Speaker Transformation Algorithm using Segmental Codebooks (STASC)

Presented by A. Brian Davis

Speaker Transformation

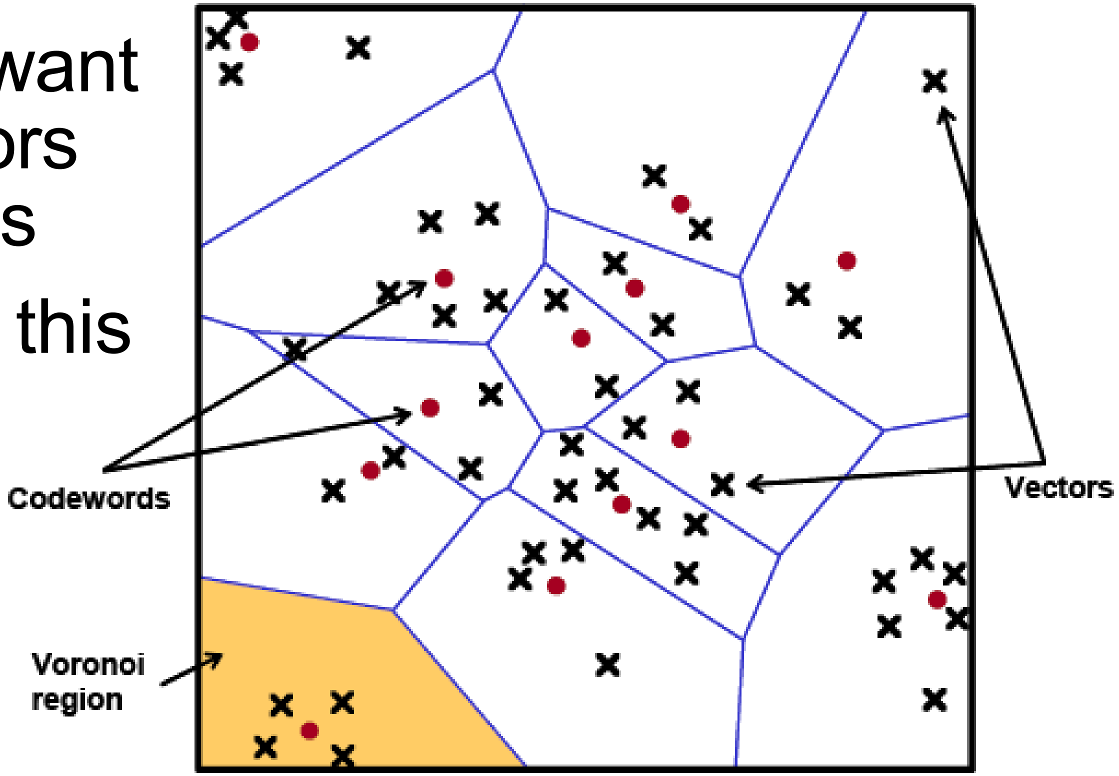
- Goal: map acoustic properties of one speaker onto another
- Uses:
  - Personification of text-speech systems
  - Multimedia
  - Preprocessing step for speech recognition
    - Reduce speaker variability
    - Practical?

Steps Involved

- Training phase
  - Given speech input from source and target, form spectral transformation
  - Inputs / outputs to transformation:
    - Segment speech small chunks (frames)
      - Formants
      - LPC cepstrum coefficients
      - Others (excitation)?
  - Can we generalize behavior of transform?
    - Codebooks/codewords
      - Vector quantization

Vector quantization

- Assign vectors to discrete set of values
- K-Means
  - For STASC, also want "average" all vectors assigned to a class
  - K-Means gives us this for free



Shamelessly stolen from Dr. Gutierrez's pattern recognition slides

LSFs

- Line spectral frequencies
- Derived (losslessly) from LPC's
  - Can convert to/from, thus can create speech from LSFs
- Relate to formant frequencies
  - Used in STASC represent vocal tract of speakers
- Stable
- Why use instead of MFCCs?

STASC (first method)

- Assumes orthographic transcription
  - What's said, in writing
- From transcription, phonemes retrieved
  - Speech segments assigned phoneme based on transcription
    - MFCCs, dMFCCs for each segment (frame) passed into HMM, most likely path using Viterbi algorithm
  - LSFs calculated per frame, labeled with phoneme from HMM
  - Phoneme centroids calculated (average LSF values all vectors labeled particular phoneme
  - One-one mapping

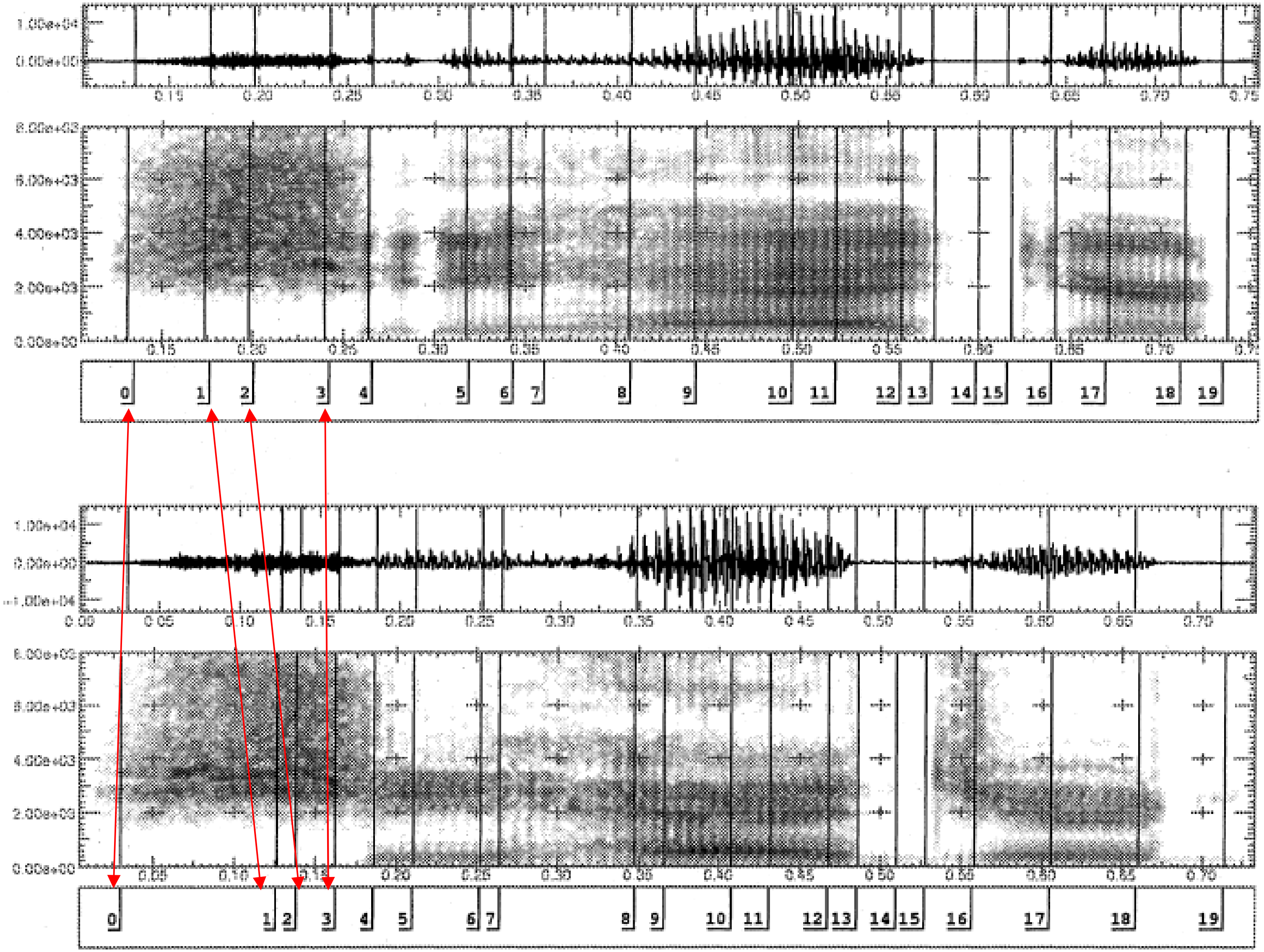


Fig. 1. The state alignments for source and target speaker utterances “She had your”.

Second method (better)

- No orthographic transcription
  - Intuitively, we know the HMM states in 1st method didn't need correspond phonemes
    - Require speakers speak same (hopefully phonetically balanced) sentence
      - Sentences with phones approx. distributed as in normal speech
  - Because fewer restrictions, need to do some extra processing of speaker's speech
    - Normalize root-mean-squared energy
    - Remove silence before/after speech

Second method transformation

- HMM trained on each sentence
  - Data from source speaker's speech segments
    - LSF vectors
  - Number of states correspond sentence length
  - Segmental k-means, separates speech segments into clusters
  - Baum-Welch algorithm train HMM on cluster averages
    - Covariance matrix uniform
- For source/target speech segments, Viterbi algorithm assigns segments to states.
- Transformation moves segments from state in source to state in target
    - Centroids

Excitation characteristic

- From previous papers, know excitation greatly influences perception of speaker
- Not trivial to transfer
  - Very different for voiced / unvoiced sounds
- Use current codebooks to transfer excitation
  - Calculate short-time average magnitude spectrum of excitation signal each “speech unit”

Codebook weight estimation

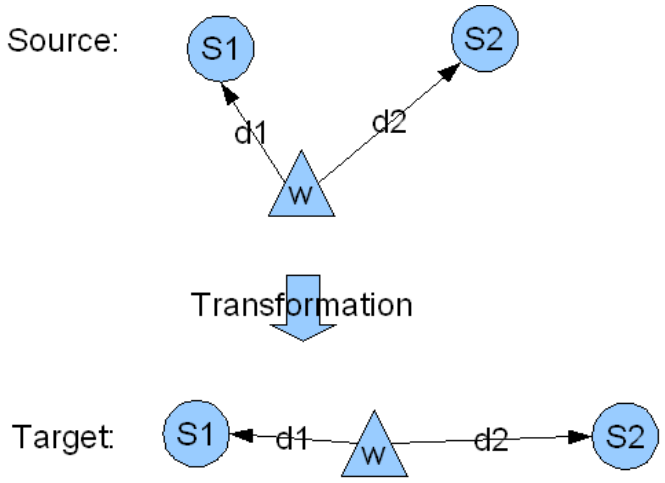
- Assume we have vector w of LSFs labeled with HMM state
- Also centroids Si of each HMM state
- Algorithm:
  - Calculate distances di from w to Si
    - Perceptual distance – closely spaced LSFs correspond to formant locations given higher weight
  - From distances, calculate weights vi, represent w as linear combination Si's
- Minimize error?

Gradient Descent

- Find local optimum weights minimize error reconstructed LSFs vs actual LSFs
- Algorithm:
  - Find gradient of difference reconstruction, predicted (weighted perceptually)
  - Weight gradient by small value (speed to convergence)
  - Add to old weights
  - Until difference in weights between iterations is sufficiently small
- Found that only few weights given large value
  - Only use 5 most likely weights
- 15% additional reduction in Itakura-Saito distance, .4 dB error

Use of weights

- Given reconstruct LSF vector (segment of speech from speaker) from linear combination of sigmoids
- Use those weights and target's sigmoids, use resulting LSFs to reconstruct speech
- Other transformations?
  - Excitation spectral characteristics
  - Prosody
  - Can estimate new weights for all, but why?



Artist's impression

Excitation and Vocal Tract

- Use weights construct excitation filter
  - linear combination of sigmoids' (average target excitation magnitude spectra) over (source EMS)
- Use weights construct vocal tract spectrum – convert transformed LSF vectors to LPCs

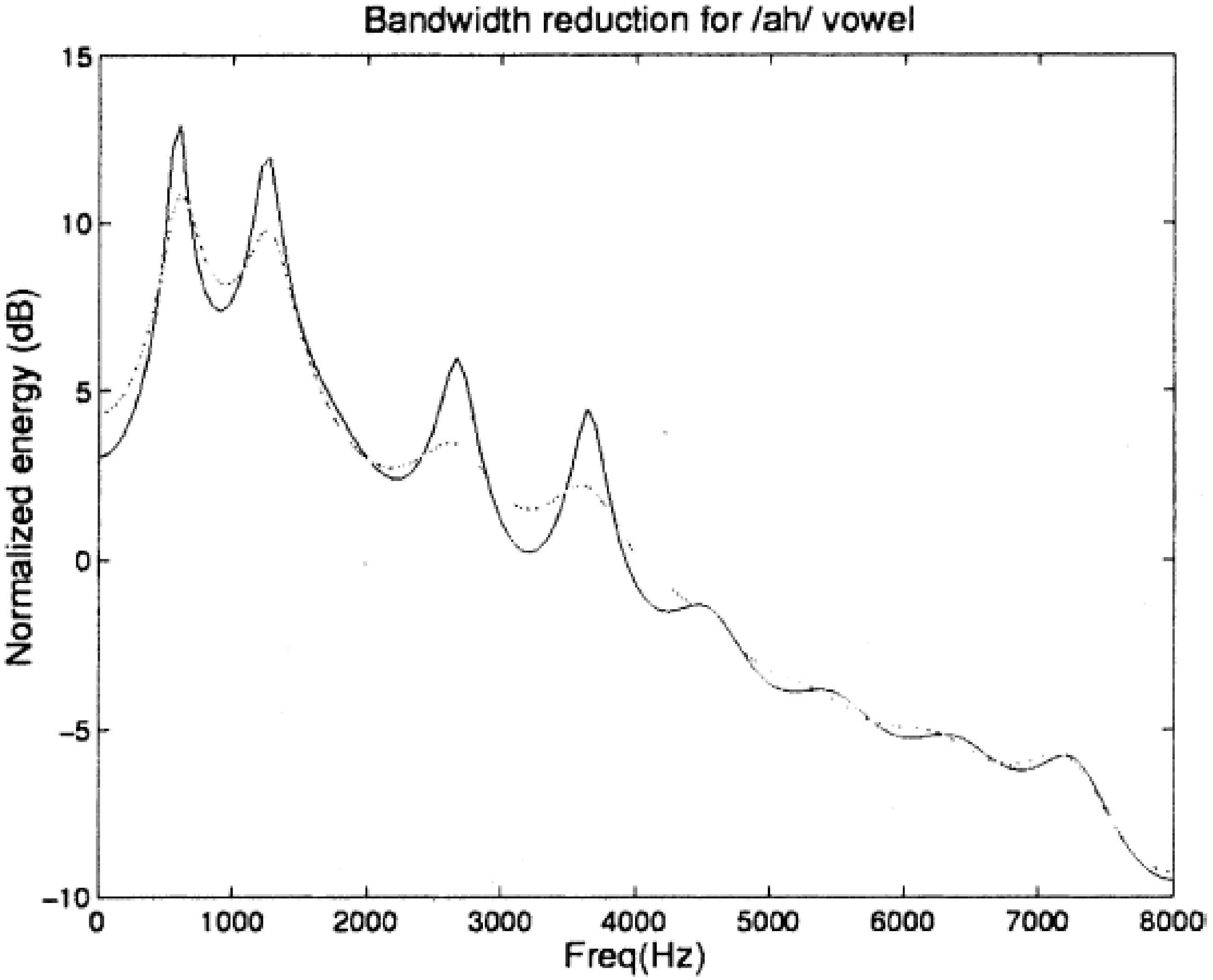
$$V_t(\omega) = \left| \frac{1}{1 - \sum_{k=1}^{P} a_k^t e^{-jk\omega}} \right|$$

  - Expansion of bandwidths; gives unnatural speech

Bandwidth modification

- Assume average formant bandwidth values of target speaker similar most likely target codeword (LSF centroid)
- Since LSFs correspond to formant locations / bandwidths, change bandwidths by changing adjacent LSF distances
- Algorithm:
  - Find LSF entries directly before/after each formant location in most likely Target codeword
  - Calculate average formant bandwidth
  - Same for corresponding speech segment LSF vectors
  - form ratio of average codeword bandwidth over segment bandwidth
  - Apply estimated bandwidth ratio to adjust LSFs of speech segment vectors
  - Enforce reasonable bandwidths (average bandwidth of most likely centroid from target speech over 20

Bandwidth modification result



Prosodic Transformation

- Pitch, duration, energy modified to mimic target
- Dynamic segment lengths
  - Constant for unvoiced, 2-3 pitch periods for voiced
- Pitch:
  - No weights involved
  - Modify f0 linearly, matching variance f0s, matching averages

Duration

- Uniform duration matching?
- Different people pronounce different phonemes differently
- Need finer control duration modification

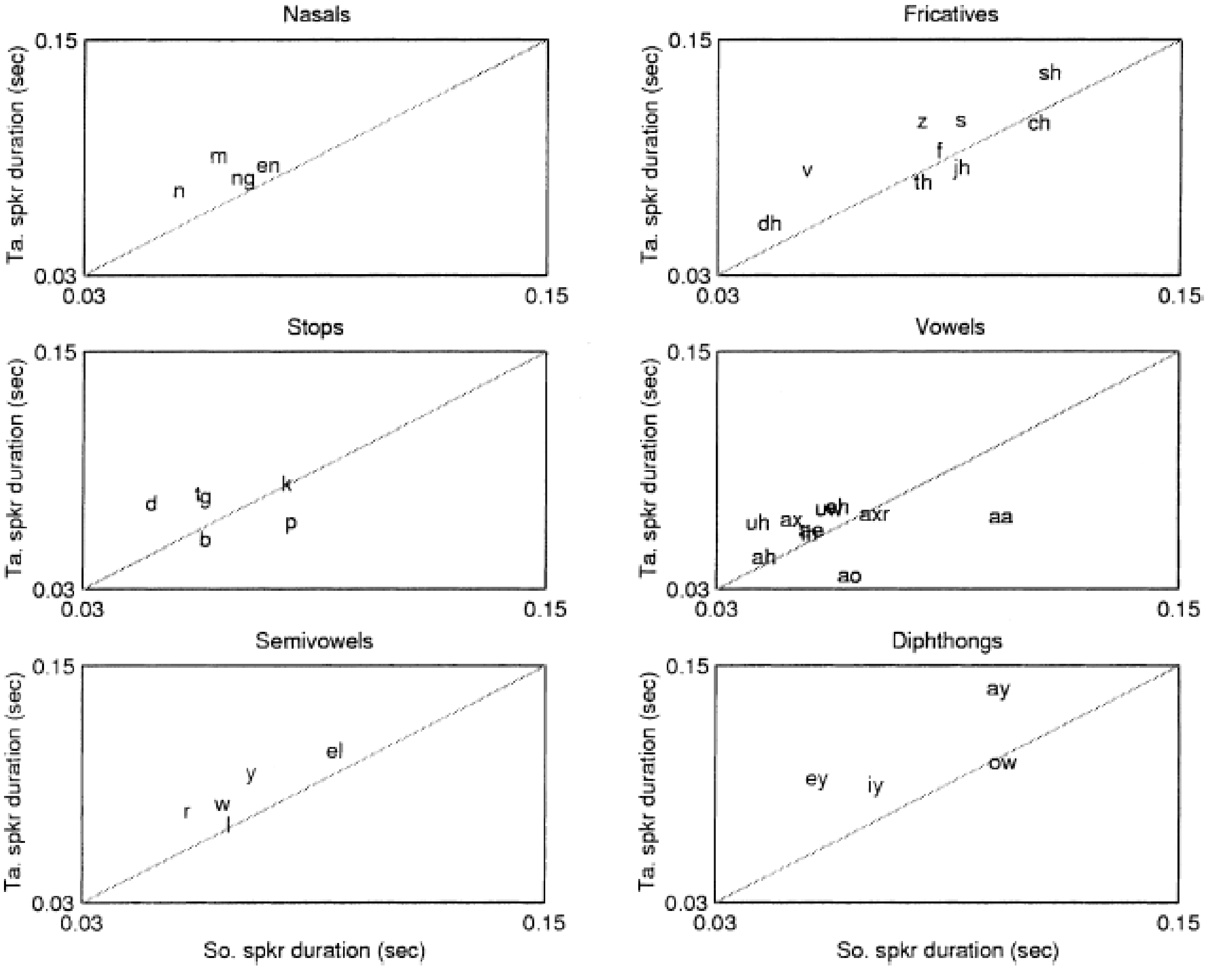
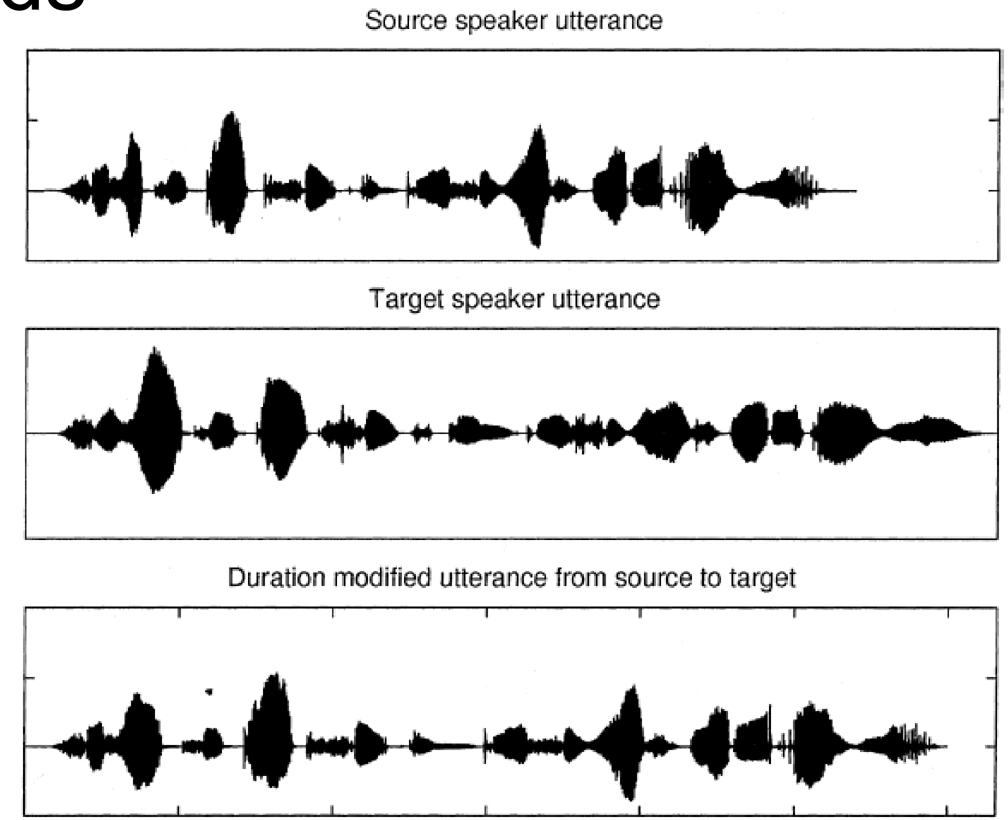


Fig. 5. Comparison of duration statistics between a source speaker and a target speaker.

Duration modification

- Duration phoneme dependent context (coarticulation)
  - Triphones as speech units
- Find speech unit centroids (durations), weights per segment, form target duration as linear combination
- Uses?
  - Human transcription



Energy scale modification

- Another characteristic of speaker
- Algorithm (finding energy scaling factor per time frame):
  - Calculate RMS energy for each codeword
  - Derive weights for representing scaling factor as linear combination (target's RMS energy) over (source's RMS energy)
  - After applying other modifications, scale energy

Evaluations

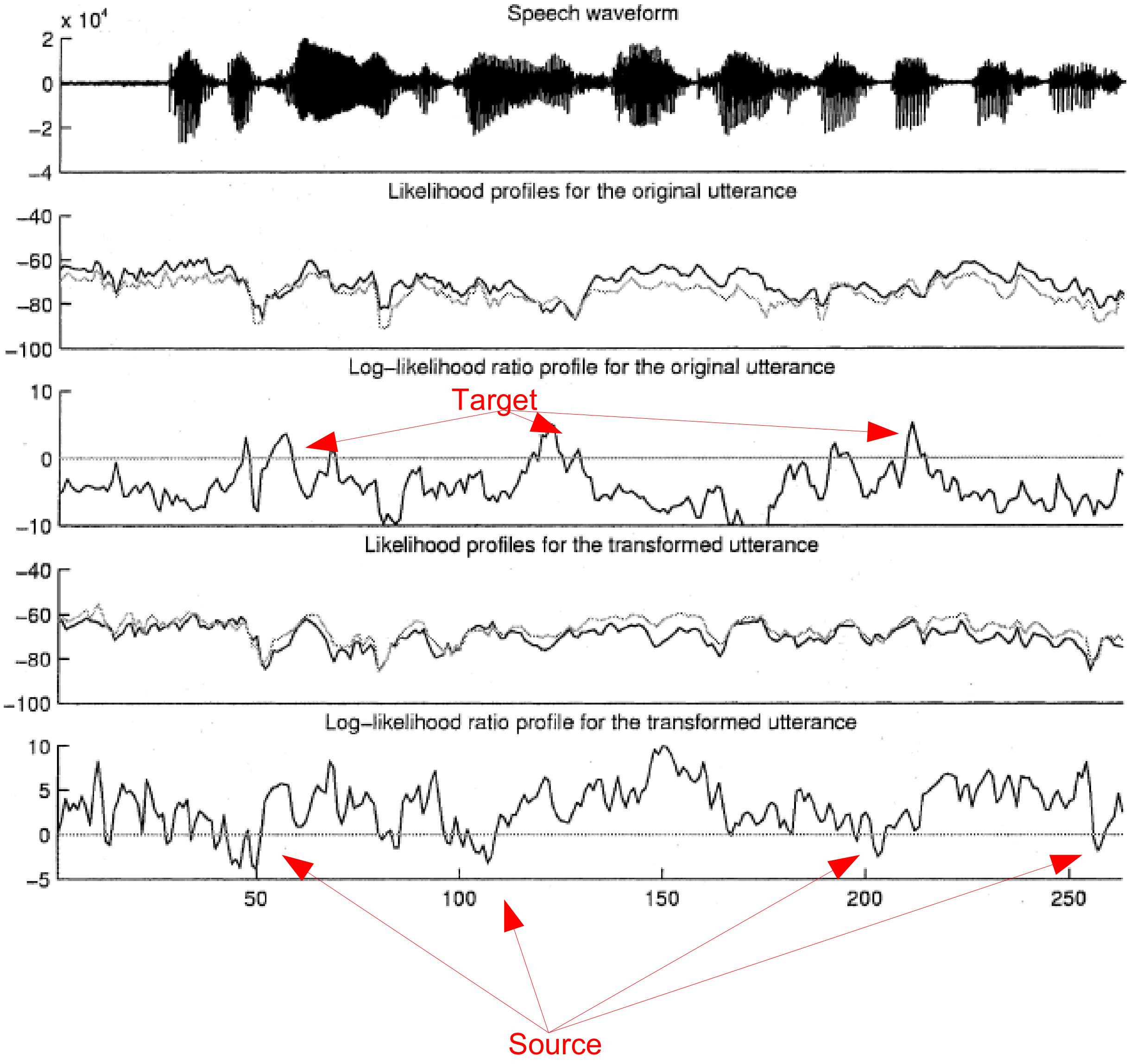
- Want to test effectiveness of transformation
  - Speaker recognition
  - Speech recognition
- Objective and subjective
  - Automatic speech recognizer
  - Human subjects
    - Test

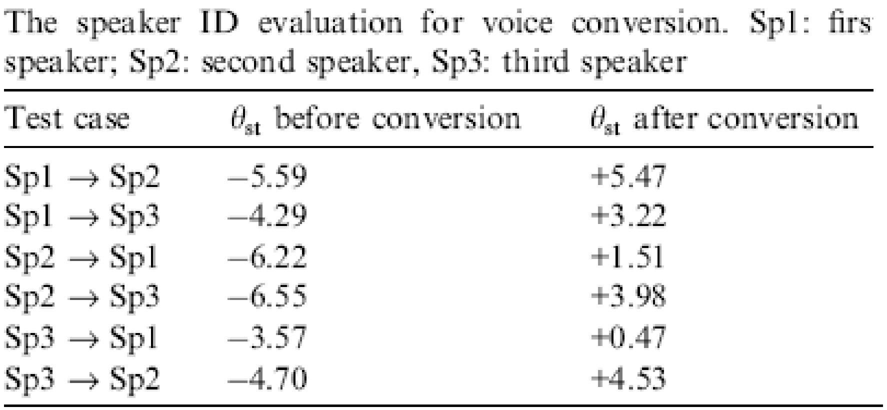
Objective

- Idea: confuse a speaker recognition machine
  - Stacking the deck
  - Confidence measure $\theta_{st} = \log\left(\frac{P(X|\lambda_t)}{P(X|\lambda_s)}\right)$
- The machine:
  - 256 mixture Gaussian mixture models
    - 24 dimension feature vector (MFCCs, deltas)
  - Binary split vector quantization
    - One vector for all, split to two in arbitrary directions
  - Train HMM
    - 3 speakers, speaking 1 hour each; 45 minutes for training
    - Different sentences (first method)
  - 15 minutes set aside for testing

Testing

- Multiple speakers
  - Each transformed another
- Context dependent





The speaker ID evaluation for voice conversion. Sp1: firs speaker; Sp2: second speaker, Sp3: third speaker

Test case	θ_{st} before conversion	θ_{st} after conversion
Sp1 → Sp2	−5.59	+5.47
Sp1 → Sp3	−4.29	+3.22
Sp2 → Sp1	−6.22	+1.51
Sp2 → Sp3	−6.55	+3.98
Sp3 → Sp1	−3.57	+0.47
Sp3 → Sp2	−4.70	+4.53

Objective (2)

- Sentence HMM
  - Source / target speak same sentences
  - 15 minutes speech from 2 M, 1F
    - Transform 1 M into M/F
- Phonetic codebooks also used; compare the two
- Measure fidelity to:
  - Cepstrum
  - Excitation spectrum
  - RMS energy
  - F0
  - Duration
- Results show sentence HMM better; increased training

Objective (2)

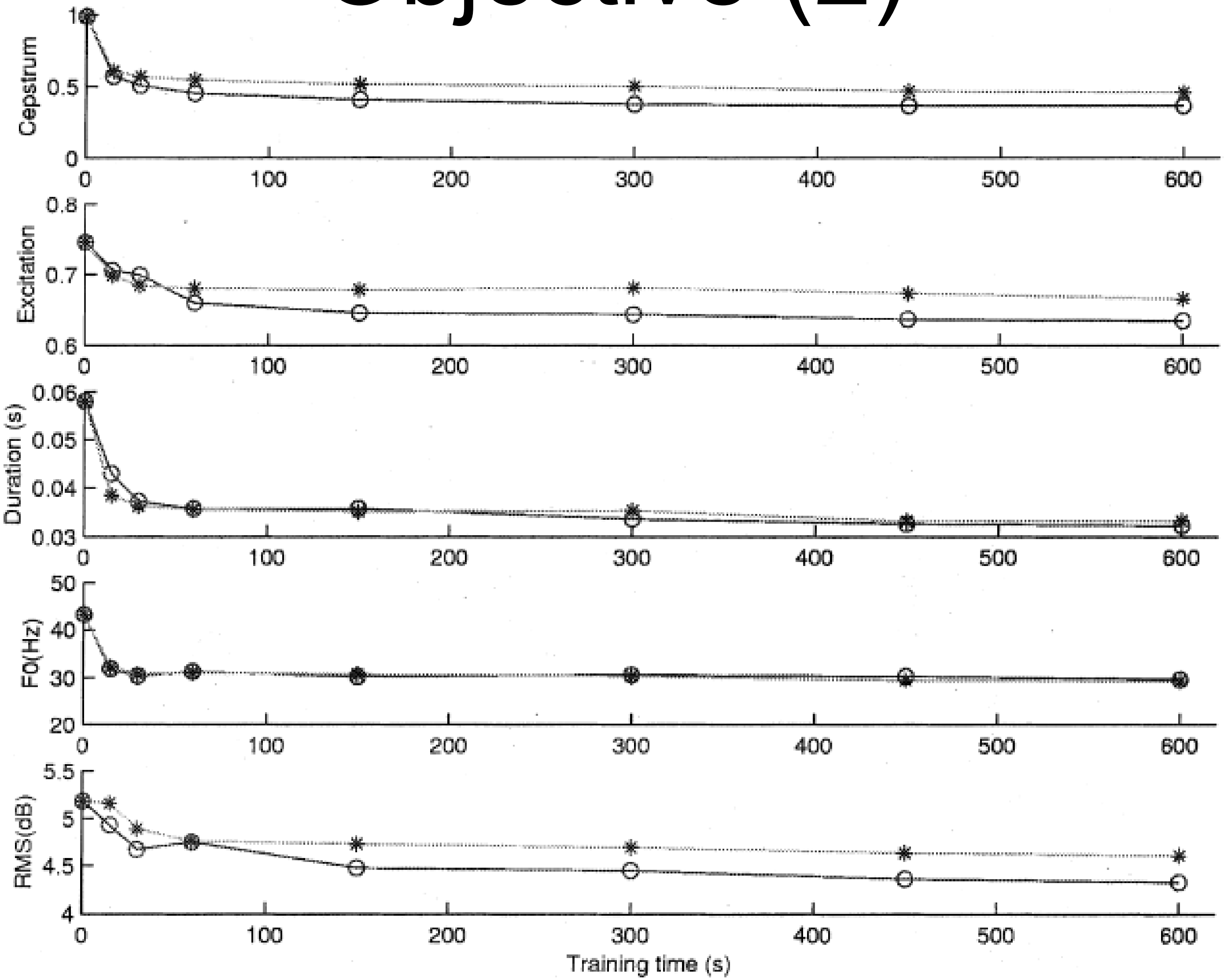


Fig. 8. Performance of the sentence HMM-based and phonetic STASC algorithms in terms of objective measures. Horizontal axis corresponds to the training duration. Vertical axis corresponds to the distance metric between mimic and target utterances in each of the 5 acoustic dimensions considered. The measures at time 0 indicate the average objective measures between the target speaker utterances and unprocessed speech from the source speaker. Dark lines: sentence HMM-based STASC. Light lines: phonetic STASC.

Subjective

- Listening experiments – no cheating
- ABX test
  - 20 stimuli presented
    - A, B listened to; X presented; (2-3 word phrases)
    - “Is X perceptually closer to A or to B in terms of speaker identity”
    - HMM based transformation
    - 100% M-F, 78% M-M
- But is it a garbled mess?

Intelligibility

- 150 short nonsense sentences (prevent inference)
- “Shipping gray paint hands even”
- Phone accuracy of natural, transformed speech compared. Phones retrieved from dictionary
- 93.8% accuracy transformed, 93.4% accuracy natural
  - Target speaker more intelligible?