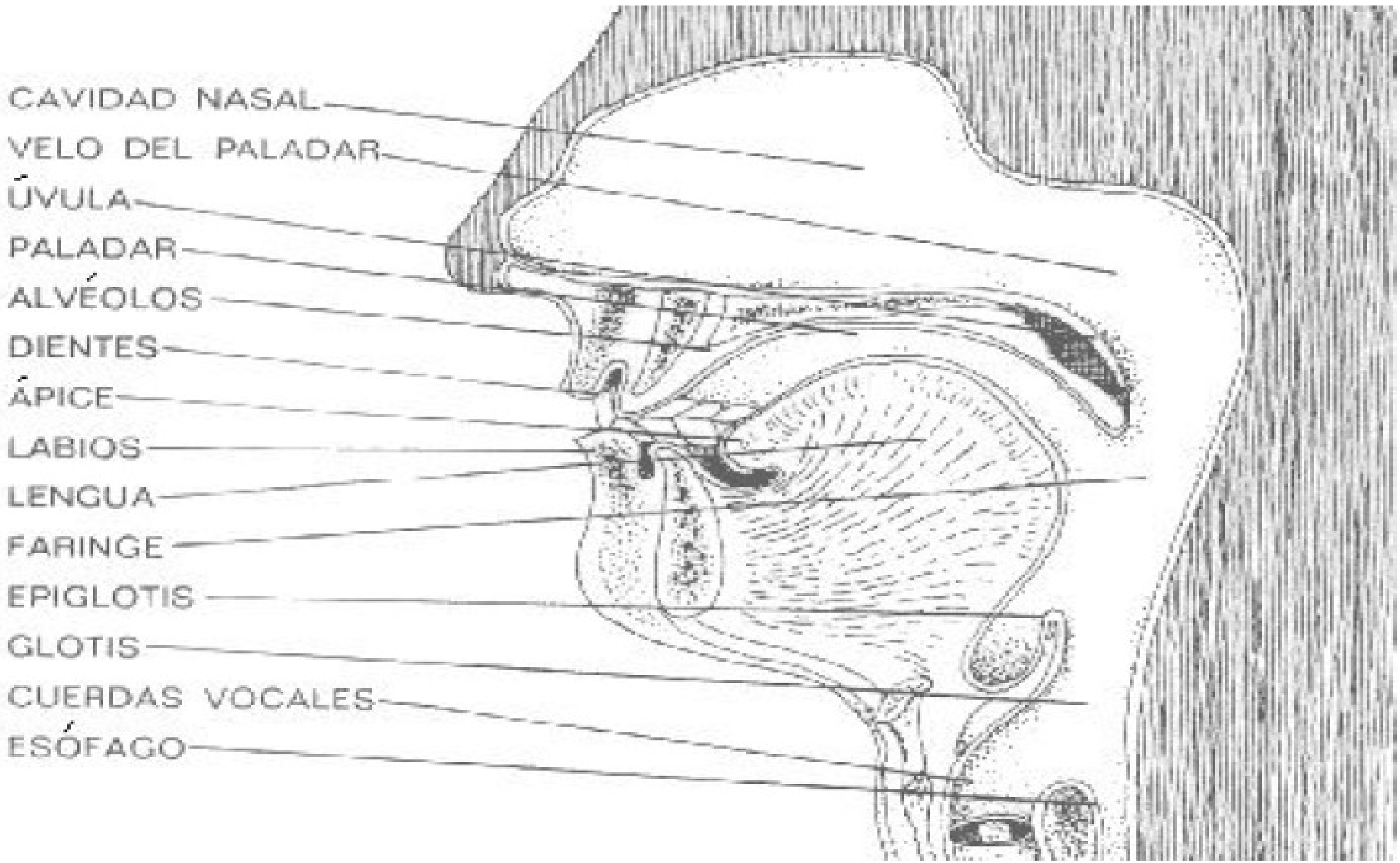
como hacer que las maquinas aprendan

Empecemos!!!

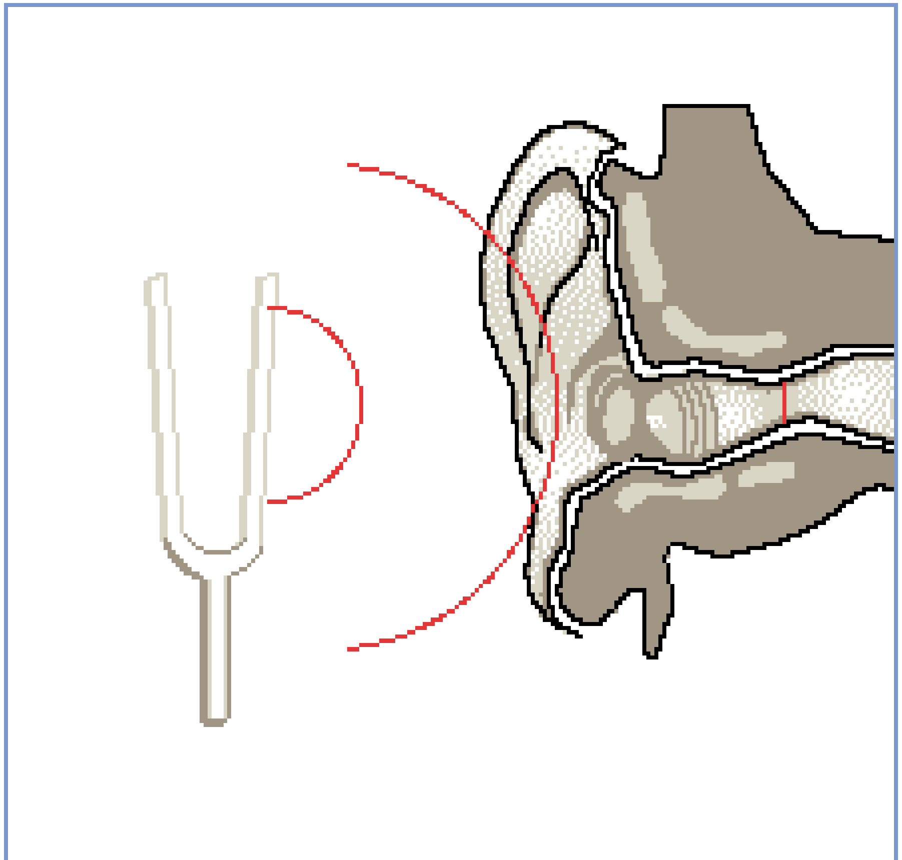
- Idea :

  Analizar el modelo biologico para poder construir el modelo computacional

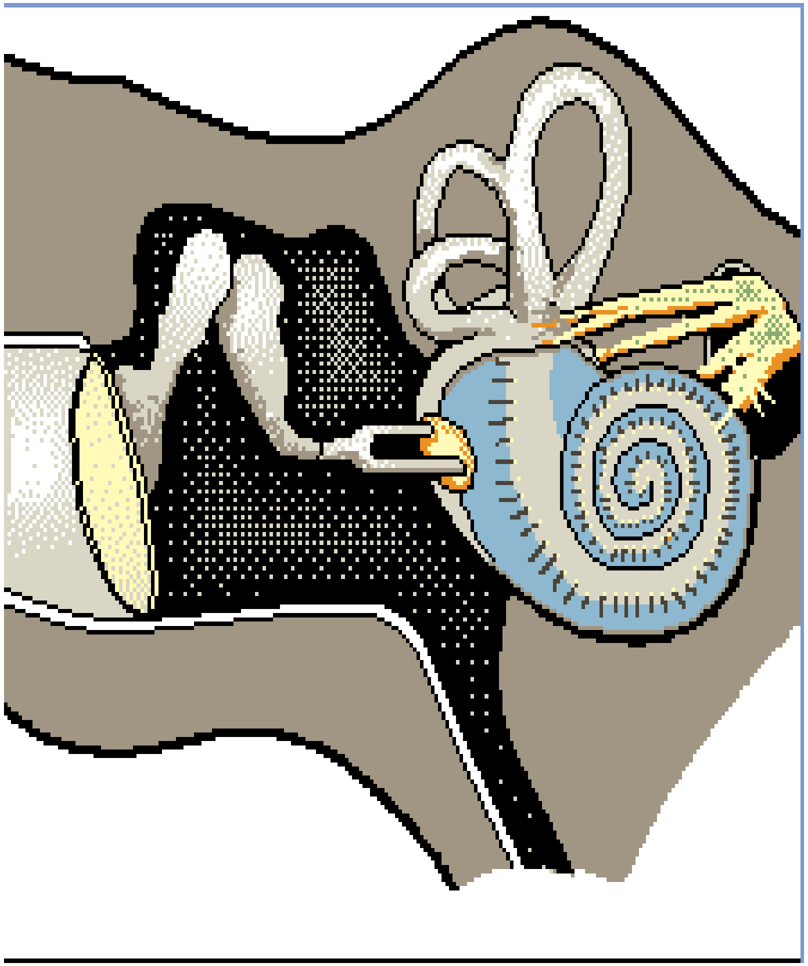
Produccion del habla



percepción del habla



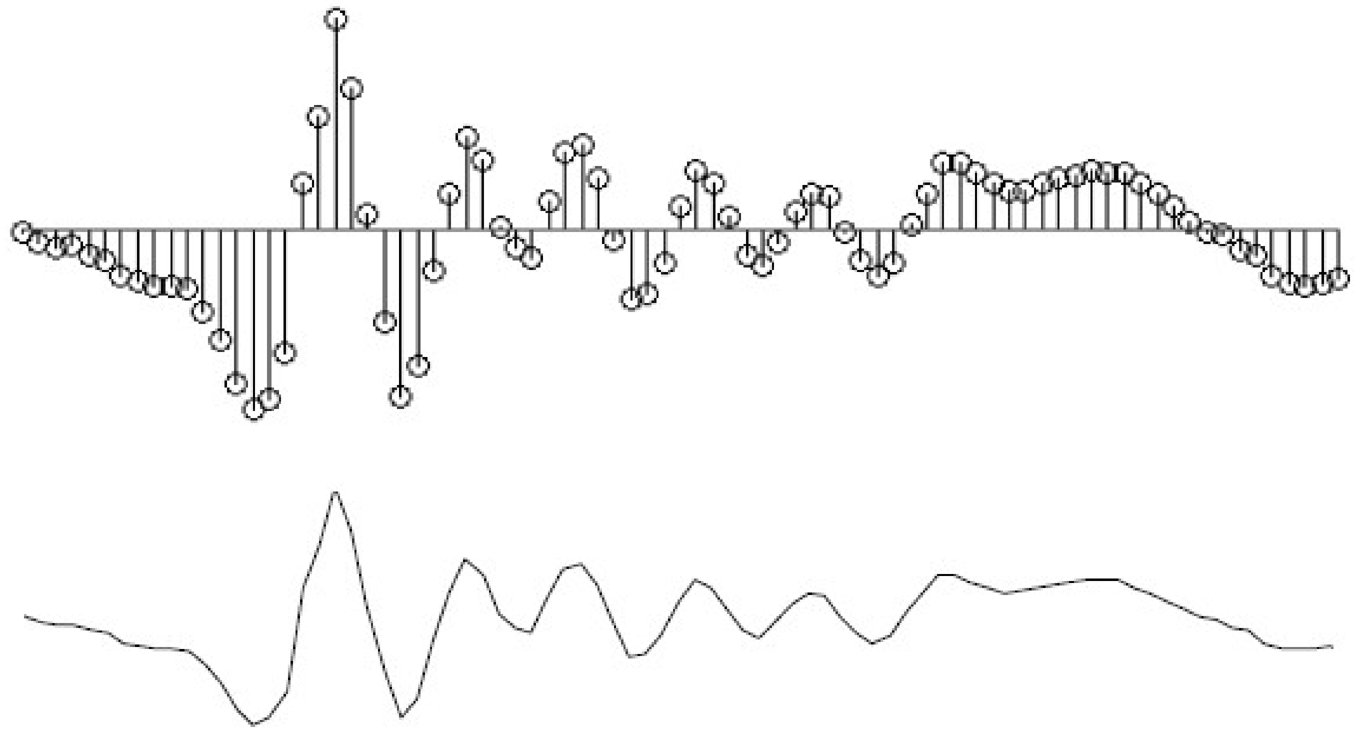
Los sonidos entran en el oído



Empecemos!!!

- Capturar la señal analogica y digitalizarla para poder usarla en la computadora

$$x[n] = x_0(nT).$$



Problema: cuantas muestras debo tomar?

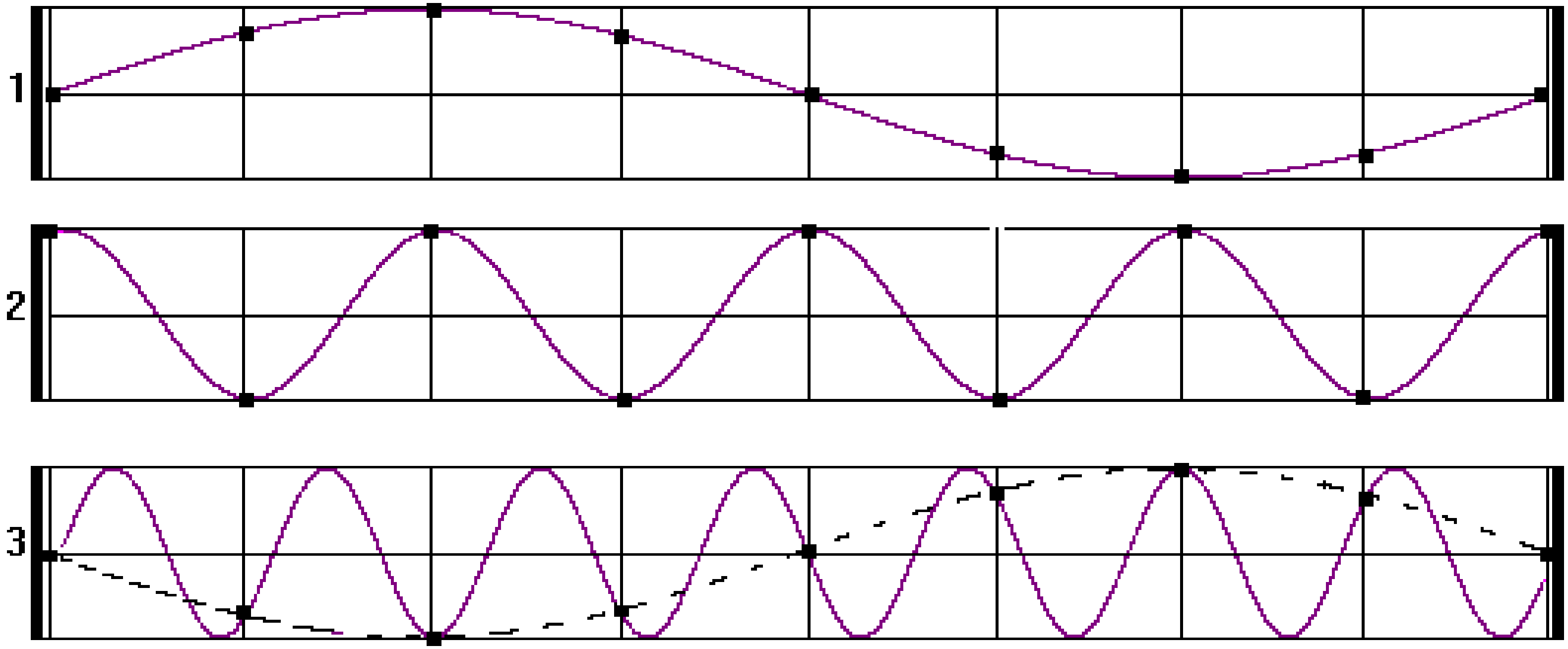
Frecuencia de muestreo $F_s = \frac{1}{T}.$

Generalmente Fs > 8000 muestras para poder obtener buenos resultados

Teorema del muestreo

Fs=2*Fa POR QUE????

Fa=más alta frecuencia de la señal

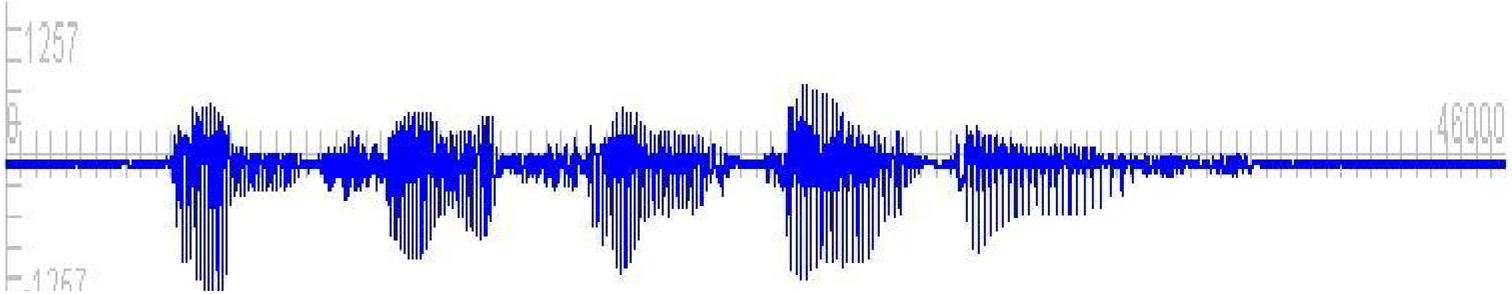


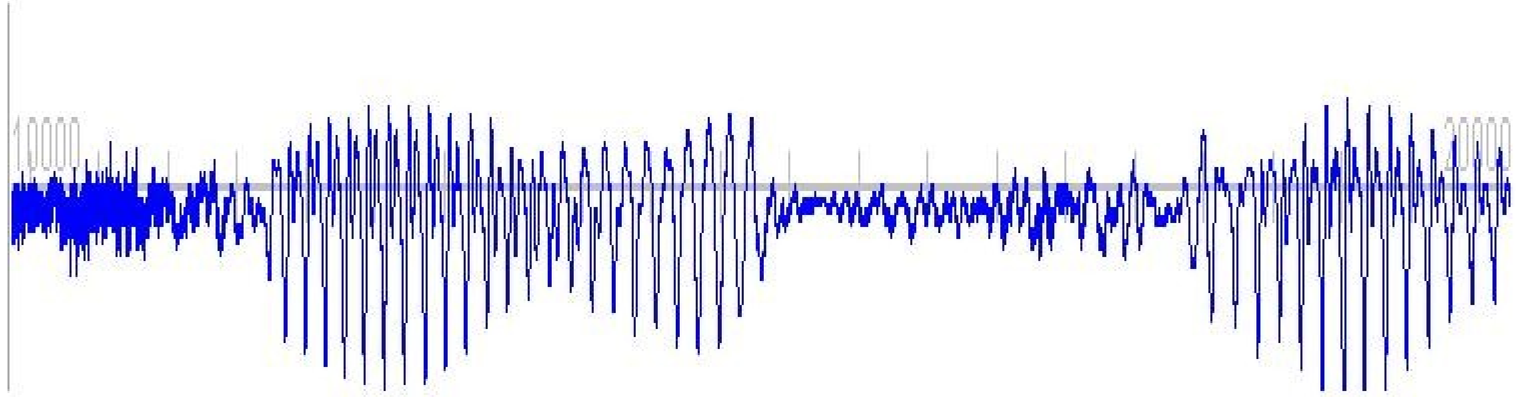
Idea

Eliminar componentes de alta frecuencia > Fs/2 !!!

Mas problemas!!!

- Vectores de 16000 elementos por segundo
- Como identifico las frecuencias ?

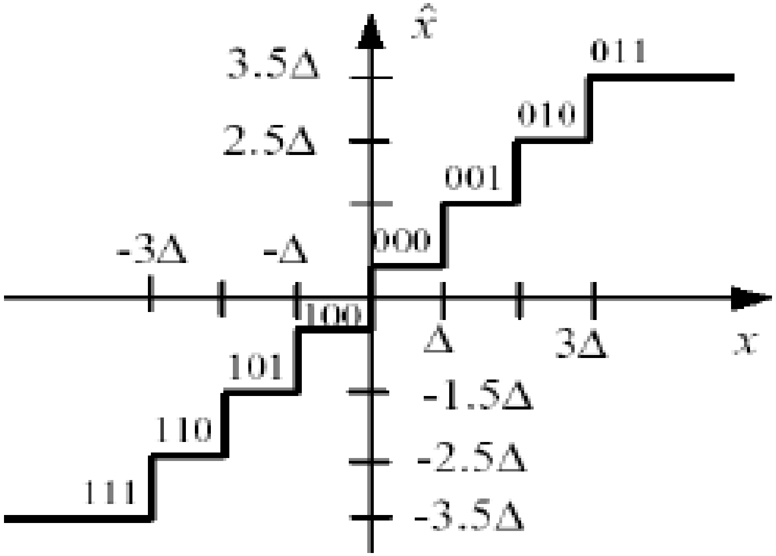


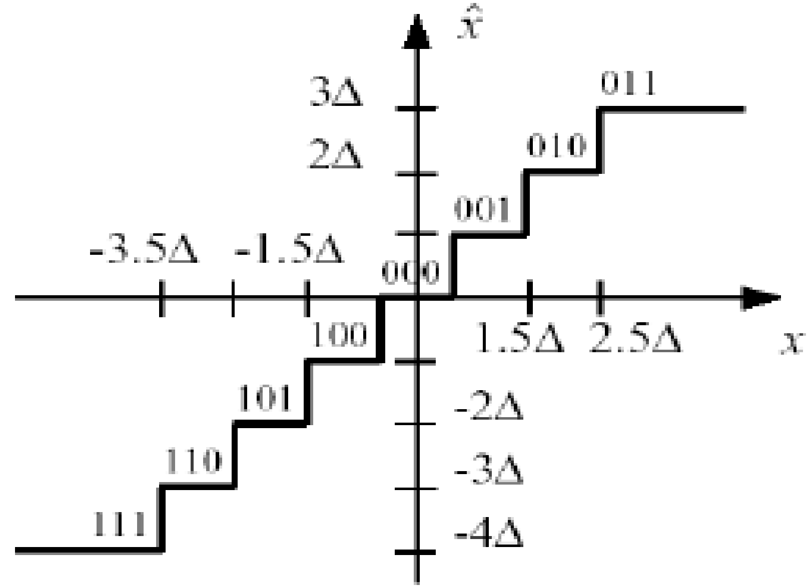


quantificación

- PCM Con B bits es posible representar 2^B niveles

$$\hat{x}[n] = Q\{x[n]\}$$





Procesamiento digital de la señal

- Algo de teoria: $y[n] = T\{x[n]\}$

  Sistemas lineales e invariantes en el tiempo:

Serán lineales si :

$$T\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 T\{x_1[n]\} + a_2 T\{x_2[n]\}$$

y serán invariantes en el tiempo si :

$$y[n - n_0] = T\{x[n - n_0]\}$$

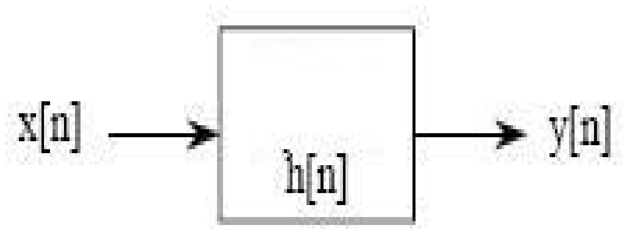
$$x[n] \longrightarrow \boxed{T\{\}} \longrightarrow y[n]$$

Procesamiento digital de la señal

- Convolucion

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

$$y[n] = x[n] * h[n]$$



Procesamiento digital de señales

- Analizar la señal en el dominio de la frecuencia : transformada de fourier

Transformada de fourier

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$$

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega$$

Donde :

$$e^{j\phi} = \cos\phi + j\sin\phi$$

Algo importante

Si $x[n] = e^{j\omega n}$

$$y[n] = \sum_{k=-\infty}^{\infty} e^{j\omega(n-k)} h[k] = e^{j\omega n} \sum_{k=-\infty}^{\infty} e^{-j\omega k} h[k] = H(e^{j\omega}) e^{j\omega n}$$

Si descomponemos $x[n] = \int X(e^{j\omega}) e^{-j\omega n} d\omega$

$$y[n] = \int H(e^{j\omega}) X(e^{j\omega}) e^{-j\omega n} d\omega$$

$$y[n] = x[n] * h[n]$$

Convolucion en el dominio del tiempo es
igual a una multiplicación en el dominio de
La frecuencia

$$Y(e^{j\omega}) = X(e^{j\omega}) H(e^{j\omega})$$

Transformada discreta de fourier

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi nk}{N}}$$

Complejidad computacional : O(n^2)

Intratable para aplicaciones con datos con mas de 8000 Muestras por segundo

Idea : utilizar divide y conquista!!!

Transformada rapida de fourier

- Existen varios algoritmos que implementan la transformada rapida de fourier en solo O(n log n)!!!
- En esta investigacion :
- Algoritmo radix-2 con diezmado en frecuencia y reordenamiento de la salida de bits mezclados.....

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{\frac{-j2\pi nk}{N}}=\sum_{n=0}^{N-1}x[n]W_N^{kn}, k=0,1,...,N-1$$

$$W_N=e^{\frac{-j2\pi}{N}}$$

$$X_k=\sum_{n=0}^{\frac{N}{2}-1}x_nW_N^{kn}+\sum_{n=\frac{N}{2}}^{N-1}x_nW_N^{kn}$$

$$X_k=\sum_{n=0}^{\frac{N}{2}-1}x_nW_N^{kn}+\sum_{n=0}^{\frac{N}{2}-1}x_{n+\frac{N}{2}}W_N^{k(n+\frac{N}{2})}$$

$$X_k=\sum_{n=0}^{\frac{N}{2}-1}(x_n+x_{n+\frac{N}{2}}W_N^{k\frac{N}{2}})W_N^{kn}, k=0,1,...,N-1$$

escogiendo los indices pares tenemos:

$$X_{2k}=\sum_{n=0}^{\frac{N}{2}-1}(x_n+x_{n+\frac{N}{2}}W_N^{kN})W_N^{2kn}$$

$$X_{2k}=\sum_{n=0}^{\frac{N}{2}-1}(x_n+x_{n+\frac{N}{2}})W_{\frac{N}{2}}^{kn}, k=0,1,...,\frac{N}{2}-1$$

definiendo $Y_k = X_{2k}$ y $y_n = x_n + x_{n+\frac{N}{2}}$ tenemos la primera mitad del problema

$$Y_k = \sum_{n=0}^{\frac{N}{2}-1} y_n W_{\frac{N}{2}}^{kn}$$

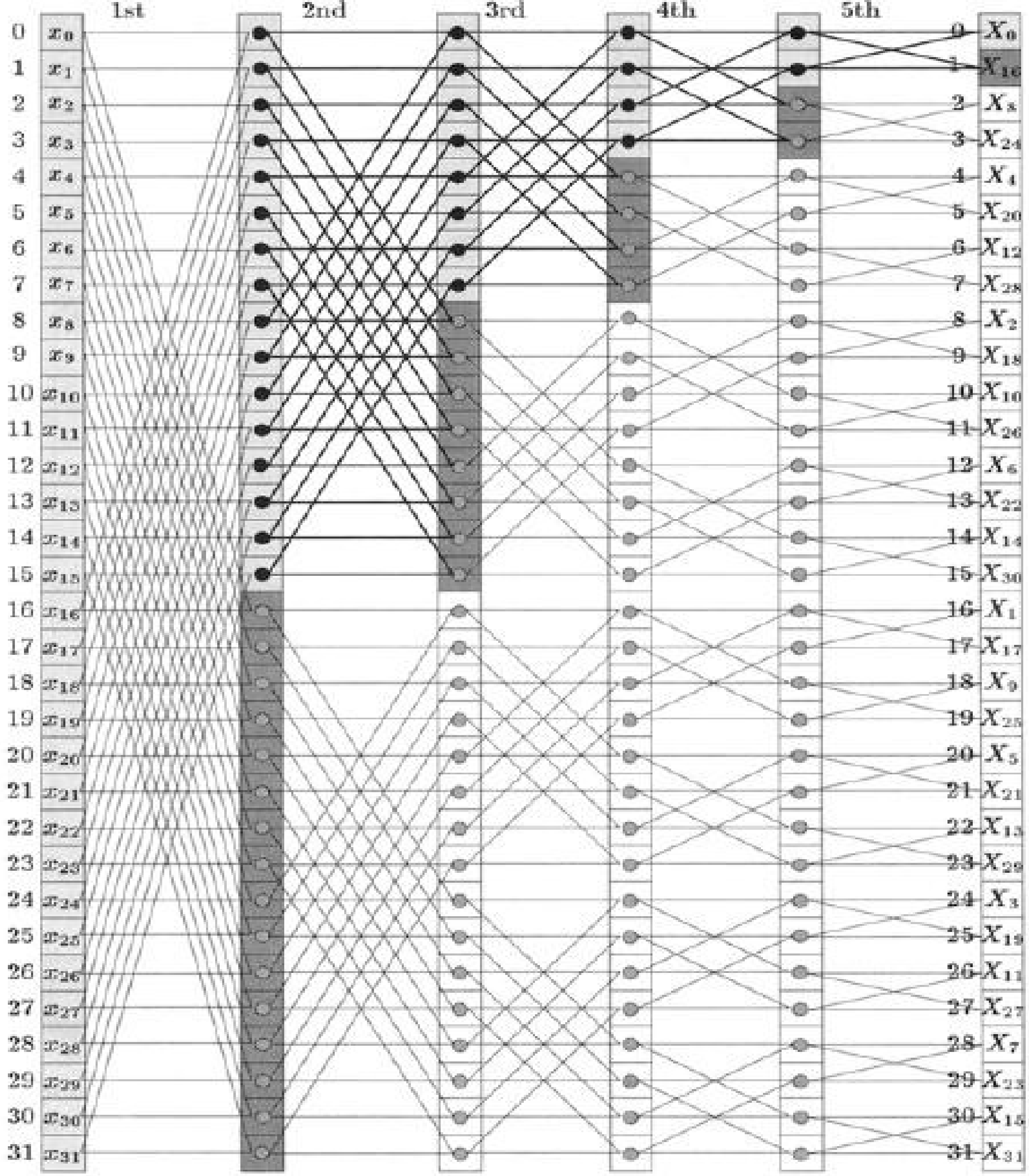
similarmente para los indices impares tenemos:

$$X_{2k+1} = \sum_{n=0}^{\frac{N}{2}-1} (x_n + x_{n+\frac{N}{2}} W_N^{(2k+1)\frac{N}{2}}) W_N^{(2k+1)n}$$

$$X_{2k+1} = \sum_{n=0}^{\frac{N}{2}-1} ((x_n - x_{n+\frac{N}{2}}) W_N^n) W_{\frac{N}{2}}^{kl}, k = 0, 1, ..., \frac{N}{2} - 1$$

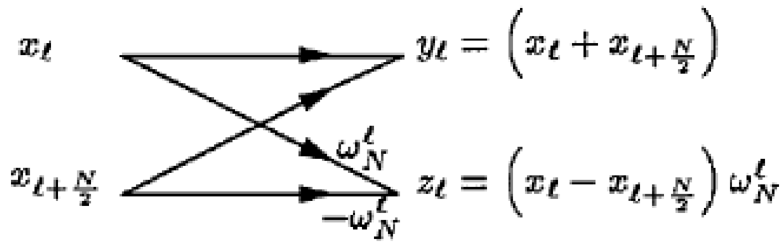
definiendo $Z_k = X_{2k+1}$ y $z_n = (x_n - x_{n+\frac{N}{2}}) W_N^n, k = 0, 1, ..., \frac{N}{2} - 1$, obtenemos la segunda mitad del problema

$$Z_k = \sum_{n=0}^{\frac{N}{2}-1} z_n W_{\frac{N}{2}}^{kn}, k = 0, 1, ..., \frac{N}{2} - 1$$



1st
2nd
3rd
4th
5th
0 x_0
1 x_1
2 x_2
3 x_3
4 x_4
5 x_5
6 x_6
7 x_7
8 x_8
9 x_9
10 x_{10}
11 x_{11}
12 x_{12}
13 x_{13}
14 x_{14}
15 x_{15}
16 x_{16}
17 x_{17}
18 x_{18}
19 x_{19}
20 x_{20}
21 x_{21}
22 x_{22}
23 x_{23}
24 x_{24}
25 x_{25}
26 x_{26}
27 x_{27}
28 x_{28}
29 x_{29}
30 x_{30}
31 x_{31}
0 X_0
1 X_{16}
2 X_8
3 X_{24}
4 X_4
5 X_{20}
6 X_{12}
7 X_{28}
8 X_2
9 X_{18}
10 X_{10}
11 X_{26}
12 X_6
13 X_{22}
14 X_{14}
15 X_{30}
16 X_1
17 X_{17}
18 X_9
19 X_{25}
20 X_5
21 X_{21}
22 X_{13}
23 X_{29}
24 X_3
25 X_{19}
26 X_{11}
27 X_{27}
28 X_7
29 X_{23}
30 X_{15}
31 X_{31}

La mariposa Gentleman-Sande



Transformada Rápida de Fourier : reordenamiento de bits mezclados

Entrada :	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
	$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$
Salida :	X_0	X_4	X_2	X_6	X_1	X_5	X_3	X_7
	$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$

Complejidad computacional

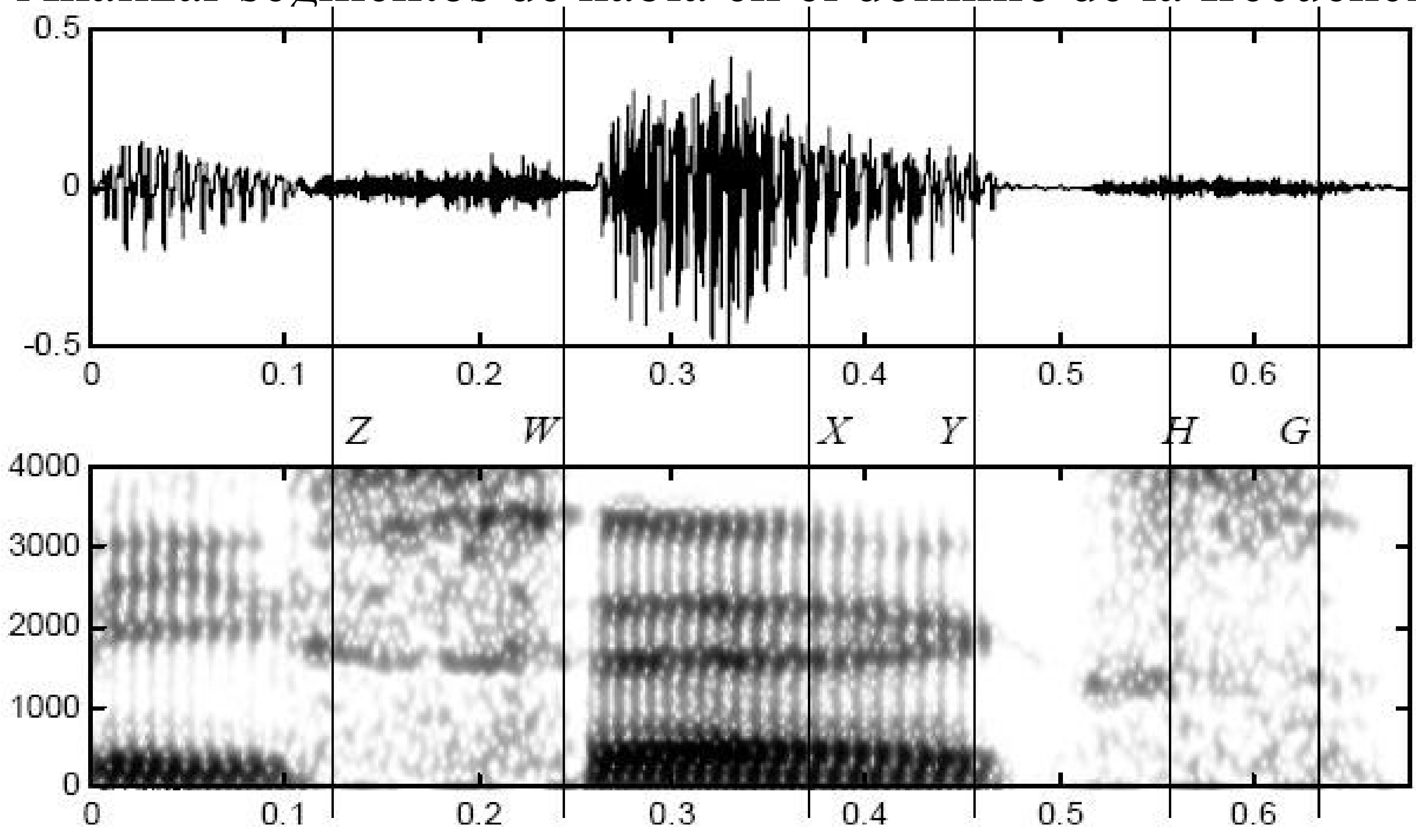
$$T(N) = \begin{cases} 2T\left(\frac{N}{2}\right) + \mathsf{C}N & \text{if } N = 2^n \geq 2\,, \\ 0 & \text{if } N = 1. \end{cases}$$

Resolviendo la ecuación de recurrencia se tiene:

$$T(N) = \mathsf{C}N \log_2 N$$

Volvamos !!!

Analizar segmentos de habla en el dominio de la frecuencia



Extraccion de características

- Existen varias tecnicas entre las mas usadas tenemos :
- Coeficientes cepstrales
- Predicción lineal
- Coeficientes cepstrales en escala Mel
- Predicción lineal perceptual

Coeficientes cepstrales en escala Mel

- Unidad minima :

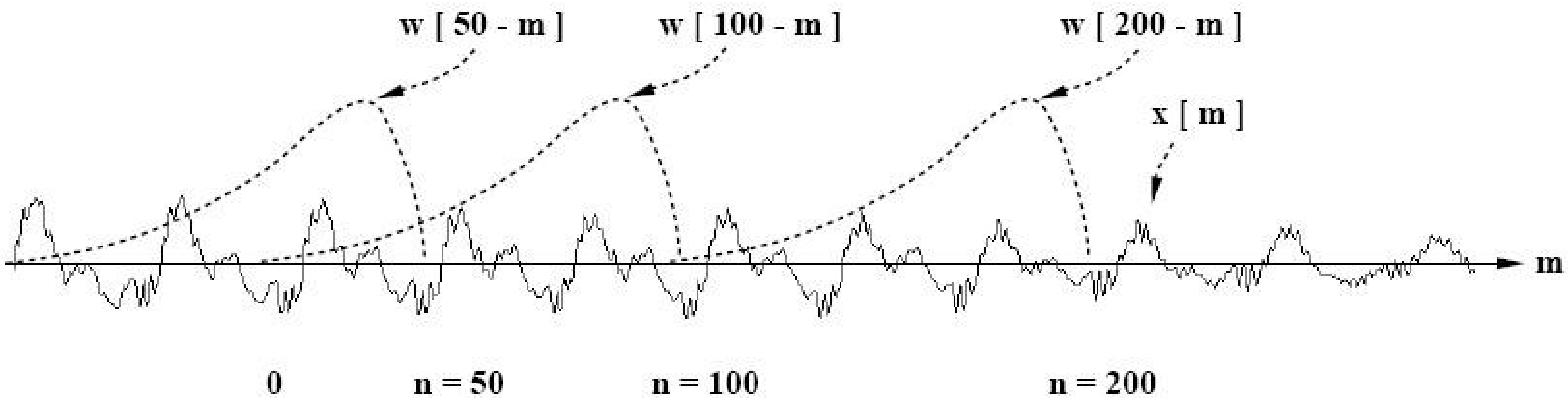
  el frame = $x^m[n]=x[n-mF]w[n]$

  x=señal de voz

  F= tamaño de paso   aprox [10mS,20mS]

  w= ventana   aprox[20mS, 25mS]

Transformada corta de fourier



$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{+\infty} w[n-m]x[m]e^{-j\omega m}$$

Que tipo de ventana utilizar?

- Tipos de ventana:

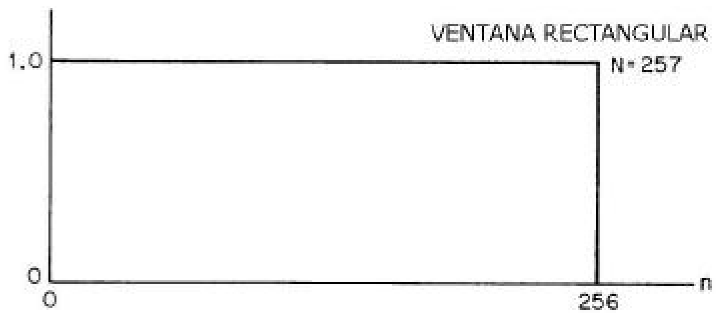
  Rectangular

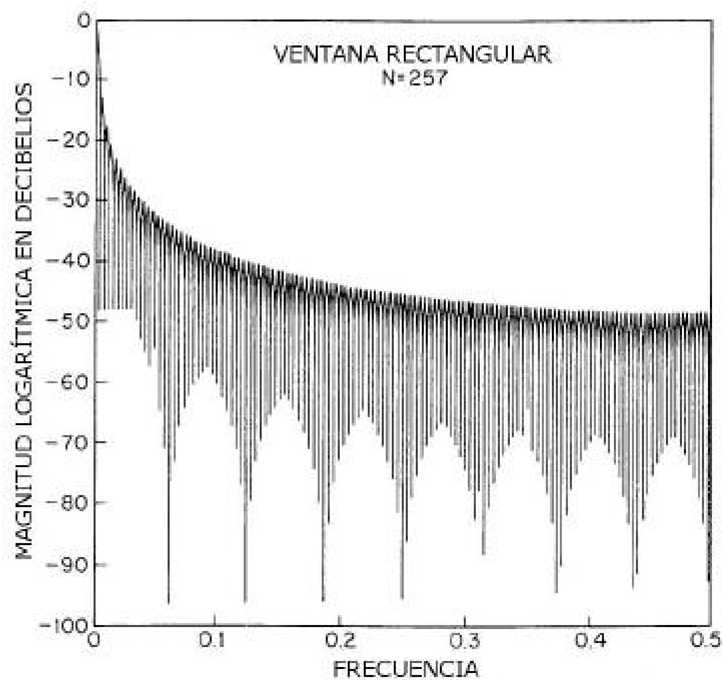
  Hamming

  Cual es mejor? Analicemos....

Ventana Rectangular

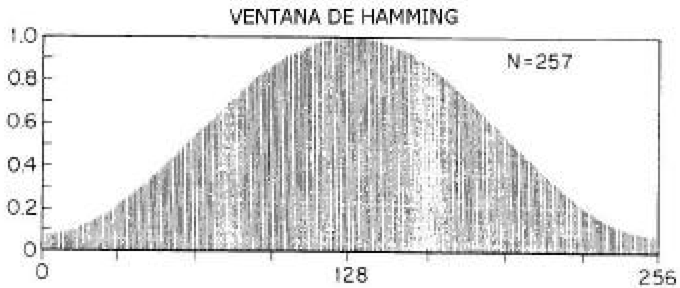
$$w[n] = 1, \qquad 0 \le n \le N-1$$

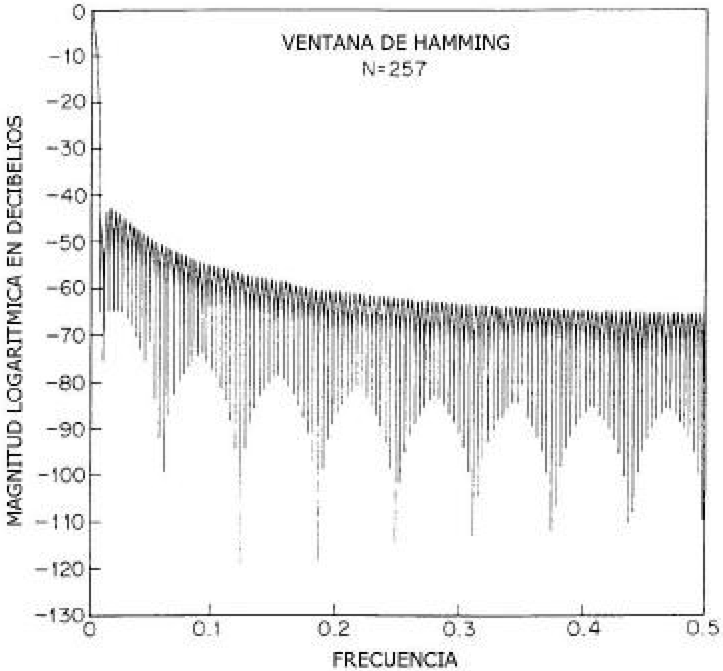




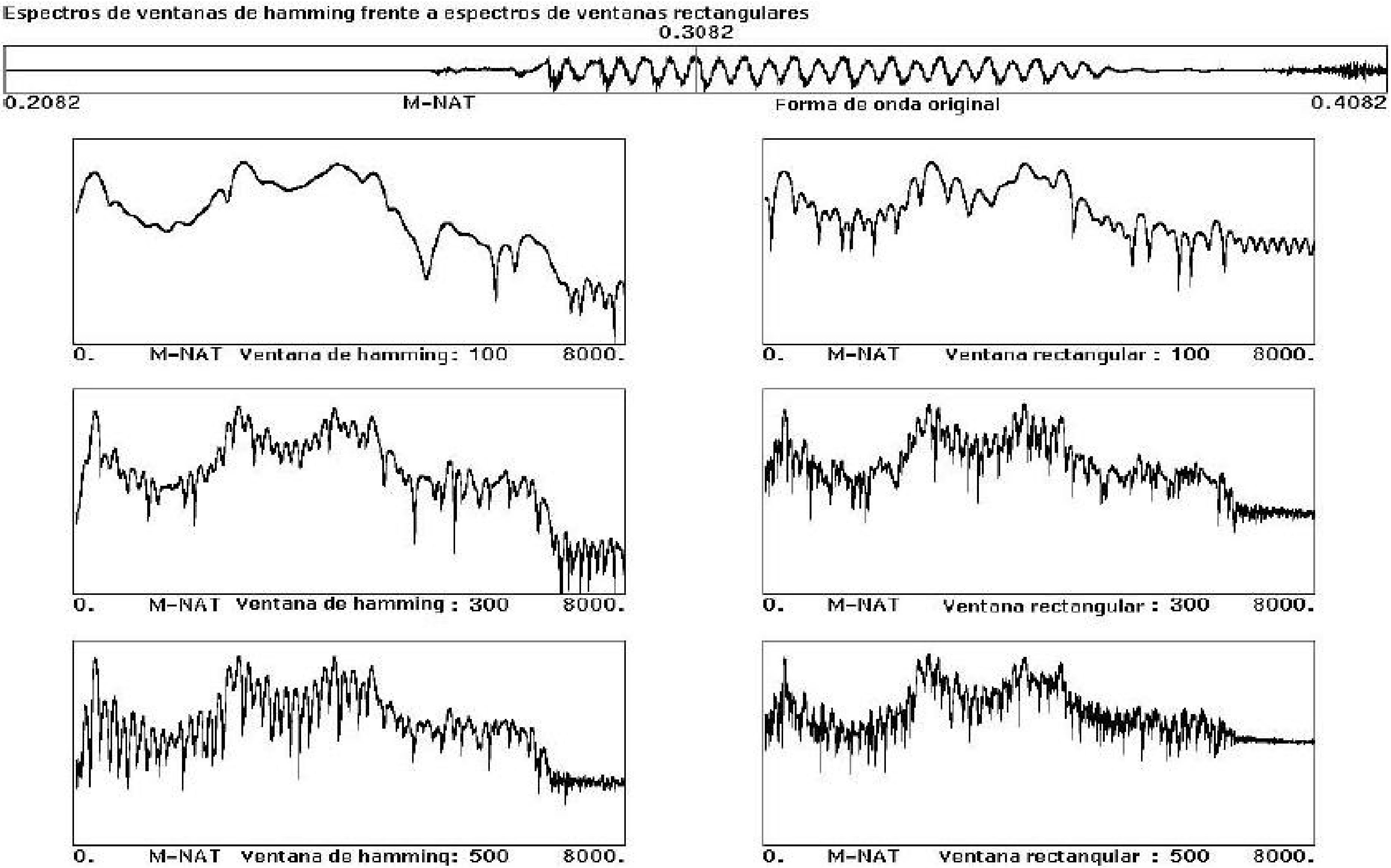
Ventana Hamming

$$w[n] = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \le n \le N-1$$



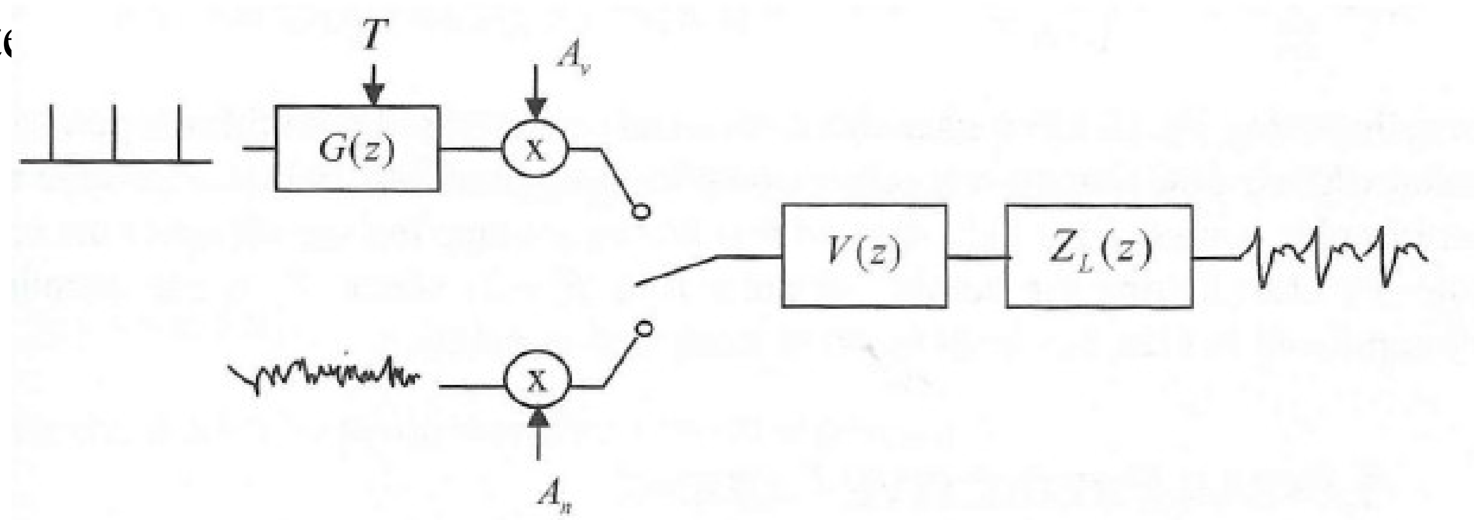


Comparacion de ventanas



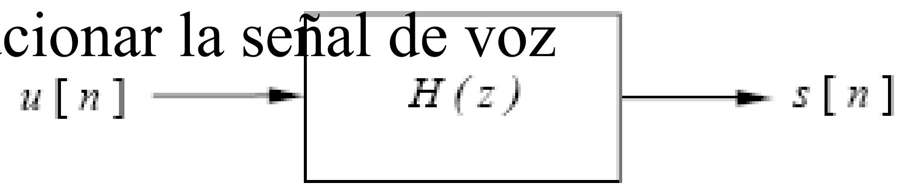
Cepstrum

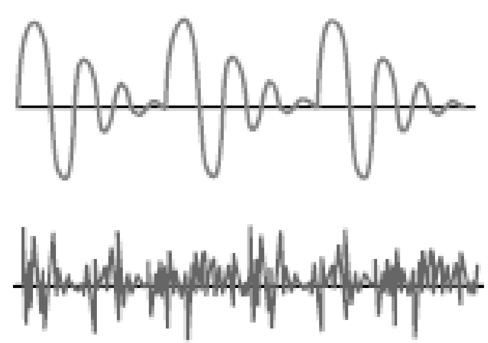
Si imaginamos la señal de voz como producto de la conmvolucion
del aire que fluye de nuestros pulmones y varios filtros correspondient



objetivo

Desconvolucionar la señal de voz





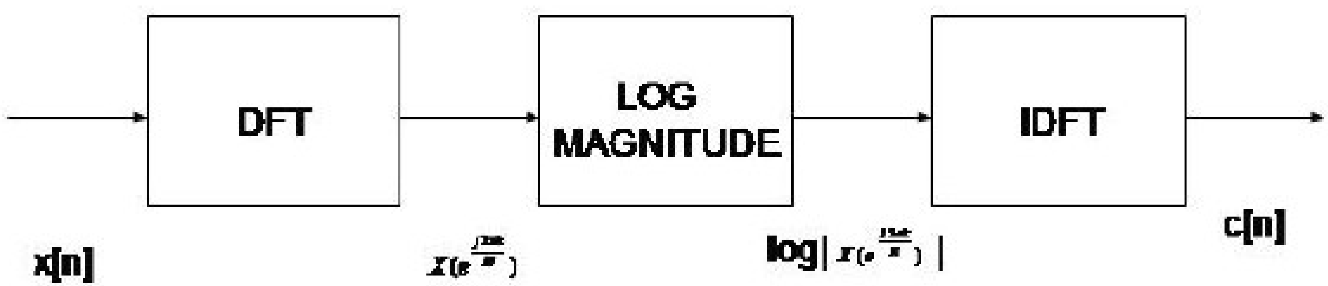
Cepstrum

$$x[n] = e[n] * h[n]$$

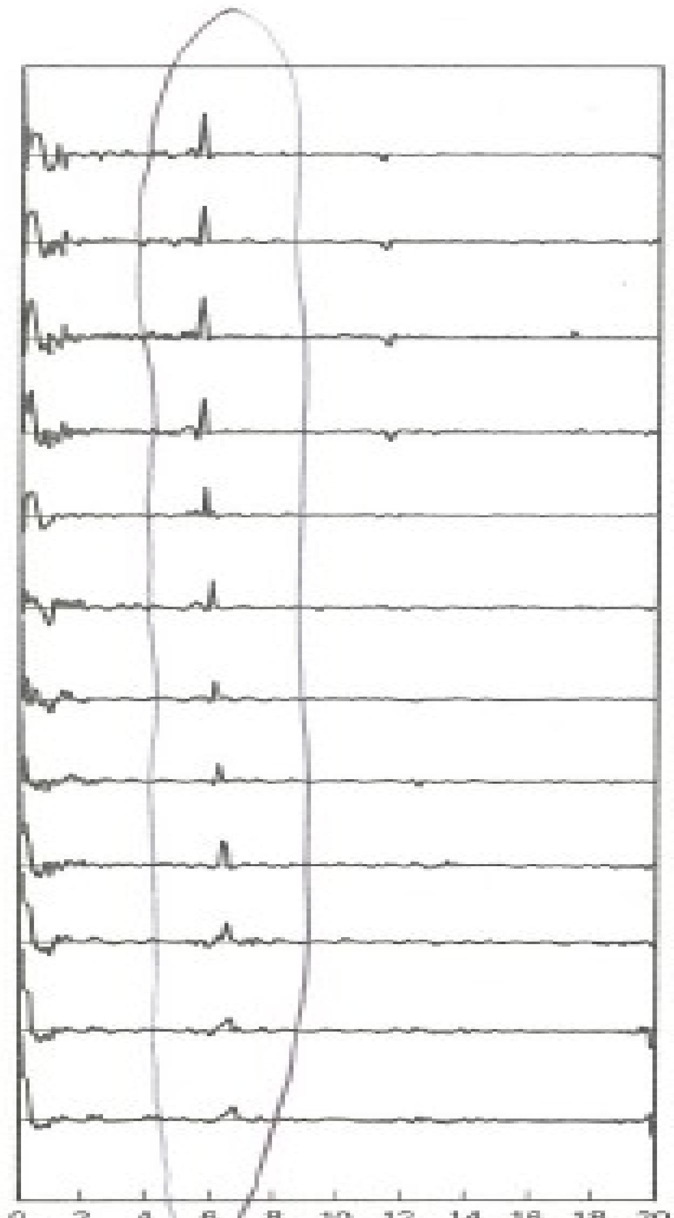
$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n]$$

El cepstrum $D[]$ de una señal digital $x[n]$ esta definido :

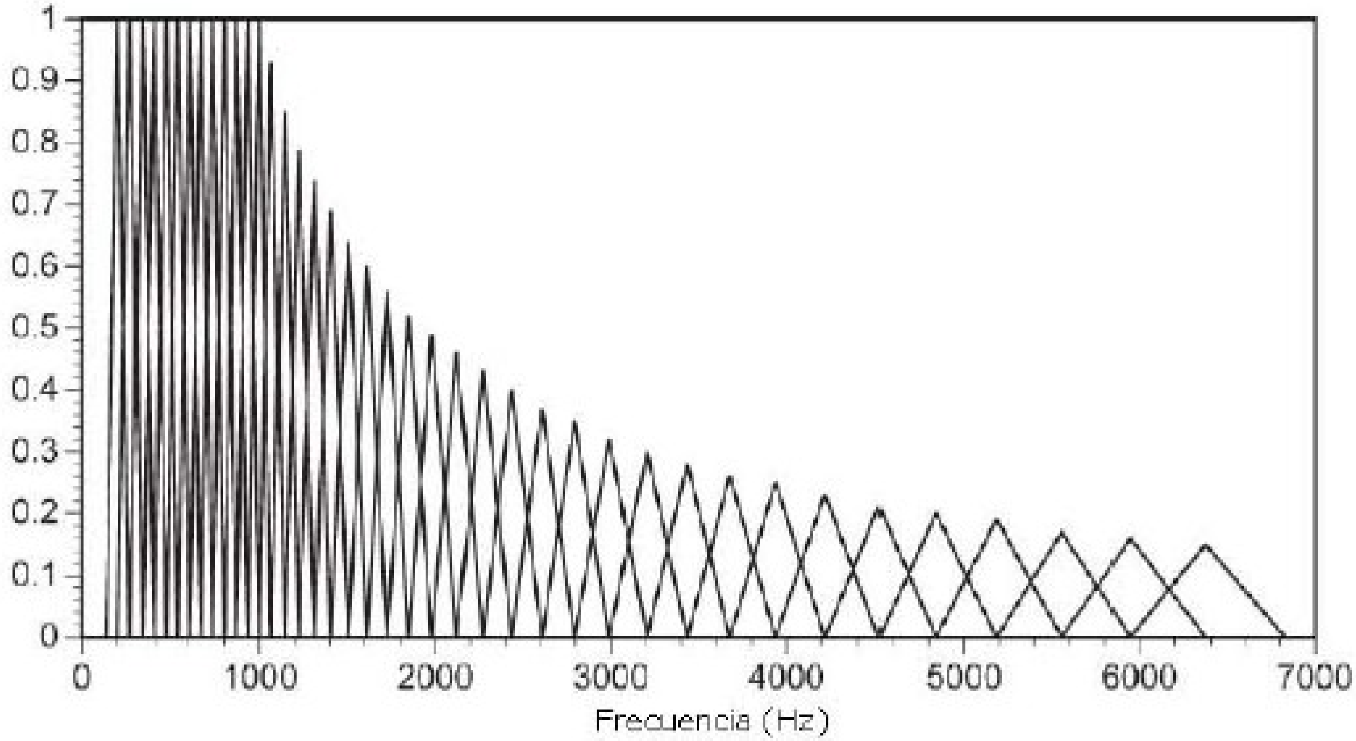
$$\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(e^{jw})| e^{jwn}$$



Cepstrum

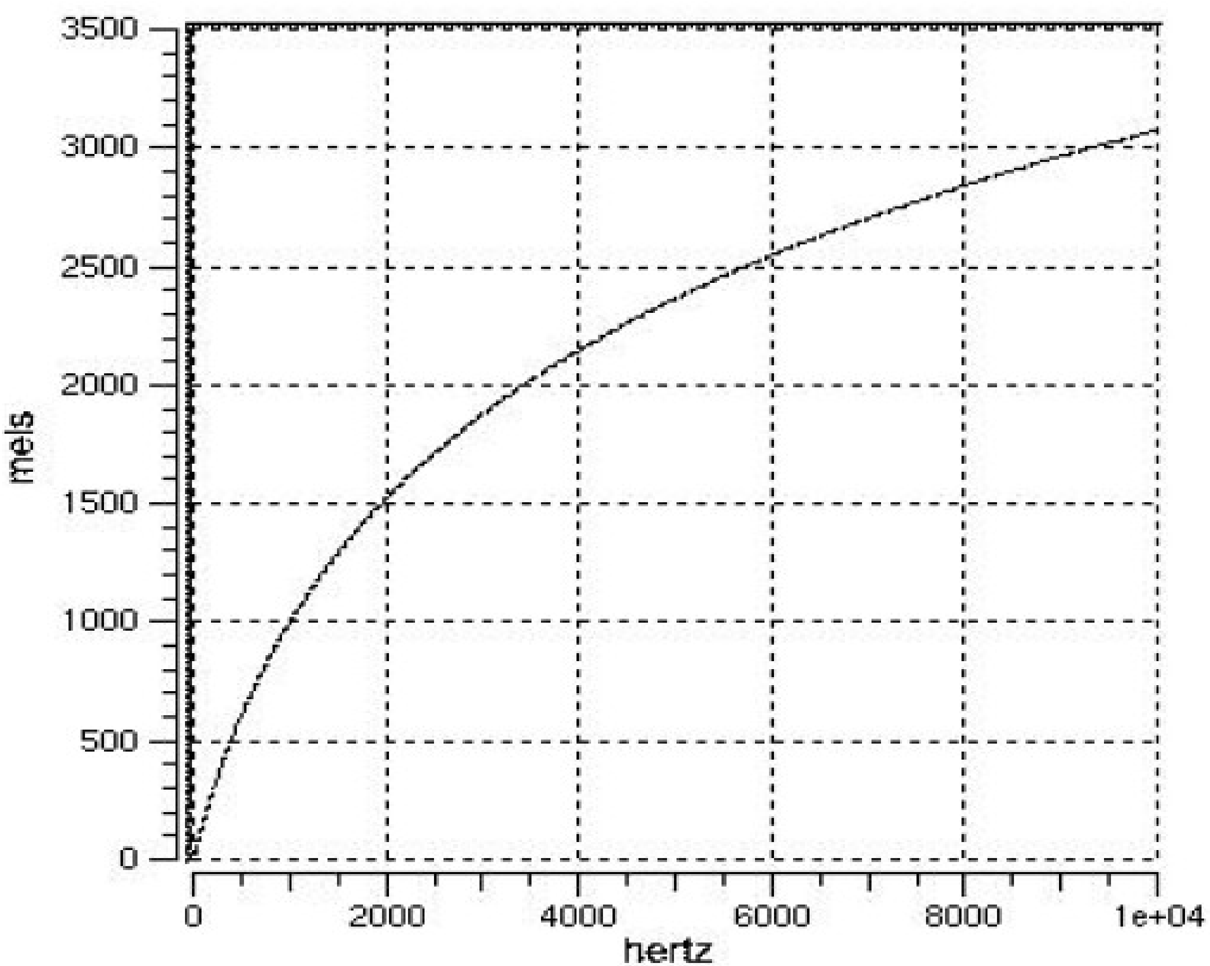


Frecuencia Mel



Frecuencia Mel

- Es una escala basada en como oímos, y se ha construido , a través de experimentos fisiológicos



Frecuencia Mel : bins

$$H_m(k) \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

$$f(m) = \frac{N}{F_S} B^{-1}(B(f_l) + m\frac{B(f_h) - B(f_l)}{m+1})$$

$$B(f) = 2595 \log_{10}(1 + f/700)$$

$$S(m) = 20\, log_{10}(\sum_{k=0}^{N-1} |X(k)| H_m(k)), 0 < m < M$$

Coeficientes Cepstrales en Frecuencia Mel

$$c[n] = \sum_{m=0}^{M-1} S(m) \cos(\pi n (m - 1/2)/M)$$

Estos seran nuestros vectores de caracteristicas, generalmente M=13

Si tenemos una señal analizada en 1000 segmentos la matriz de caracteristicas tendrá 1300 valores, mucho menos que 16000!!!

Reconocimiento del Habla como clasificación de Patrones

- Posibles técnicas
  - Redes bayesianas
  - Modelos ocultos de Markov
  - Redes Neuronales (mapas autoorganizativos , redes de clasificacion espacio temporal de Tramas)

Dynamic Time Warping

- Un algoritmo optimizado que hace uso de la programación dinamica y es usado muchas veces por las técnicas anteriormente mencionadas ejem: modelos ocultos de markov

Dynamic Time Warping

- Para una palabra *A* buscar una palabra *Aw* que minimice la *distancia(A,Aw)*

A y Aw conjunto de valores de caracteristicas

$$D(A, A_w) = \sum_{t=1}^{T} DF(A(t), A_w(t))$$

donde DF es la distancia entre frames, y T es el numero de Frames que tiene una

Dynamic Time Warping

- Caso 1
  - A y B del mismo tamaño → Facil!!!

- Caso 2
  - Diferentes longitudes

Que podemos hacer?

El tiempo es el enemigo

- Normalizacion lineal???
  - *Distancia(A,Aw)=distancia(A,Aw)*

$$t' = t\frac{longitud(A_w)}{longitud(A)}$$

- Pero

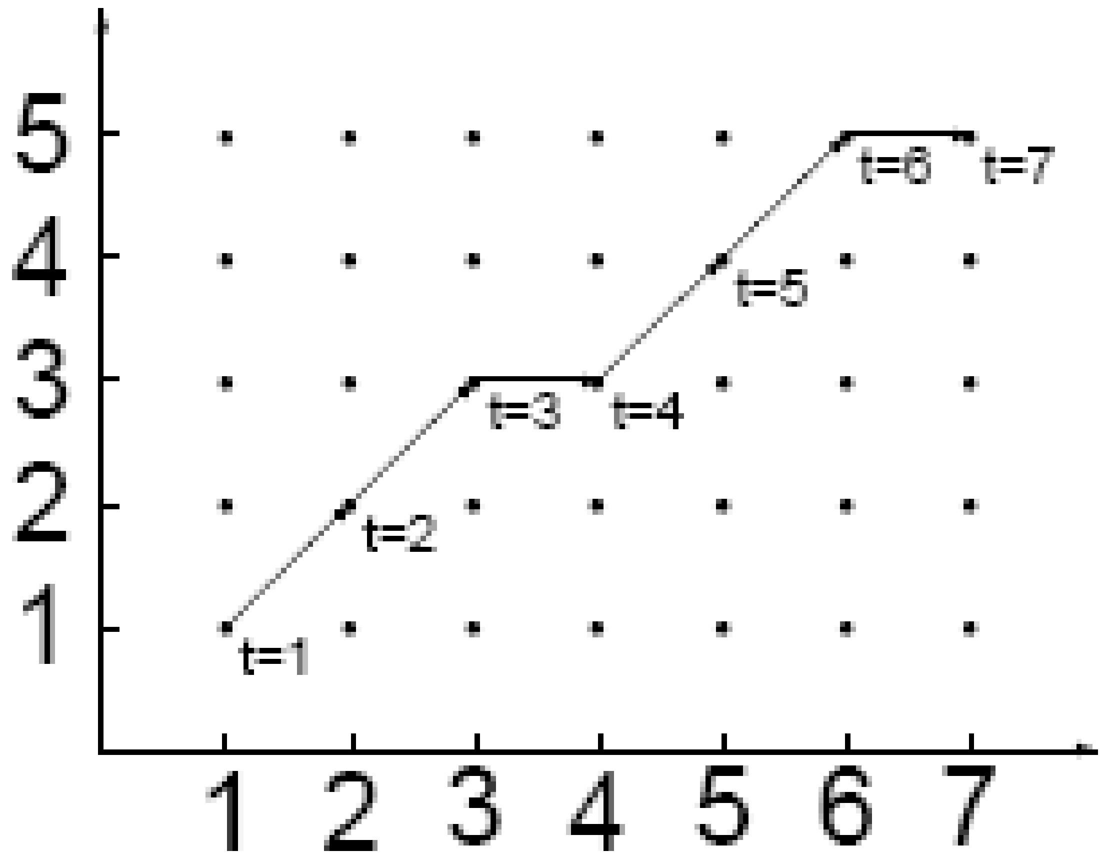
  S i l e n c i o | casa | silencio

  Silencio |c a s a |s i l e n c i o

Mala idea!!!

Dynamic Time Warping

Distancia (A,B)=Distancia(A(w(1)) , B(w(2)))



Dynamic Time Warping: reglas de juego

Condición de frontera:

$$i(1) = 1, j(1) = 1 \quad i(K) = I, j(K) = J$$

Condición de Monoticidad:

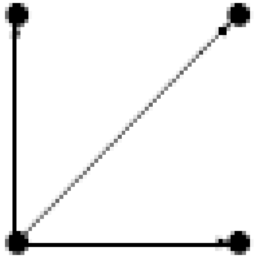
$$i(k-1) \leq i(k) \qquad j(k-1) \leq j(k)$$

Condición de Continuidad:

$$i(k) - i(k-1) \leq 1 \qquad j(k) - j(k-1) \leq 1$$

relación entre dos consecutivos

$$c(k-1) = \begin{cases} (i(k), j(k-1)) \\ (i(k-1), j(k-1)) \\ (i(k-1), j(k)) \end{cases}$$



Dynamic Time Warping: reglas de juego

- Analizar todas las distancias y encontrar la mejor es EXPONENCIAL!!!

- Solucion Programación dinámica
  - La solucion puede verse como un problema de La ruta mas corta $O(n^2)$

Dynamic Time Warping

- Formalmente:
  - Señales de voz $A = a_1, a_2, ..., a_i, ..., a_I$ $B = b_1, b_2, ..., b_j, ..., b_J$

    encontrar $F = c(1), c(2), ..., c(k), ..., c(K)$

    donde:

    $$c(k) = (i(k), j(k))$$

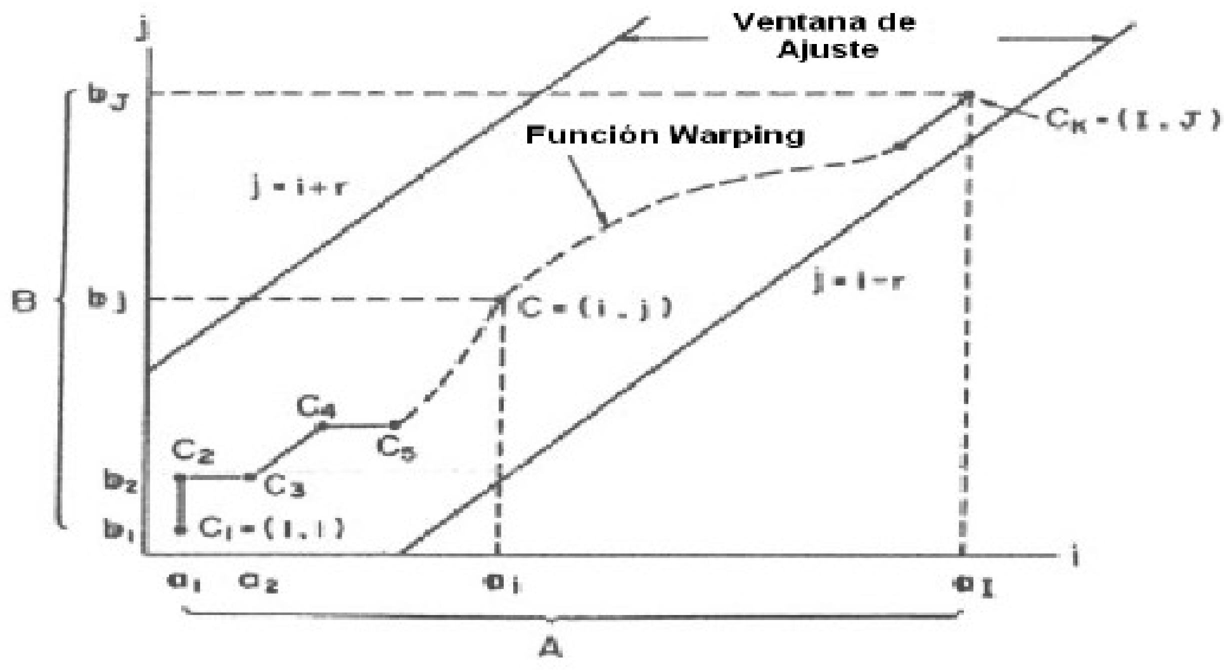
    La distancia: $d(c) = d(i,j) = ||a_i - b_j||$

La distancia normalizada entre dos patrones A y B está definida como:

$$D(A,B) = \min\left[\frac{\sum_{k=1}^{K} d(c(k)).w(k)}{\sum_{k=1}^{K} w(k)}\right]$$

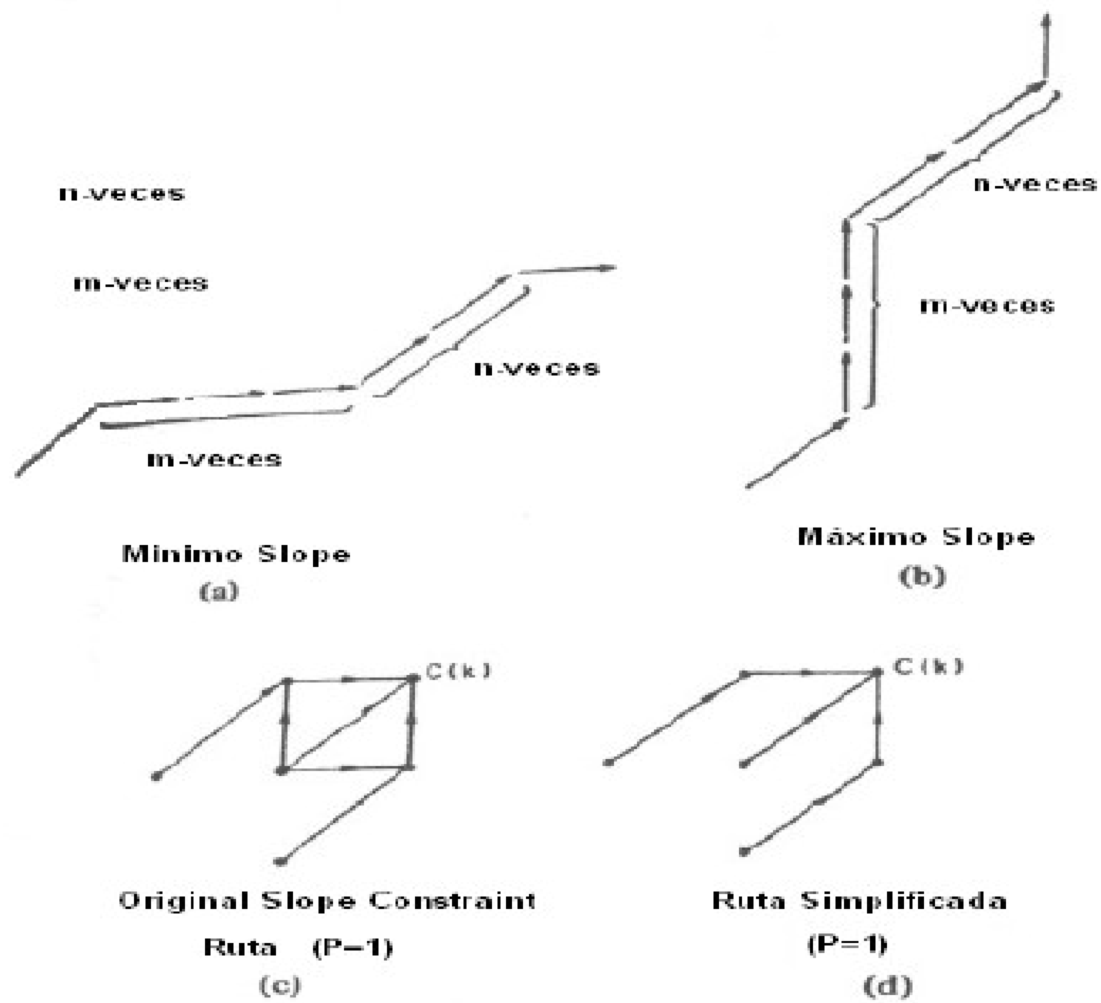
Dynamic Time Warping

- Idea: restringir un poco las condiciones : ventana de ajuste O(n)!!!! $|i(k) - j(k)| \leq r$

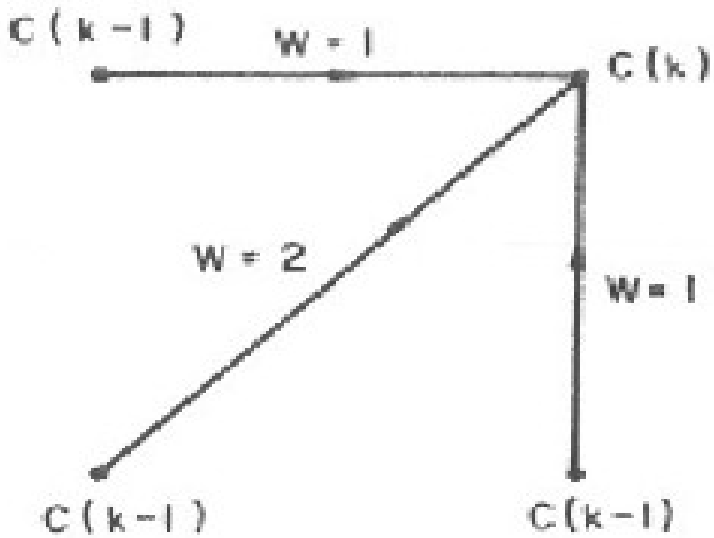


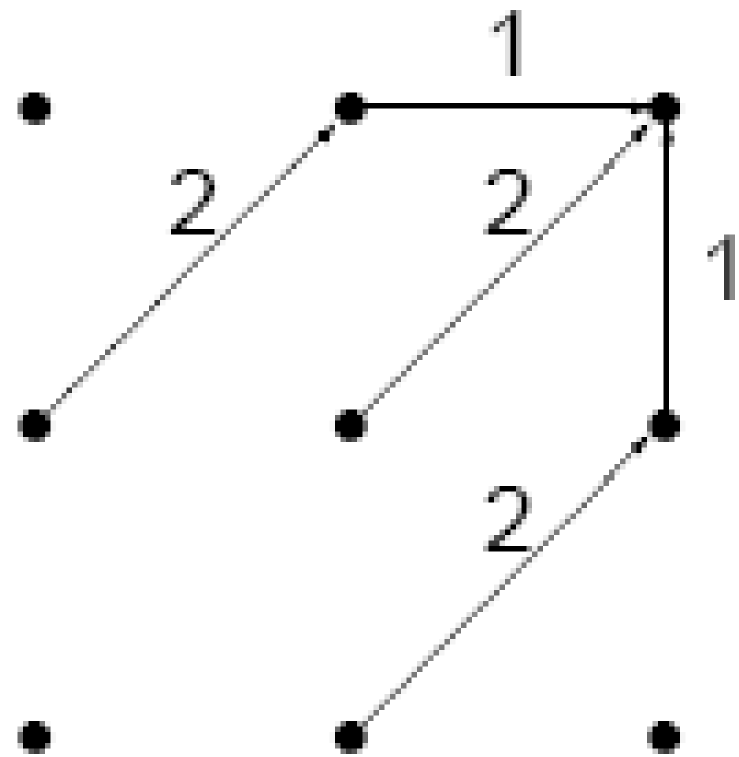
- Slope constraint

$$p = \frac{n}{m}$$



Dynamic Time Warping





Algoritmo PD matching

Condición Inicial :

$$g_1(c(1)) = d(c(1))w(1)$$

Ecuaciones PD:

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k))w(k)])$$

Distancia Normalizada en el Tiempo :

$$D(A, B) = \frac{1}{N} g_k(c(k))$$

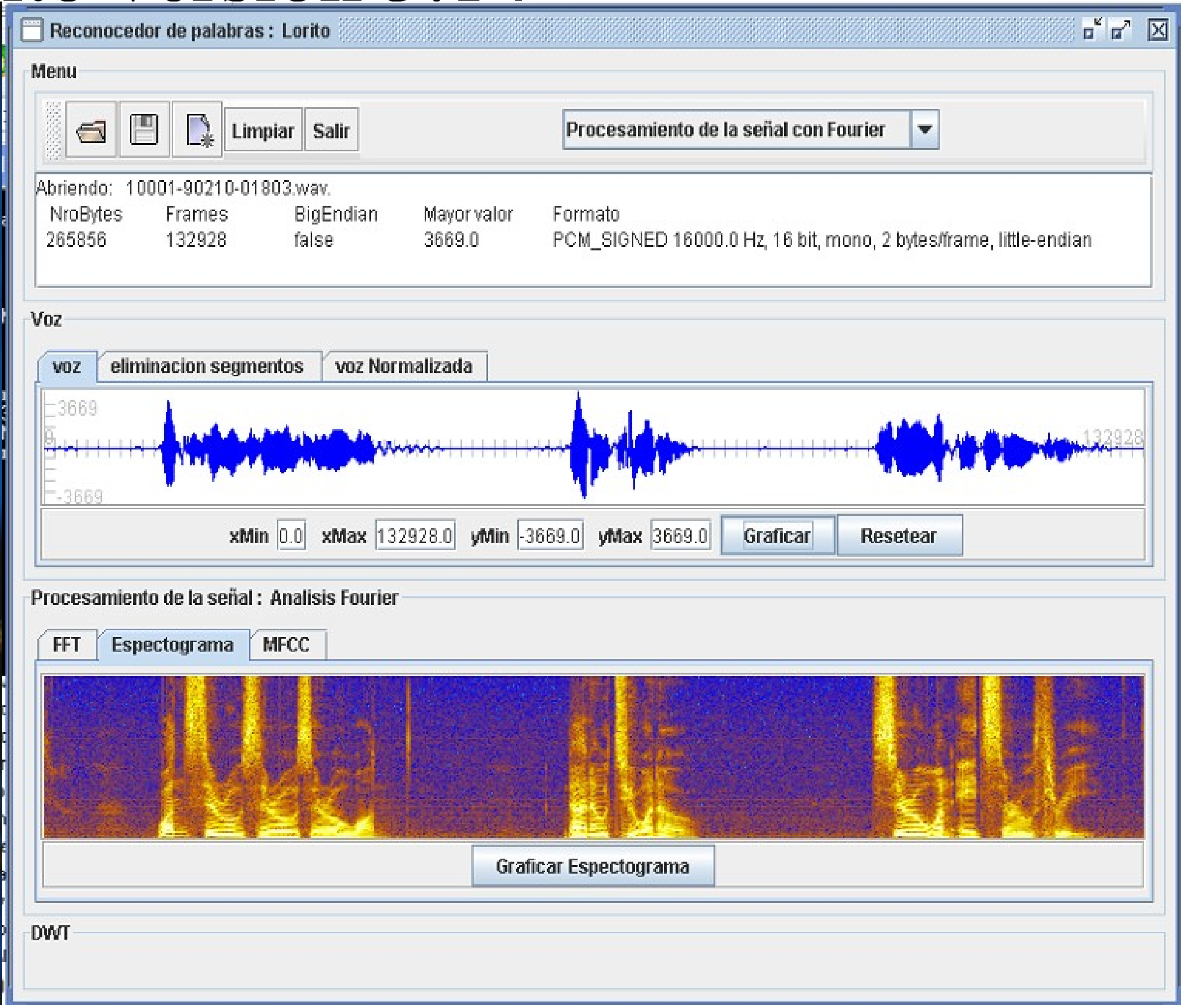
Ideas!!!

- Utilizar probabilidades
- Modelos hibridos
- Gramaticas
- Tecnicas de busqueda

- Mucho por recorrer

Presentacion del software para pruebas

Lorito version 3.14



Preguntas????

ideas???

eso es todo amigos!!!!

contacto:

jorjasso@hotmail.com