Modeling Prosody for Speaker Recognition: Why Estimating Pitch May Be a Red Herring

Kornel Laskowski & Qin Jin

Carnegie Mellon University
Pittsburgh PA, USA

28 June, 2010
Features in Speech Processing

ALL FEATURES
Features in Speech Processing
Features in Speech Processing

<table>
<thead>
<tr>
<th>INSTANTANEOUS (&quot;SEGMENTAL&quot;)</th>
<th>SPECTRAL &quot;ENVELOPE&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINE (e.g., HARMONIC) STRUCTURE</td>
<td></td>
</tr>
</tbody>
</table>
Features in Speech Processing

<table>
<thead>
<tr>
<th>Spectral &quot;Envelope&quot;</th>
<th>Fine (e.g., Harmonic) Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral Slope LPCCs</td>
<td>Pitch Voicing Likelihood</td>
</tr>
<tr>
<td>Segment MFCCs</td>
<td>Voicing Classes</td>
</tr>
<tr>
<td>Segment Likelihoods</td>
<td>Harmonic-to-Noise</td>
</tr>
<tr>
<td>Energies</td>
<td>Harmonic-to-Subharmonic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instantaneous (&quot;Segmental&quot;)</th>
<th>Trajectory (&quot;Supra-Segmental&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differences</td>
<td>Differences</td>
</tr>
<tr>
<td>Trends</td>
<td>Trends</td>
</tr>
<tr>
<td>Perturbation Statistics</td>
<td>Perturbation Statistics</td>
</tr>
<tr>
<td>Ngram Posteriors</td>
<td>Ngram Posteriors</td>
</tr>
<tr>
<td>Class Durations</td>
<td>Class Durations</td>
</tr>
</tbody>
</table>

PROSODIC FEATURES

ASR FEATURES

STRUCTURE

FINE (e.g., HARMONIC) SPECTRAL "ENVELOPE"

INSTANTANEOUS ("SEGMENTAL")

DIFFERENCES

TRENDS

PERTURBATION STATISTICS

NGRAM POSTERIORS

CLASS DURATIONS

HARMONIC-TO-NOISE CLASSES

HAMONIC-TO SUBHARMONIC

Laskowski & Jin  ODYSSEY 2010, Brno, Czech Republic
Features in Speech Processing

Prolegomena
Harmonic Structure
Experiments
Analysis
Conclusions
Addenda

Features

- **ASR** features
  - Spectral "envelope" features:
    - Spectral slope
    - LPCCs
    - Segment classes
    - Segment likelihoods
    - Energies
  - Instantaneous ("segmental") features:
    - MFCCs
    - Segment classes
  - Differences features:
    - Trends
    - Perturbation statistics
    - N-gram posteriors
    - Class durations
  - "Prosodic" features:
    - Pitch
    - Voicing likelihood
    - Voicing classes
    - Harmonic-to-noise
    - Harmonic-to-subharmonic
    - Trends
    - Perturbation statistics
    - N-gram posteriors
    - Class durations
Features in Speech Processing

**SPECTRAL "ENVELOPE"**
- Spectral Slope
- LPCCs
- Segment MFCCs
- Segment Likelihoods
- Energies

**FINE (e.g., HARMONIC) STRUCTURE**
- Pitch
- Voicing Likelihood
- Voicing Classes
- Harmonic-to-Noise Likelihood
- Harmonic-to-Subharmonic

**INSTANTANEOUS "PROSODIC" FEATURES**

**DIFFERENCES**
- Trends
- Perturbation Statistics
- N-gram Posteriors
- Class Durations

**TRAJECTORY ("SUPRA-SEGMENTAL")**
- Trends
- Perturbation Statistics
- N-gram Posteriors
- Class Durations

**INSTANTANEOUS STRUCTURE**
- Fine (e.g., Harmonic) Spectral Envelope
- Voicing Likelihood
- Voicing Classes
- Harmonic-to-Noise Likelihood
- Harmonic-to-Subharmonic
Features in Speech Processing

- **SPECTRAL "ENVELOPE"**
  - Spectral Slope
  - LPCCs
  - Segment MFCCs
  - Segment Likelihoods
  - Energies

- **FINE (e.g., HARMONIC) STRUCTURE**
  - Pitch
  - Voicing Likelihood
  - Voicing Classes
  - Harmonic-To-Noise Likelihood
  - Harmonic-To-Subharmonic

- **INSTANTANEOUS"PROSODIC" FEATURES**
  - Differences
  - Trends
  - Perturbation Statistics
  - Ngram Posteriors
  - Class Durations

- **SEGMENTAL"PROSODIC" FEATURES**
  - Differences
  - Trends
  - Perturbation Statistics
  - Ngram Posteriors
  - Class Durations
Pitch Estimation & Processing

- this talk explores what happens inside here
- low-level feature computation

1-best estimate
Pitch Estimation & Processing

- this talk explores what happens inside here
- low-level feature computation

PITCH DETECTION

1-best estimate → N-best estimates
Pitch Estimation & Processing

- this talk explores what happens inside here
- low-level feature computation

1-best estimate

N-best estimates

DYNAMIC PROGRAMMING

1-best estimate

Laskowski & Jin

ODYSSEY 2010, Brno, Czech Republic
this talk explores what happens inside here

low-level feature computation
this talk explores what happens inside here

low-level feature computation

• Pitch Estimation & Processing

1−best estimate

N−best estimates

DYNAMIC PROGRAMMING

1−best estimate

(OPTIONAL) SMOOTHING LINEARIZATION STYLIZATION
this talk explores what happens inside here

low-level feature computation

Pitch Estimation & Processing

1-best estimate → N-best estimates → Dynamic Programming

1-best estimate → (optional) Smoothing Linearization Stylization → (optional) Normalization Conditioning
Pitch Estimation & Processing

- this talk explores what happens inside here
- low-level feature computation

Diagram:

1. **Pitch Detection**
   - 1-best estimate
   - N-best estimates

2. **Dynamic Programming**
   - 1-best estimate

3. **Optional**
   - Smoothing
   - Linearization
   - Stylization

4. **Optional**
   - Normalization
   - Conditioning

5. **Trajectory Model Parameter Estimation**
this talk explores what happens inside here

- low-level feature computation
  - approx. at the level of MFCCs

Laskowski & Jin  
ODYSSEY 2010, Brno, Czech Republic
Pitch Estimation & Processing

- this talk explores what happens inside here
- low-level feature computation
  - approx. at the level of MFCCs
this talk explores what happens inside here

- low-level feature computation
  - approx. at the level of MFCCs
Inside (Per-Frame) Pitch Detection

Essentially a 2-step process:

1. Begin with a source-domain $x$
   - Typically, the short-time FFT

2. Compute the transformed-domain $y = f(x)$
   - Autocorrelation spectrum
   - Real cepstrum
   - Comb filterbank energies
   - And many others

3. Find the supremum of $y$, $F_0 = \arg\max_y y$
Inside (Per-Frame) Pitch Detection

Essentially a 2-step process:

0. begin with a source-domain \( x \)
   - typically, the short-time FFT

1. compute the transformed-domain \( y = f(x) \)
   - autocorrelation spectrum
   - real cepstrum
   - comb filterbank energies
   - and many others

2. find the supremum of \( y \), \( F_0 = \arg \max_y \)
Essentially a 2-step process:

1. begin with a **source-domain** $x$
   - typically, the short-time FFT

2. compute the **transformed-domain** $y = f(x)$
   - autocorrelation spectrum
   - real cepstrum
   - comb filterbank energies
   - and many others

3. find the supremum of $y$, $F_0 = \arg \max y$
Essentially a 2-step process:

0. begin with a source-domain \( x \)
   - typically, the short-time FFT

1. compute the transformed-domain \( y = f(x) \)
   - autocorrelation spectrum
   - real cepstrum
   - comb filterbank energies
   - and many others

2. find the supremum of \( y \), \( F_0 = \arg \max y \)
Outline of this Talk

1. Harmonic Structure Transform
2. Experiment: closed-set classification, 10-second trials
   - matched-multisession, matched-channel conditions
   - contrast with get\_f0-estimated pitch
   - contrast with MFCCs
3. Analysis
   - simulated perturbations
   - spectral envelope ablation
4. Conclusions
Schroeder’s “Harmonic Product Spectrum”

Given a continuous short-time spectrum $S(f)$, Schroeder proposed

$$\Sigma(f) = 20 \log_{10} \sum_{n=1}^{N} |S(n f)|$$


Schroeder’s “Harmonic Product Spectrum”

Given a continuous short-time spectrum $S(f)$, Schroeder proposed

$$\Sigma(f) = 20 \log_{10} \sum_{n=1}^{N} |S(nf)|$$

Noll dubbed this “harmonic compression”. (Distinctly non-linear.)


Schroeder’s “Harmonic Product Spectrum”

Given a continuous short-time spectrum \( S(f) \), Schroeder proposed

\[
\Sigma(f) = 20 \log_{10} \sum_{n=1}^{N} |S(nf)|
\]

Noll dubbed this “harmonic compression”. (Distinctly non-linear.)


Dirac Comb Filterbank

- the alternative: design a continuous-frequency **comb filter**
  - for each candidate fundamental frequency of interest

- no “compression difficulties” during discretization
  - filtering is a linear operation
  - here: each filter is defined over 300–8000 Hz
  - a set of such comb filters (here: 400) yields a **filterbank**
  - from 50 Hz to 450 Hz, spaced 1 Hz apart

---

Dirac Comb Filterbank

- the alternative: design a continuous-frequency **comb filter**
  - for each candidate fundamental frequency of interest

![Diagram of Dirac Comb Filterbank]

- no “compression difficulties” during discretization
  - filtering is a **linear** operation

- here: each filter is defined over 300–8000 Hz
- a set of such comb filters (here: 400) yields a **filterbank**
  - from 50 Hz to 450 Hz, spaced 1 Hz apart

---

Dirac Comb Filterbank

- the alternative: design a continuous-frequency **comb filter**
  - for each candidate fundamental frequency of interest

![Diagram of Dirac Comb Filterbank](image)

- no “compression difficulties” during discretization
  - filtering is a **linear** operation

- here: each filter is defined over 300–8000 Hz
  - a set of such comb filters (here: 400) yields a **filterbank**
    - from 50 Hz to 450 Hz, spaced 1 Hz apart

---

the alternative: design a continuous-frequency **comb filter**
for each candidate fundamental frequency of interest

- no “compression difficulties” during discretization
- filtering is a **linear** operation
- here: each filter is defined over 300–8000 Hz
- a set of such comb filters (here: 400) yields a **filterbank**
  - from 50 Hz to 450 Hz, spaced 1 Hz apart

---

Discrete Comb Filterbank

- in software, have a **discrete FFT**
  - sampling frequency: 16 kHz
  - frame size: 32 ms
  - 257 discrete real, non-negative frequencies (bins)

Here: assume each comb tooth is triangular

Riemann sample the triangular comb filter

Note: the resulting discrete comb filters are **not** harmonic

---

Discrete Comb Filterbank

- in software, have a **discrete FFT**
  - sampling frequency: 16 kHz
  - frame size: 32 ms
  - 257 discrete real, non-negative frequencies (bins)

- here: assume each comb tooth is triangular
- Riemann sample the triangular comb filter

- note: the resulting discrete comb filters are **not** harmonic

---

Discrete Comb Filterbank

- in software, have a **discrete** FFT
  - sampling frequency: 16 kHz
  - frame size: 32 ms
  - 257 discrete real, non-negative frequencies (bins)

![Graph showing discrete comb filter](image)

1. here: assume each comb tooth is triangular
2. Riemann sample the triangular comb filter

---

**Note:** the resulting discrete comb filters are **not** harmonic

---

Discrete Comb Filterbank

- in software, have a **discrete FFT**
  - sampling frequency: 16 kHz
  - frame size: 32 ms
  - 257 discrete real, non-negative frequencies (bins)

1. here: assume each comb tooth is triangular
2. Riemann sample the triangular comb filter

- note: the resulting discrete comb filters are **not** harmonic

---

Discrete Comb Filterbank

- in software, have a **discrete** FFT $x$
  - sampling frequency: 16 kHz
  - frame size: 32 ms
  - 257 discrete real, non-negative frequencies (bins)

1. here: assume each comb tooth is triangular
2. Riemmann sample the triangular comb filter

- note: the resulting discrete comb filters are **not** harmonic

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- the discrete comb filterbank forms a matrix $\mathbf{H}$
  1. its application to FFT $\mathbf{x}$ is a matrix multiplication $(\mathbf{H}^T \mathbf{x})$
  2. we take the logarithm at the output (as for Mel energies)
  3. and subtract the log-energy found everywhere else in $\mathbf{x}$

$$\hat{\mathbf{H}} = 1 - \mathbf{H}$$
$$\mathbf{y} = \log (\mathbf{H}^T \mathbf{x}) - \log (\hat{\mathbf{H}}^T \mathbf{x})$$

- $\mathbf{y}$ is effectively a vector of harmonic-to-noise ratios (HNRs)

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- the discrete comb filterbank forms a matrix $\mathbf{H}$
  - its application to FFT $\mathbf{x}$ is a matrix multiplication $(\mathbf{H}^T \mathbf{x})$
  - we take the logarithm at the output (as for Mel energies)
  - and subtract the log-energy found everywhere else in $\mathbf{x}$

$$\tilde{\mathbf{H}} \equiv 1 - \mathbf{H}$$

$$\mathbf{y} = \log (\mathbf{H}^T \mathbf{x}) - \log (\tilde{\mathbf{H}}^T \mathbf{x})$$

- $\mathbf{y}$ is effectively a vector of harmonic-to-noise ratios (HNRs)

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- the discrete comb filterbank forms a matrix $\mathbf{H}$
- its application to FFT $\mathbf{x}$ is a matrix multiplication $(\mathbf{H}^T \mathbf{x})$
- we take the logarithm at the output (as for Mel energies)
- and subtract the log-energy found everywhere else in $\mathbf{x}$

\[
\tilde{\mathbf{H}} \equiv 1 - \mathbf{H},
\]
\[
y = \log (\mathbf{H}^T \mathbf{x}) - \log (\tilde{\mathbf{H}}^T \mathbf{x})
\]

- $y$ is effectively a vector of harmonic-to-noise ratios (HNRs)

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- the discrete comb filterbank forms a matrix $\mathbf{H}$
  - its application to FFT $\mathbf{x}$ is a matrix multiplication $(\mathbf{H}^T\mathbf{x})$
  - we take the logarithm at the output (as for Mel energies)
  - and subtract the log-energy found everywhere else in $\mathbf{x}$

\[
\tilde{\mathbf{H}} \equiv 1 - \mathbf{H} \\
\mathbf{y} = \log (\mathbf{H}^T\mathbf{x}) - \log (\tilde{\mathbf{H}}^T\mathbf{x})
\]

- $\mathbf{y}$ is effectively a vector of harmonic-to-noise ratios (HNRs)

Normalization of Harmonic Energy by Non-Harmonic Energy

- The discrete comb filterbank forms a matrix $H$
  - its application to FFT $x$ is a matrix multiplication $(H^T x)$
  - we take the logarithm at the output (as for Mel energies)
  - and subtract the log-energy found everywhere else in $x$

\[
\tilde{H} \equiv 1 - H
\]

\[
y = \log (H^T x) - \log (\tilde{H}^T x)
\]

- $y$ is effectively a vector of harmonic-to-noise ratios (HNRs)

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- The discrete comb filterbank forms a matrix $\mathbf{H}$.
  - Its application to FFT $\mathbf{x}$ is a matrix multiplication $(\mathbf{H}^T \mathbf{x})$.
  - We take the logarithm at the output (as for Mel energies).
  - And subtract the log-energy found everywhere else in $\mathbf{x}$.

$$
\tilde{\mathbf{H}} \equiv 1 - \mathbf{H}
$$

$$
\mathbf{y} = \log (\mathbf{H}^T \mathbf{x}) - \log (\tilde{\mathbf{H}}^T \mathbf{x})
$$

- $\mathbf{y}$ is effectively a vector of harmonic-to-noise ratios (HNRs).

---

Normalizing Harmonic Energy by Non-Harmonic Energy

- The discrete comb filterbank forms a matrix $H$
  - Its application to FFT $x$ is a matrix multiplication $(H^T x)$
  - We take the logarithm at the output (as for Mel energies)
  - And subtract the log-energy found everywhere else in $x$

\[
\tilde{H} \equiv 1 - H
\]
\[
y = \log (H^T x) - \log (\tilde{H}^T x)
\]

- $y$ is effectively a vector of harmonic-to-noise ratios (HNRs)

---

Feature Vector Decorrelation

- the elements of \( y \) are correlated
- transform \( y \) by
  1. subtracting global mean
  2. orthogonalizing (rotating) via data-dependent \( \mathcal{F}^{-1}_{\text{CORR}} \)
  3. truncating non-positive eigenvalue dimensions
- yields the *harmonic structure cepstral coefficients*

\[
\text{HSCC} = \mathcal{F}^{-1}_{\text{CORR}} \left( \log \left( H^T x \right) - \log \left( \tilde{H}^T x \right) \right)
\]

\[
= \mathcal{F}^{-1}_{\text{CORR}} \left( \log \left( H^T x \right) \right) - \mathcal{F}^{-1}_{\text{CORR}} \left( \log \left( \tilde{H}^T x \right) \right)
\]

- two options for \( \mathcal{F}^{-1}_{\text{CORR}} \):
  1. PCA: conditionally independent of labels
  2. LDA: conditioned on labels
Similarities with the Mel Filterbank, $M$

\[ \text{MFCC} = \mathcal{F}_{\text{COS-II}}^{-1} \left( \log \left( M^T x \right) \right) - \langle \text{normalization term} \rangle \]

\[ \text{HSCC} = \mathcal{F}_{\text{CORR}}^{-1} \left( \log \left( H^T x \right) \right) - \langle \text{normalization term} \rangle \]

columns of $M$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

columns of $H$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Similarities with the FFV Spectrum

HST (here)  FFV (previous work)
Similarities with the FFV Spectrum

HST (here)  FFV (previous work)

frame FFT
\( x_t \)
**Similarities with the FFV Spectrum**

**HST (here)**

- frame FFT $x_t$
- idealized FFT (comb filter $h$)
- $f_h[i - 1]$
- $f_h[i]$  
- $f_h[i + 1]$

**FFV (previous work)**

Laskowski & Jin  
ODYSSEY 2010, Brno, Czech Republic
Similarities with the FFV Spectrum

HST (here)  FFV (previous work)

frame FFT $x_t$  idealized FFT (comb filter $h$)

$f_h[i - 1]$  $f_h[i]$  $f_h[i + 1]$
Similarities with the FFV Spectrum

HST (here)  FFV (previous work)

frame FFT \(x_t\)  idealized FFT (comb filter \(h\))

\[ f_h[i - 1] \]
\[ f_h[i] \]
\[ f_h[i + 1] \]

as a function of \(i\)
Similarities with the FFV Spectrum

HST (here)

frame FFT $x_t$

idealized FFT (comb filter $h$)

$\text{frame FFT } x_t$

FFV (previous work)

$\text{frame FFT } x_t$

$\text{frame FFT } x_{t-1}$

as a function of $i$
Similarities with the FFV Spectrum

**HST (here)**

- frame FFT $\mathbf{x}_t$
- idealized FFT (comb filter $\mathbf{h}$)
- $f_h[i - 1]$
- $f_h[i]$
- $f_h[i + 1]$

![Diagram showing HST (here)]

**FFV (previous work)**

- frame FFT $\mathbf{x}_t$
- dilated $\mathbf{x}_{t-1}$
- frame FFT $\mathbf{x}_{t-1}$
- $2^{\alpha_i}, i < 0$
- $2^{\alpha_i}, i > 0$

![Diagram showing FFV (previous work)]

Laskowski & Jin | ODYSSEY 2010, Brno, Czech Republic
--- | ---
12/29
Similarities with the FFV Spectrum

**HST (here)**

- `frame FFT ` $\mathbf{x}_t$
- `idealized FFT (comb filter h)`

```
fh[i - 1]  fh[i]  fh[i + 1]
```

as a function of $i$

**FFV (previous work)**

- `frame FFT ` $\mathbf{x}_t$
- `dilated $\mathbf{x}_{t-1}$`
- `frame FFT ` $\mathbf{x}_{t-1}$

```
2^{ai}, i < 0  2^{ai}, i > 0
```

Laskowski & Jin  ODYSSEY 2010, Brno, Czech Republic
Similarities with the FFV Spectrum

**HST (here)**

- Frame FFT $x_t$
- Idealized FFT (comb filter $h$)

\[
\begin{align*}
  f_h[i - 1] \\
  f_h[i] \\
  f_h[i + 1]
\end{align*}
\]

as a function of $i$

**FFV (previous work)**

- Frame FFT $x_t$
- Dilated $x_{t-1}$
- Frame FFT $x_{t-1}$

\[
\begin{align*}
  2^{\alpha i}, & \quad i < 0 \\
  2^{\alpha i}, & \quad i > 0
\end{align*}
\]

as a function of $i$
Experiments: Data

- WSJ: LDC CSR-I (WSJ0) & LDC CSR-II (WSJ1)
- 102 female (♀) speakers, 95 male (♂) speakers
- closed-set classification, 10-second trials
  - TrainSet: 5 minutes
  - DevSet: 3 minutes, # trials: 1775 (♀) and 1660 (♂)
  - TestSet: 3 minutes, # trials: 1510 (♀) and 1412 (♂)
- matched channel, Sennheiser HMD414 (.wv1)
- matched multi-session:
  - 4–20 sessions per speaker
  - Train-/Dev-/Test- Sets drawn from most sessions
F₀/GMM Baseline System (not in paper)

1. extract $F_0$ using `get_f0`
   - Snack Sound Toolkit: ESPS, default settings
   - note: relies on dynamic programming

2. transform voiced frames to $\log_2$ domain
   - ignore unvoiced frames

<table>
<thead>
<tr>
<th>$N_G$</th>
<th>Female DevSet</th>
<th>Female EvalSet</th>
<th>Male DevSet</th>
<th>Male EvalSet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.31</td>
<td>12.71</td>
<td>17.15</td>
<td>17.41</td>
</tr>
<tr>
<td>8</td>
<td>17.48</td>
<td>17.94</td>
<td>25.91</td>
<td>27.62</td>
</tr>
<tr>
<td>16</td>
<td>16.70</td>
<td>17.44</td>
<td>26.21</td>
<td>27.44</td>
</tr>
<tr>
<td>256</td>
<td><strong>17.62</strong></td>
<td><strong>18.36</strong></td>
<td>25.91</td>
<td>26.02</td>
</tr>
</tbody>
</table>
# HSCC System Configuration

<table>
<thead>
<tr>
<th>Parameter/Aspect</th>
<th>HSCC System</th>
</tr>
</thead>
<tbody>
<tr>
<td>pre-emphasis</td>
<td>no</td>
</tr>
<tr>
<td>framing</td>
<td>8ms/32ms</td>
</tr>
<tr>
<td>window</td>
<td>Hann</td>
</tr>
<tr>
<td>$N_D$</td>
<td>to optimize</td>
</tr>
<tr>
<td>$N_G$</td>
<td>to optimize</td>
</tr>
<tr>
<td>UBM</td>
<td>no</td>
</tr>
<tr>
<td>SAD</td>
<td>no</td>
</tr>
</tbody>
</table>
HSCC Vector Rotation and Truncation

- Pick number of dimensions $N_D$
  - Set number of (diagonal-covariance) Gaussians $N_G = 1$
  - Train PCA, LDA on TRAINSET
  - Choose $N_D$ to maximize accuracy on DEVSET

![Graph showing accuracy vs. $N_D$ for PCA and LDA](image-url)
with \( N_D \) fixed, find \( N_G \) to maximize DEVSET accuracy → 256

<table>
<thead>
<tr>
<th>System</th>
<th>Female, ♀</th>
<th>Male, ♂</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEV TEST</td>
<td>DEV TEST</td>
</tr>
<tr>
<td>get_f0</td>
<td>17.62 18.36</td>
<td>26.21 27.44</td>
</tr>
<tr>
<td>HSCC/LDA</td>
<td>99.72 99.87</td>
<td>99.70 99.65</td>
</tr>
</tbody>
</table>

1. there **is** speaker-discriminative information in the transformed-domain, **beyond the** arg max
   - discarding it leads to much worse performance
2. improving arg max estimation appears unnecessary
   - arg max estimation = pitch estimation
### Contrastive MFCC/GMM System

<table>
<thead>
<tr>
<th>Parameter/Aspect</th>
<th>HSCC System</th>
<th>MFCC System</th>
</tr>
</thead>
<tbody>
<tr>
<td>pre-emphasis</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>framing</td>
<td>8ms/32ms</td>
<td>8ms/32ms</td>
</tr>
<tr>
<td>window</td>
<td>Hann</td>
<td>Hamming</td>
</tr>
<tr>
<td>$N_D$</td>
<td>52-53 (opt)</td>
<td>20</td>
</tr>
<tr>
<td>$N_G$</td>
<td>256 (opt)</td>
<td>256 (opt)</td>
</tr>
<tr>
<td>UBM</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>SAD</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
## Results II

<table>
<thead>
<tr>
<th>System</th>
<th>Female, $\varphi$</th>
<th>Male, $\sigma^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dev</td>
<td>Test</td>
</tr>
<tr>
<td>HSCC/LDA</td>
<td>99.72</td>
<td>99.87</td>
</tr>
<tr>
<td>MFCC</td>
<td>98.66</td>
<td>99.27</td>
</tr>
<tr>
<td>MFCC/LDA</td>
<td>98.71</td>
<td>99.27</td>
</tr>
<tr>
<td>HSCC/LDA $\oplus$ MFCC</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

1. HSCC performance comparable to MFCC performance
   - in these experiments, always better
2. equal-weight score-level fusion can yield improvement
   - HSCC and MFCC appear complementary
Some Perturbations

Evaluate several types of perturbation:

1. source-domain frequency range ablation
   - low frequency (LF) cutoff
   - high frequency (HF) cutoff

2. transformed-domain frequency resolution

3. source-domain spectral envelope ablation

Simplify analysis suite by:

- using \( N_G = 1 \) diagonal-covariance Gaussian per speaker
- computing accuracy \( \text{DevSet} \) only
- plotting accuracy as a function of \( N_D \)
Source-Domain Low Frequency (LF) Range

- modify the low-frequency cutoff for source-domain (FFT) x
Source-Domain Low Frequency (LF) Range

- modify the low-frequency cutoff for source-domain (FFT) x
Source-Domain Low Frequency (LF) Range

- modify the low-frequency cutoff for source-domain (FFT) $x$

Accuracy vs. $N_D$

- $0-8000$: $92.00$ (32)
- $300-8000$: $95.94$ (52)
- $600-8000$: $95.38$ (43)

- $0-8000$: $97.11$ (36)
- $300-8000$: $98.67$ (53)
- $600-8000$: $97.71$ (28)
Source-Domain High Frequency (HF) Range

- modify the high-frequency cutoff for source-domain (FFT)

300 Hz 8 kHz
Source-Domain High Frequency (HF) Range

- modify the high-frequency cutoff for source-domain (FFT) x

![Frequency Cutoff Diagram]

- 0 Hz to 2 kHz
- 300 Hz to 4 kHz
- 300 Hz to 8 kHz
Source-Domain High Frequency (HF) Range

- modify the high-frequency cutoff for source-domain (FFT) $x$

![Graph showing accuracy vs. $N_D$ for different frequency ranges for male and female voices.](image)
Transformed-Domain Frequency Resolution

- modify the resolution of the transformed-domain $y$

![Graph showing transformed-domain frequency resolution with 400 filters, 1.0 Hz apart.](image)

400 filters
1.0 Hz apart
Transformed-Domain Frequency Resolution

- modify the resolution of the transformed-domain $y$

- 200 filters 2.0 Hz apart
- 400 filters 1.0 Hz apart
- 800 filters 0.5 Hz apart
Transformed-Domain Frequency Resolution

- modify the resolution of the transformed-domain $y$

![Graph showing accuracy vs. $N_D$ for different frequencies and types](image)

- Female
  - $0.5\text{Hz}$: 97.63 (52)
  - $1.0\text{Hz}$: 95.94 (52)
  - $2.0\text{Hz}$: 93.80 (38)

- Male
  - $0.5\text{Hz}$: 99.04 (40)
  - $1.0\text{Hz}$: 98.67 (53)
  - $2.0\text{Hz}$: 96.87 (55)
Source-Domain Spectral Envelope Ablation

- lifter the low-quefrency components of source-domain (FFT) x
- low-order CCs approximate low-order MFCCs

lifter 0 CCs
Source-Domain Spectral Envelope Ablation

- lifter the low-quefrency components of source-domain (FFT) \( \times \)
- low-order CCs approximate low-order MFCCs

\[
\text{Laskowski & Jin  
ODYSSEY 2010, Brno, Czech Republic}
\]
Source-Domain Spectral Envelope Ablation

- lifter the low-quefrency components of source-domain (FFT) ×
- low-order CCs approximate low-order MFCCs

Accuracy

![Graph showing accuracy vs. ND for male and female voices with different CC settings.](image-url)
Analysis Findings

- HSCC representation appears to be robust to perturbation
  - low-frequency source-domain range (♀: 4%, ♂: 1.5%)
  - high-frequency source-domain range (♀: 4%, ♂: 5%)
  - transformed domain resolution (♀: 4%, ♂: 2%)
  - source-domain envelope ablation (♀: 2.5%, ♂: 1.5%)
- generally, performance for ♀ speakers more sensitive
- even under perturbed conditions, vastly outperform the system based on pitch alone
- not known how a pitch tracker would perform
Summary of Findings

1. Information available to (but discarded by) (some) pitch trackers is valuable.

2. HSCC performance is comparable to MFCC performance.

3. HSCC information is complimentary to MFCC information.

4. HSCC modeling is as easy as MFCC modeling.
The presented evidence suggests:

1. should not invest time in improving estimation of the transformed-domain arg max (i.e., pitch)
   - simply model the entire transformed-domain

2. if require pitch for other ("high-level") features
   - should not discard transformed-domain following arg max estimation

3. using the entire transformed-domain may lead to a paradigmatic shift in the modeling of prosody
Of Immediate Interest ... 

1. don’t know how the HSCC vector compares to other “instantaneous” prosody vectors
2. don’t know how the HSCC vector performs under session, channel, distance, or vocal effort mismatch conditions
3. other classifiers might be better-suited to the size of the transformed-domain (SVMs, etc.)
4. existing prosody systems employ high-level features
   - first-, second-, $N$th-order differences
   - modulation spectrum
5. would prefer data-independent feature rotation/compression
   - would significantly improve understanding
   - would permit UBMing
   - would allow use in large-dataset tasks (e.g., NIST SRE)
Thank You!

This work was particularly inspired by:


Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
estimate the FFV spectrum $g[\rho]$
- estimate the power spectra $F_L$ and $F_R$
- dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
- dot product with undilated $F_L$
- repeat for a continuum of $\rho$ values

pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum \( g[\rho] \)
  - estimate the power spectra \( F_L \) and \( F_R \)
  - dilate \( F_R \) by a factor \( 2^\rho, \rho > 0 \)
  - dot product with undilated \( F_L \)
  - repeat for a continuum of \( \rho \) values

- pass \( g(\rho) \) through a filterbank to yield \( G \in \mathbb{R}^7 \)
- decorrelate \( G \)
estimate the FFV spectrum $g[\rho]$

- estimate the power spectra $F_L$ and $F_R$
- dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
- dot product with undilated $F_L$
- repeat for a continuum of $\rho$ values

pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$

decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
  - decorrelate $G$
Fundamental Frequency Variation

- estimate the FFV spectrum $g[\rho]$
  - estimate the power spectra $F_L$ and $F_R$
  - dilate $F_R$ by a factor $2^\rho$, $\rho > 0$
  - dot product with undilated $F_L$
  - repeat for a continuum of $\rho$ values

- pass $g(\rho)$ through a filterbank to yield $G \in \mathbb{R}^7$
- decorrelate $G$
Fundamental Frequency Variation (3)

\[ \rho = 2^{-0.0342} = 0.9766 \]
leave left FFT as is

dilate right FFT by \( \rho \)

\[ \rho = 2^0 = 1 \]
leave left FFT as is
leave right FFT as is

\[ \rho = 2^{+0.0342} = 1.0240 \]
dilate left FFT by \( \rho \)
leave right FFT as is

\[ g(\rho) = 0.0261 \]
\[ g(\rho) = 0.2299 \]
\[ g(\rho) = 0.6877 \]
Fundamental Frequency Variation (3)
### Some Distant Numbers?

<table>
<thead>
<tr>
<th></th>
<th>EvalSet1 (Sess Mat)</th>
<th>EvalSet2 (Sess Mis)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chan</td>
<td>Chan</td>
</tr>
<tr>
<td></td>
<td>Mat</td>
<td>Mis</td>
</tr>
<tr>
<td>MFCC</td>
<td>100.0</td>
<td>95.2</td>
</tr>
<tr>
<td>$HSCC_{old}$</td>
<td>100.0</td>
<td>67.0</td>
</tr>
<tr>
<td>$HSCC_{new}$</td>
<td>100.0</td>
<td>78.3</td>
</tr>
<tr>
<td>err (%rel)</td>
<td>0</td>
<td>34.2</td>
</tr>
</tbody>
</table>

**Table:** Classification accuracy (in %) using several different feature types, including the improved harmonic structure cepstral coefficients $HSCC_{new}$, in matched (“Mat”) and mismatched (“Mis”) session (“Sess”) and channel (“Chan”) conditions. “err (%rel)” indicates the relative reduction of error, in percent, from $HSCC_{old}$ to $HSCC_{new}$. 
What Do HSCCs Represent?