Speech Processing 15-492/18-492

Voice Conversion
Voice Conversion

- **Live (or offline)**
  - Convert an existing voice to another
  - Use only a small amount of target speech

- **Uses:**
  - Synthesis without collecting lots of data
  - Disguising voices
  - Emotional voices without full synthesis support

- **Also called**
  - Voice transformation, Voice morphing
Voice Identity

What makes a voice identity

- **Lexical Choice**:
  - Woo-hoo,
  - I pity the fool ... 
- Phonetic choice
- Intonation and duration
- Spectral qualities (vocal tract shape)
- Excitation
Voice Conversion techniques

- **Full ASR and TTS**
  - Much too hard to do reliably

- **Codebook transformation**
  - ASR HMM state to HMM state transformation

- **GMM based transformation**
  - Build a mapping function between frames
Learning VC models

- **First need to get parallel speech**
  - Source and Target say same thing
  - Use DTW to align (in the spectral domain)
  - Trying to learn a functional mapping
  - 20-50 utterances

- **“Text-independent” VC**
  - Means no parallel speech available
  - Use some form of synthesis to generate it
VC Training process

- Extract F0, power and MFCC from source and target utterances
- DTW align source and target
- Loop until convergence
  - Build GMM to map between source/target
  - DTW source/target using GMM mapping
VC Training process

Source F0 → log → Compute Means And Std. Devs.
Target F0 → log → Compute Means And Std. Devs.

Source Speaker Filter Features → Add Dynamic Features → Power Threshold → DTW → Train GMM W/EM

Target Speaker Filter Features → Add Dynamic Features → Power Threshold

Iterate
VC Run-time

Source Speech -> Power -> F0 -> log scale z-map -> MLSA Filter

Source Speech -> Filter Features -> GMM map -> MLPG
- **Festvox GMM transformation suite (Toda)**

<table>
<thead>
<tr>
<th></th>
<th>awb</th>
<th>bdl</th>
<th>jmk</th>
<th>slt</th>
</tr>
</thead>
<tbody>
<tr>
<td>awb</td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
</tr>
<tr>
<td>bdl</td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
</tr>
<tr>
<td>jmk</td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
</tr>
<tr>
<td>slt</td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
<td><img src="speaker.png" alt="Sound" /></td>
</tr>
</tbody>
</table>
VC in Synthesis

- **Can be used as a post filter in synthesis**
  - Build `kal_diphone` to target VC
  - Use on all output of `kal_diphone`

- **Can be used to convert a full DB**
  - Convert a full db and rebuild a voice
Style/Emotion Conversion

- **Unit Selection (or SPS)**
  - Require lots of data in desired style/emotion

- **VC technique**
  - Use as filter to main voice (same speaker)
  - Convert neutral to angry, sad, happy …
Can you say that again?

- Voice conversion for speaking in noise
- Different quality when you repeat things
- Different quality when you speak in noise
  - Lombard effect (when very loud)
  - “Speech-in-noise” in regular noise
Collect data

- Randomly play noise in person’s ears
- Normal
- In Noise

Collect 500 of each type

Build VC model

- Normal -> in-Noise

Actually

- Spectral, duration, f0 and power differences
Synthesis in Noise

- **For bus information task**
- **Play different synthesis information utts**
  - With SIN synthesizer
  - With SWN synthesizer
  - With VC (SWN->SIN) synthesizer
- **Measure their understanding**
  - SIN synthesizer better (in Noise)
  - SIN synthesizer better (without Noise for elderly)
Transterpolation

- Incrementally transform a voice X%
  - BDL-SLT by 10%
  - SLT-BDL by 10%
- Count when you think it changes from M-F
- Fun but what are the uses …
De-identification

- **Remove speaker identity**
  - But keep it still human like

- **Health Records**
  - HIPAA laws require this
  - Not just removing names and SSNs

- **Remove identifiable properties**
  - Use Voice conversion to remove spectral
  - Use F0/duration mapping to remove prosodic
  - Use ASR/MT techniques to remove lexical
VC and SPS

- Becoming closely related
  - Small amount of target speaker
  - Use larger background models
Cross Lingual Voice Conversion

- Use phonetic mapping synthesis
  - Sounds like very accented speech
- Use VC to convert the output
  - Require only small amount of target language