

CECS401  
Fundamentals of Spoken Language Processing

Note-12  
Thursday 10/7/99

## **H. Speech vs. Background Discrimination**

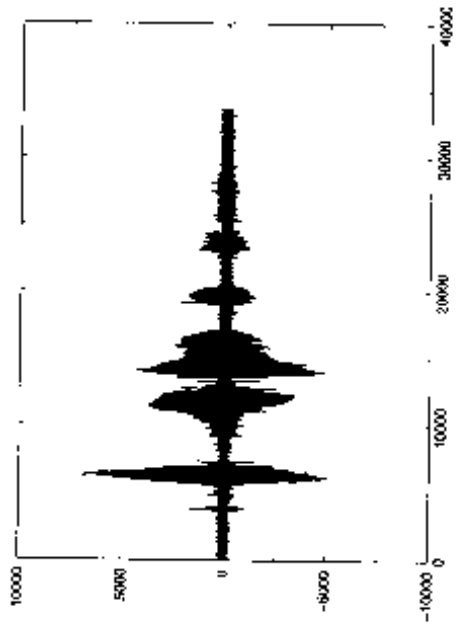
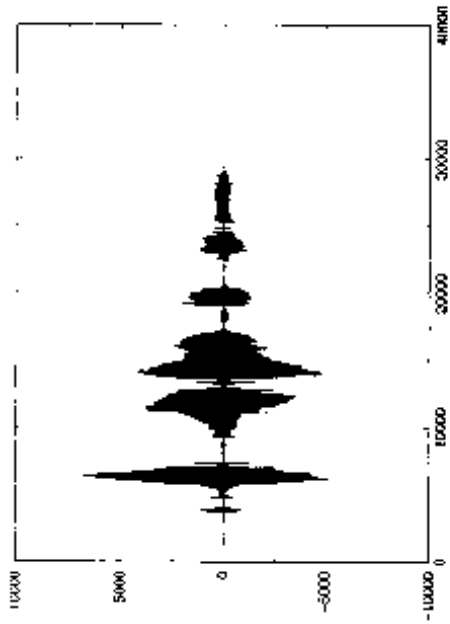
For accurate recognition of speech, it is desired to separate speech signal from background sound. The process of locating the beginning and ending points of speech from a recording is called speech end-points detection.

**Explicit end-point detection:**

First determine the end-points of speech, and then perform speech recognition. Such end-point detection is needed by template-based recognition systems.

**Implicit end-point detection:**

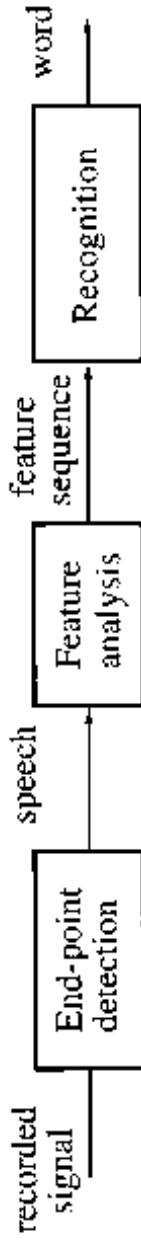
The determination of speech end-points is embedded in speech recognition. Such end-point detection is easily implemented in statistical-model based speech recognition systems.



### Background sounds

- Silence
- Noise: fan, machine engine, door slam, TV, conversation, etc.
- speech artifacts: lip smacks, heavy breathing, etc.

### Explicit end-point detection using energy and zero-crossing



- energy: discrimination of speech and silence
- zero-crossing: discrimination of speech and noise

Short-time energy  $M_n$  (actually magnitude):

$$\begin{aligned} M_n &= \sum_{m=0}^{N-1} |s_n(m)| \\ &= \sum_{m=-\infty}^{\infty} |s(m)|w(n-m) \end{aligned}$$

$w(m)$  is a short-time window of length  $N$ .

Zero-crossing rate  $Z_n$ :

$$Z_n = \sum_{m=-\infty}^{\infty} |\operatorname{sgn}[s(m+1)] - \operatorname{sgn}[s(m)]|w(n-m)$$

$$\operatorname{sgn}[x] = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

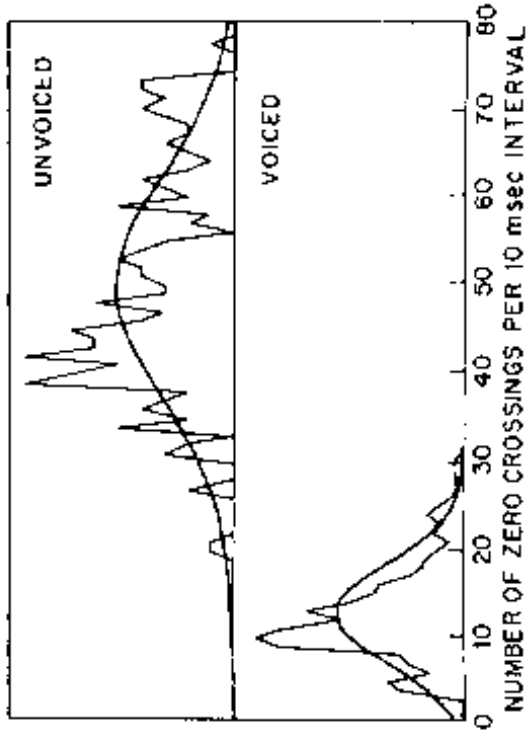
Zero-crossing rate measures the signal frequency content (particularly true for narrow band signals).

A sinusoidal signal has two zero-crossings per cycle.

If the sampling rate is  $F_s$  and the frequency is  $F_0$ , then the number of zero-crossings in a window is

$$Z = \frac{NT_s}{T_0} \times 2 = \frac{2N}{F_s} F_0$$

Example:



30 zero-crossings / 10 ms — 3,000 Hz.

80 zero-crossings / 10 ms — 8,000 Hz.

	voiced speech	unvoiced speech	noisc
energy	high	low	low
zero-crossing	low	high	low

A simple algorithm for speech end-points detection

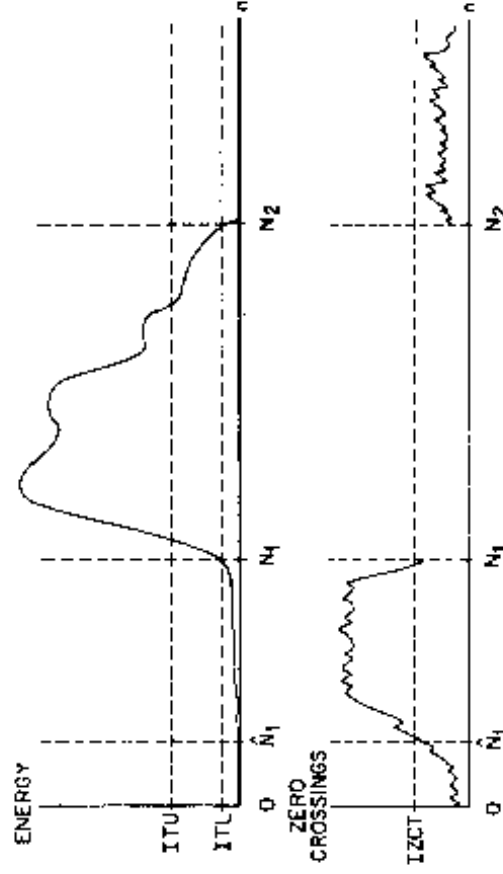


Fig. 4.16 Typical example of average magnitude and zero-crossing measurements for a word with a strong fricative at the beginning. (After Rabiner and Sambur [6].)

ITU, ITL — high and low threshold of magnitudes  
 IZCT — threshold of zero-crossing

Step-1.

Identify high-energy region of speech from both ends (forward and backward) using the high-energy threshold ITU.

Step-2.

Refine the detected boundaries by extending from the high energy region outward using the low threshold ITL, producing the end-points  $N_1$  and  $N_2$ .

Step-3.

Further refine the boundaries using the zero-crossing threshold IZCT, generating the final end-points  $\hat{N}_1$  and  $\hat{N}_2$ .

The threshold values are determined during speech inactive periods.

## **I. Speech Pattern Comparison using Dynamic Time Alignment**

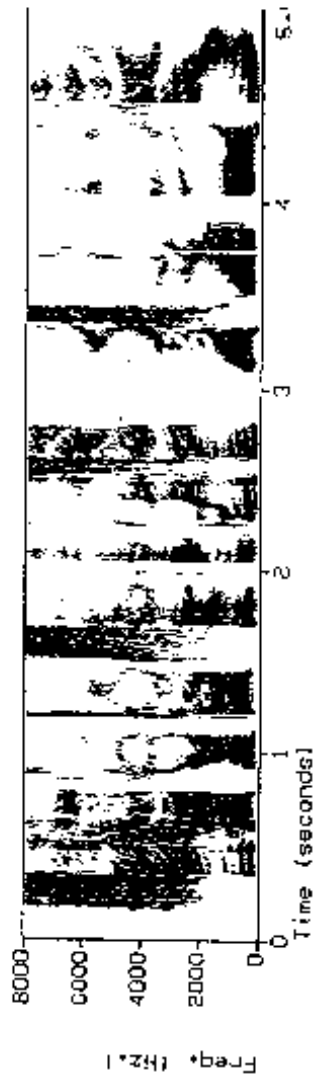
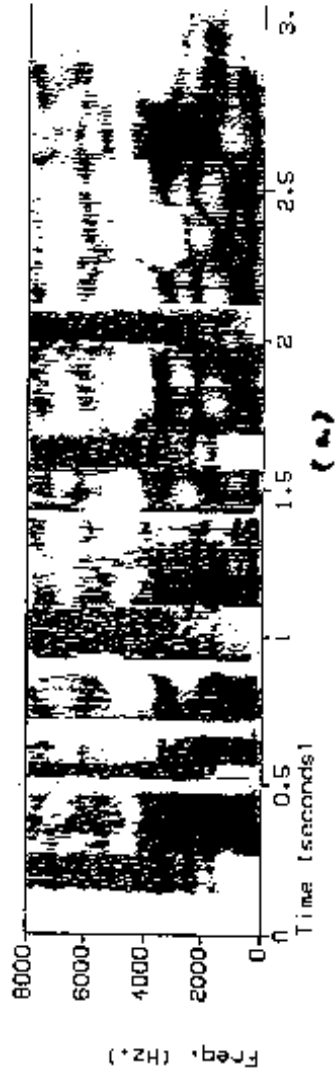
### **The need for time alignment:**

- Speech rate across talkers can vary by more than a factor of two.
- Durations of words vary with speaking context and environment,  
e.g.,  
emphasized words, words produced in noisy conditions usually  
have longer durations.
- Through time alignments, speech templates of different durations can be compared on a common time axis.

**Example**

Two SAI sentences in TIMIT

"She had your dark suit in greasy wash water  
all year."



(b)

### Template Matching:

Assume a stored template  $Y$  of duration  $T_Y$  and a test pattern  $X$  of duration  $T_X$ , i.e.,

$$Y = (y_1, y_2, \dots, y_{T_Y})$$

$$X = (x_1, x_2, \dots, x_{T_X})$$

How to compare the two patterns and compute the distance  $D(X, Y)$ ?

### Example

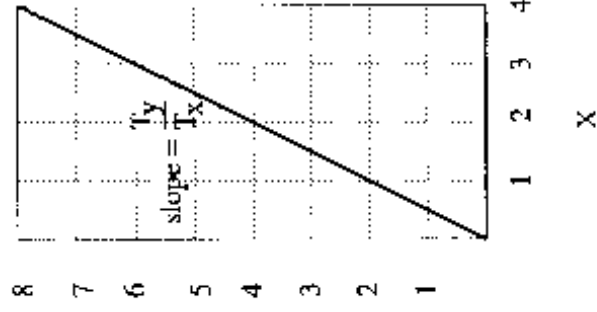
$$Y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$$

$$(w, w, w, w, iy, iy, iy, iy)$$

$$X = (x_1, x_2, x_3, x_4)$$

$$(w, w, iy, iy)$$

Linear time-alignment:



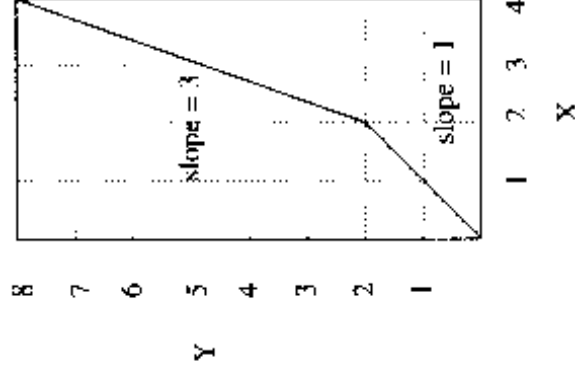
$$D(X, Y) = d(x_1, y_2) + d(x_2, y_4) + d(x_3, y_6) + d(x_4, y_8)$$

A good match is achieved.

Example:  $Y$  is changed to  $(w, w, iy, iy, iy, iy, iy, iy)$ .

Using linear-time alignment will produce a mismatch at  $d(x_2, y_4)$ .

Nonlinear time-alignment:



$$D(X, Y) = d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_5) + d(x_4, y_8)$$

Again, a good match is achieved.

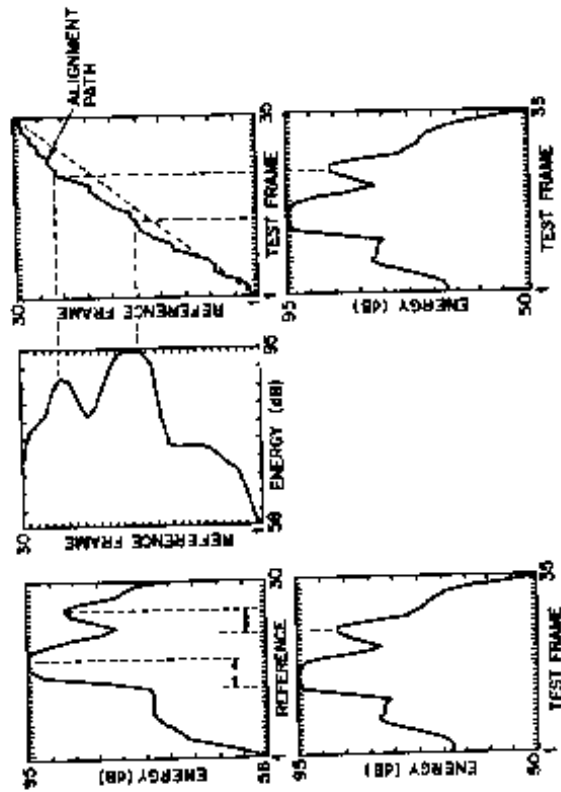


Figure 4.51 Example illustrating the need for nonlinear time alignment of two versions of a spoken word.

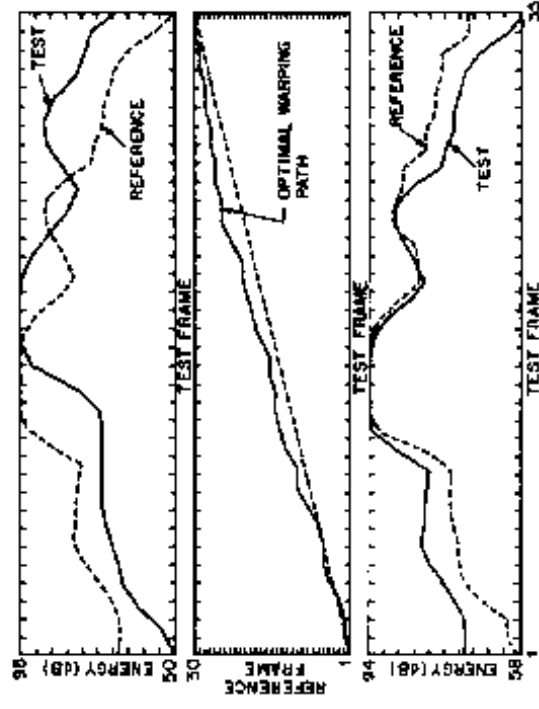


Figure 4.52 Illustration of the effectiveness of dynamic time warping alignment of two versions of a spoken word.

## Dynamic Time Warping (DTW)

Mapping the time axes of  $Y$  and  $X$  into a common time axis through searching an optimal time-warping path.

Definitions:

time axis of $X$	$i_x$	$1 \leq i_x \leq T_x$
time axis of $Y$	$i_y$	$1 \leq i_y \leq T_y$
common time index	$k$	$1 \leq k \leq T$
warping function	$i_x = \phi_x(k)$	$i_y = \phi_y(k)$

Warping path:  $\phi = ((\phi_x(1), \phi_y(1)), (\phi_x(2), \phi_y(2)), \dots, (\phi_x(T), \phi_y(T)))$

Constraints:

$$\begin{aligned}\phi_x(1) &= 1, \phi_x(T) = T_x \\ \phi_y(1) &= 1, \phi_y(T) = T_y\end{aligned}$$

$$D_\phi(X, Y) = \sum_{k=1}^T d(x_{\phi_x(k)}, y_{\phi_y(k)})$$

**Example**

Specify the warping path chosen in previous example.

$$\begin{aligned} \mathbf{k}=1 & \quad \phi_x(1) = 1, \phi_y(1) = 1 \\ \mathbf{k}=2 & \quad \phi_x(2) = 2, \phi_y(2) = 2 \\ \mathbf{k}=3 & \quad \phi_x(3) = 3, \phi_y(3) = 5 \\ \mathbf{k}=4 & \quad \phi_x(4) = 4, \phi_y(4) = 8 \end{aligned}$$

$$\phi = ((1, 1), (2, 2), (3, 5), (4, 8))$$

**Example**

Given the warping path below, determine the warping functions  $\phi_x$  and  $\phi_y$ .

