An Algorithm for Determining the Endpoints for Isolated Utterances

L.R. Rabiner and M.R. Sambur

The Bell System Technical Journal, Vol. 54, No. 2, Feb. 1975, pp. 297-315

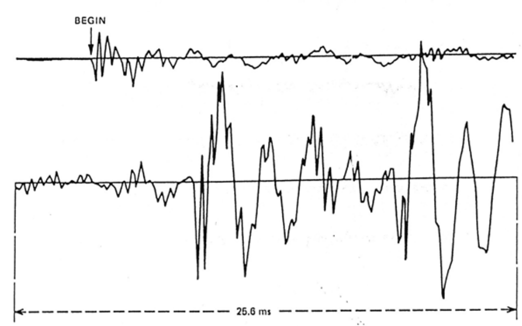
Outline

- Intro to problem
- Solution
- Algorithm
- Summary

Motivation

- Word recognition needs to detect word boundaries in speech
- Recognizing silence can reduce:
  - Processing load
  - (Network not identified as savings source)
  - (Hands-free operation not identified as convenience)
- Relatively easy in sound proof room, with digitized tape

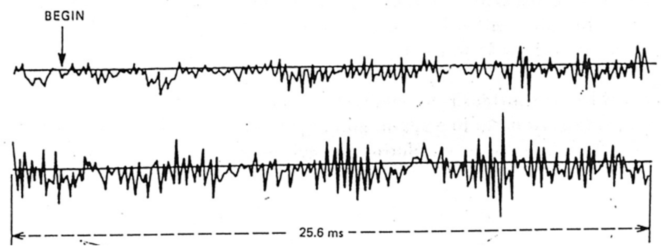
Visual Recognition



"Eight"

- Easy
- Note how quiet beginning is (tape)

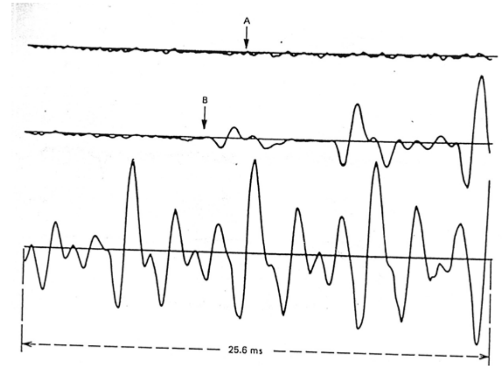
Slightly Tougher Visual Recognition



"Six"

- "sss" starts crossing the 'zero' line, so can still detect

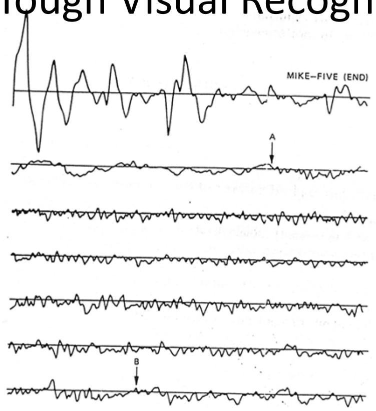
Tough Visual Recognition



"Four"

- Eye picks 'B', but 'A' is real start
  - /f/ is a *weak fricative*

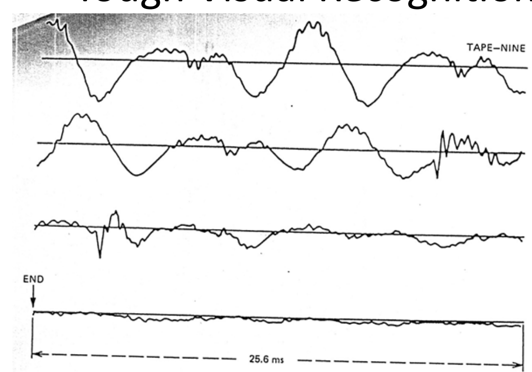
Tough Visual Recognition



"Five"

- Eye picks 'A', but 'B' is real endpoint
  - V becomes *devoiced*

Tough Visual Recognition



"Nine"

- Difficult to say where final trailing off ends

The Problem

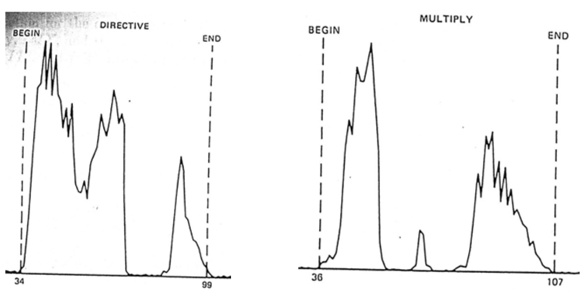
- Noisy computer room with background noise
  - Weak fricatives: /f, th, h/
  - Weak plosive bursts: /p, t, k/
  - Final nasals (ex: "nine")
  - Voiced fricatives becoming devoiced (ex: "five")
  - Trailing off of sounds (ex: "binary", "three")
- Need to do with simple, efficient processing
  - Avoid hardware costs

The Solution

- Two measurements:
  - Energy
  - Zero crossing rate
- Show: simple, fast, accurate

Energy

- Sum of magnitudes of 10 ms of sound, centered on interval:
  - $E(n) = \sum_{i=-50 \text{ to } 50} |s(n+i)|$



Zero (Level) Crossing Rate

- Remember, digital audio values are changes in air pressure (higher or lower than base)
- Base/midpoint is "zero"
  - But is always positive if unsigned (e.g., 127 if unsigned byte)
- Zero crossing rate is number of zero crossings per 10 ms
  - Normal number of cross-overs during silence
  - Increase in cross-overs during speech

The Algorithm: Startup

- At initialization, record sound for 100ms
  - A measure background noise
  - Assume ‘silence’
- Compute average (IZC’) and std dev (σ) of zero crossing rate
- Choose zero-crossing threshold (IZCT)
  - Threshold for unvoiced speech
  - IZCT = min(25 / 10ms, IZC’ + 2 σ)

The Algorithm: Thresholds

- Compute energy, $E(n)$, for interval
  - Get max, IMX
  - Have ‘silence’ energy, IMN
  - Compute to values:
    - $I1$ = 0.03 * (IMX – IMN) + IMN
      (3% of peak energy)
    - $I2$ = 4 * IMN
      (4x silent energy)
- Get energy thresholds (ITU and ITL)
  - ITL = MIN($I1$, $I2$)
  - ITU = 5 * ITL

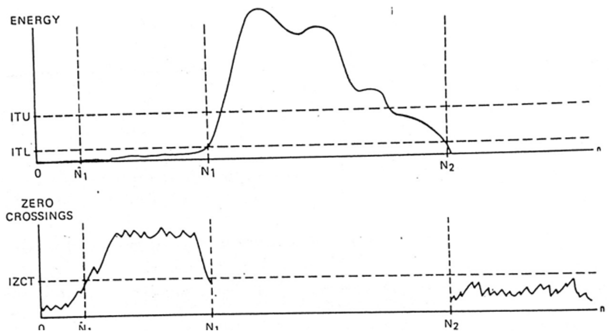
The Algorithm: Energy Computation

- Search sample for energy greater than ITL
  - Save as start of speech, say *s*
- Search for energy greater than ITU
  - *s* becomes start of speech
  - If energy falls below ITL, restart
- Search for energy less than ITL
  - Save as end of speech
- Results in conservative estimates
  - Endpoints may be outside

The Algorithm: Zero Crossing Computation

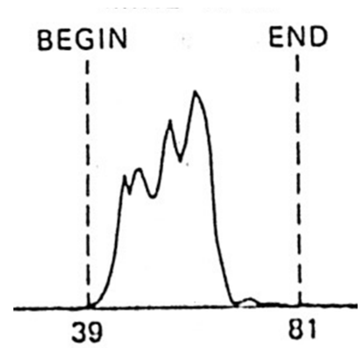
- Search back 250 ms
  - Count number of intervals where rate exceeds IZCT
    - If 3+, set starting point, *s*, to first time
    - Else *s* remains the same
- Do similar search after end

The Algorithm: Example



(Word begins with strong fricative)

Algorithm: Examples



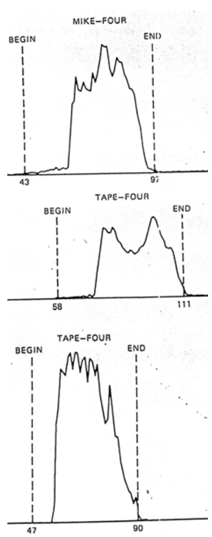
"Half"

- Caught trailing /f/

Algorithm: Examples

"Four"

(Notice how different each "four" is)



Evaluation: Part 1

- 54-word vocabulary
- Read by 2 males, 2 females
- No gross errors (off by more than 50 ms)
- Some small errors
  - Losing weak fricatives
  - None affected recognition

Evaluation: Part 2

- 10 speakers
- Count 0 to 9
- No errors at all

Evaluation: Part 3

- Your Project 1b…