Exploring different voice conversion approaches and their applications

Qualifying Exam

Seyed Hamidreza Mohammadi

Center for Spoken Language Understanding (CSLU)
Oregon Health & Science University (OHSU)
Portland, Oregon, USA

June 9, 2014

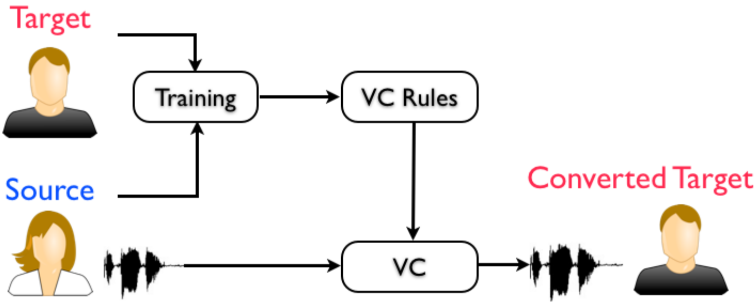
Voice Conversion

Voice Conversion (VC)

process the speech of a *source* speaker to sound like a *target* speaker

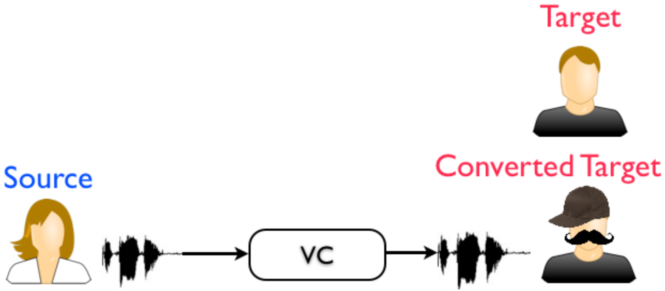
- Applications
  - personalized TTS
    - for individuals with disabilities
    - message readers with custom/sender identities
  - movie dubbing
  - interpretive services by human or machine
- Important criteria
  - speaker similarity (evaluates conversion accuracy)
  - speech quality (evaluates speech naturalness)

Voice Conversion



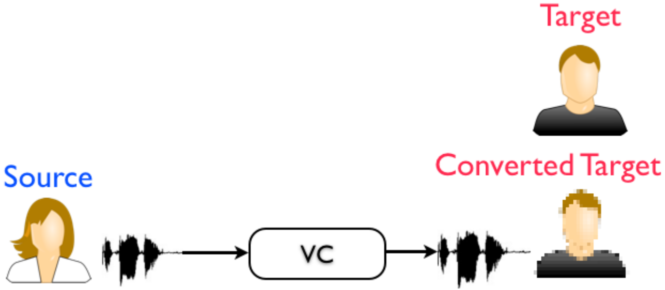
VC accuracy

conversion accuracy (speaker similarity)



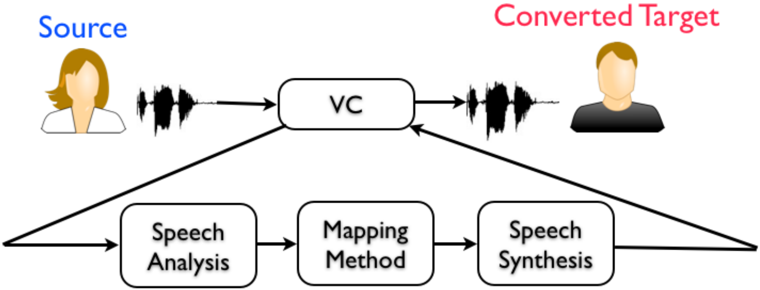
VC quality

speech quality



VC components

Voice conversion components



Analysis/Synthesis models

- Speech Analysis/Syntheis models:
  - source-filter model of speech:
    - assume independence between source (vocal folds) and filter (vocal tract)
    - source: A pulse/noise model or mixed-excitation
    - filter: Linear Predictive Coding (LPC), Mel-Cepstral Coding (MCEP)
    - Lower quality, but less parameters (more robust to model)
  - sinusoidal model of speech:
    - assume speech is an addition of sinusoids
    - pitch-synchronous analysis
    - e.g. Sinusoidal Coding, Harmonic+Noise Coding
    - Higher quality, bur more parameters (harder to model)

Linear Predictive Coding

- Linear predictive analysis is one of the most widely used speech analysis techniques.
- The linear system can be modeled by an all-pole system:

$$H(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)}$$

- we can write the following system

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Ge[n]$$

- the linear predictive signal:

$$\tilde{s}[n] = \sum_{k=1}^{p} a_k s[n-k]$$

- compute linear coefficients by minimizing the following using Levinson-Durbin recursion:

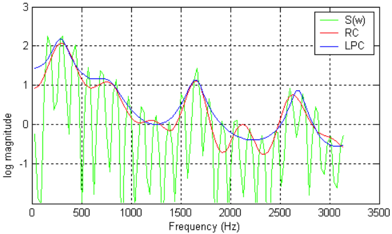
$$E = \frac{1}{N} \sum_{n=1}^{N} (s[n] - \tilde{s}[n])^2$$

Linear Predictive Coding

- provides an accurate model of the speech
- relative high speed of computation.
- Line Spectral Frequencies (LSFs)
  - are a completely reversible representation of LPCs
  - always produce stable filters
  - good quantization properties
- For synthesis, a voicing flag is computed.
  - For voiced frames, a periodic pulse with a period of $1/f_0$ is used.
  - For noisy frames, a random noise signal is used.

Mel-Cepstrum Coding

- LP Disadvantages
  - LP analysis, models the peaks (poles) accurately but not the valleys (zeros).
  - LSFs suffer from coefficient mismatch and also they are highly correlated.
- Cepstrum:
  - Cepstrum puts equal weigh on modeling poles and zeros.
  - Cepstrum coefficients are also uncorrelated.



http://www.advsolned.com/example_speech_coding.html

Mel-Cepstrum Coding

- Simple cepstrum is computed as the $c[m] = ifft(log(abs(fft(s[n]))))$
- The synthesis filter $H(z) = \exp \sum_{m=0}^{M} c_m z^{-m}$
- Additional considerations for taking mel-scaling into account
- MLSA filter is used to create the synthesis filter
- filter not easily realizable -> approximation (called Padé approximation) is used to build the synthesis filter in a few iterations
- In this study, we use the mel scaled cepstrum (MCEP) using SPTK toolkit.

Sinusoidal Analysis/Synthesis

- The speech waveform = sum of sine waves (with different frequencies and amplitudes)

$$s[n] = \sum_{l=1}^{L} A_l \sin \omega_0 l + \phi_l$$

- A_l: amplitude
- ϕ_l: phase.
- ω_0: fundamental frequency, is estimated beforehand.
- Analysis is pitch-synchronous
- Very high quality
- Disadvantages:
  - High number of parameters, hard to model using statistical models
  - Phase is hard to model

VC Mapping Problem

- Let $x = [x_1, ..., x_t]$ be D-dimensional source
- and $y = [y_1, ..., y_t]$ be target feature vectors
- We want to build some VC rules that is able to convert any given source feature vector to target feature vector

Mapping approaches

1. Generative
   - compact parametrization of speech (LSF, MCEP)
   - direct mapping from input to output parameters
   - quality is limited by parametric vocoder
2. Transmutative
   - high-fidelity speech model (Sinusoidal)
   - difficult to train satisfactory direct mapping
     - high-dimensional feature space
     - especially for very small training sets
   - instead, use a constrained mapping
     - change prominent characteristics while leaving less-prominent characteristics unmodified

Generative approach — examples

- $\mathcal{F}(x; \lambda^*) = \hat{y}$
- Implementations of $\mathcal{F}$:
  - Frame Selection (FS) [Sundermann 2006, Dutoit 2007]
  - Gaussian Mixture Models (GMM) [Stylianou 1998, Kain 98, Toda 2007]
  - Artificial Neural Networks (ANN) [Desai 2010]
- Speech model / feature types:
  - line spectral frequencies (LSF) [Kain 1998]
  - mel-cepstrum (MCEP) [Tokuda 1995]

Problem

Speech quality is limited by the parametric vocoder

Frame Selection

- The Frame Selection (FS) algorithm is a memory-based approach.
- Similar to unit-selection in the context of Text-to-Speech (TTS) systems
- The FS algorithm employs two cost functions: target cost and concatenation cost.
- For each input vector, we select the K-nearest source training feature vectors based on their target cost ($K = 10$ in this study).
- A Viterbi algorithm then selects the best sequence of target training feature vectors, minimizing a combined target and concatenation cost.

Joint Density GMM

- Let $z_t = [z_1, ..., z_t]$ be the joint feature vector where $z_t = [x_t^\top, y_t^\top]^\top$.
- the joint density of source and target feature vectors is modeled using a Gaussian Mixture Model (GMM).

$$P(z_t|\lambda^z) = \sum_{m=1}^{M} w_m N(z_t; \mu_m^z, \Sigma_m^z)$$

- Conditional Probability

$$P(y_t|x_t, \lambda_m^z) = N(y_t; \mu_m^y + \Sigma_m^{yx} {\Sigma_m^{xx}}^{-1}(x_t - \mu_m^x), \Sigma_m^{yy} - \Sigma_m^{yx} {\Sigma_m^{xx}}^{-1} \Sigma_m^{xy})$$

- Conversion rule:

$$\hat{y}_t = \sum_{m=1}^{M} P(m|x_t, \lambda_m^z)(\mu_m^y + \Sigma_m^{yx} {\Sigma_m^{xx}}^{-1}(x_t - \mu_m^x))$$

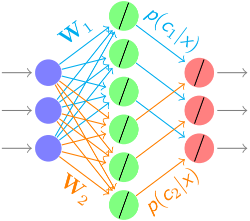
Artificial Neural Networks

- In this study, we use a two-layered ANN as mapping function
- Each layer of the network is of the form $\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$, where $\mathbf{x}$ and $\mathbf{h}$ are the input and output of that layer, respectively.
- Conversion rule:

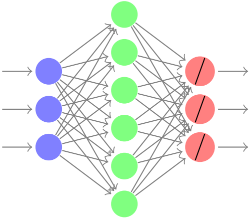
$$\hat{y}_t = W_2 f(W_1 \mathbf{x_t} + \mathbf{b}_1) + \mathbf{b}_2$$

where $f(.)$ is a nonlinear activation function (e.g. sigmoid).

ANN vs GMM



(a) GMM (Q=2)

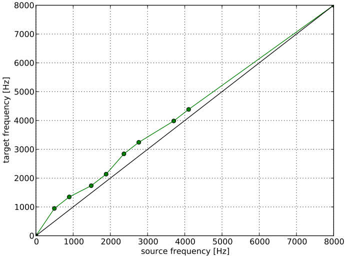


(b) ANN (6 hidden neurons)

Figure: GMM vs. ANN

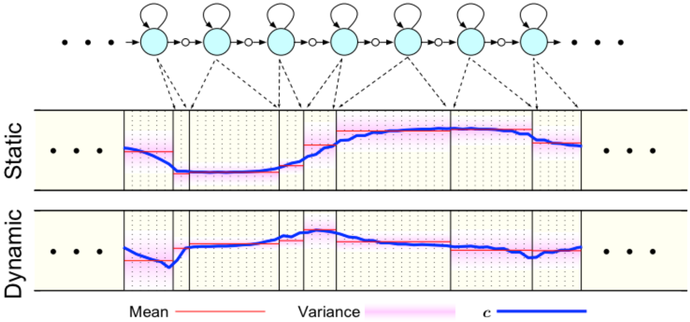
Transmutative approach

- The idea is to use analysis with high number of parameters
- Instead of using one of the previous mapping methods, use a constrained modification
- The modifications are performed by computing some "cues" from the source speech
- Usually, source-target formant frequencies are used as cues to warp the frequency spectrum
- This method is called Dynamic Frequency Warping



Trajectory Generation Algorithm

- The input to this algorithm are both static and dynamic parts of the feature sequence plus their covariance matrices.
- Its goal is to generate a smooth trajectory based on the above input [Tokuda 1995].



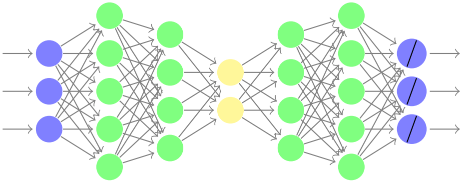
Tokuda 2009, http://www.sp.nitech.ac.jp/~tokuda/tokuda_interspeech09_tutorial.pdf

Experiment [1]: DNN and FS

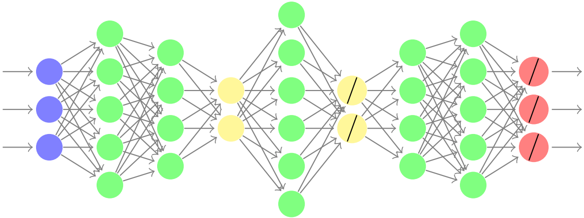
- DNN
  - pre-training multilayered ANN using a Deep Autoencoder (DAE)
  - The speaker independent DAE is trained on other speakers (not the source-target)
- FS
  - Frame Selection
  - Trajectory Generation is used after that to improve quality

[1] Seyed Hamidreza Mohammadi and Alexander Kain. Deep learning strategies for voice conversion. submitted to INTERSPEECH, 2014.

Experiment [1]: DNN and FS



(a) Deep Autoencoder



(b) Deep Neural Network

Figure: Deep Neural Networks

Experiment [1]: Features

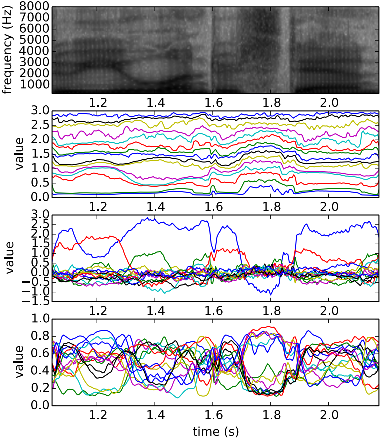


Figure: From top to bottom, spectrogram of the utterance "Do you understand?", and corresponding LSF features, MCEP features, and DMCEP features.

Data

- A corpus of $11=7+4$ speakers was used in this study.
- 7 speakers: approximately 1–2 hours of each speaker to train the speaker-independent DAE.
- 4 speaker: (two males: M1, M2, two females: F1, F2) were used for training and testing the voice conversion system.
- "large" training set: 70 sentences
- "small" training set: 2 sentences
- testing: 20 conversational sentences.
- four different conversions: two intra-gender (M1$\rightarrow$ M2, F2$\rightarrow$ F1) and two cross-gender (M2$\rightarrow$ F2, and F1$\rightarrow$ M1)

Training

- Features
  - MCEP (24^{th} order)
  - LSF (18^{th} order)
  - DMCEP (deep MCEP) (15^{th} order)
  - DLSF (deep LSF) (15^{th} order)
- Mapping
  - FS (k=10)
  - GMM
  - ANN
  - DNN

Perceptual Evaluation

- comparative mean opinion score (CMOS) test
- listeners hear two utterances A and B with the *same* content and the *same* speaker but in two *different* conditions,
- they are then asked to indicate wether they thought B was better or worse than A, using a five-point scale consisting of +2 (much better), +1 (somewhat better), 0 (same), -1 (somewhat worse), -2 (much worse).
- Amazon Mechanical Turk experiment: 40 listeners listen to 40 sentences

Speech Quality

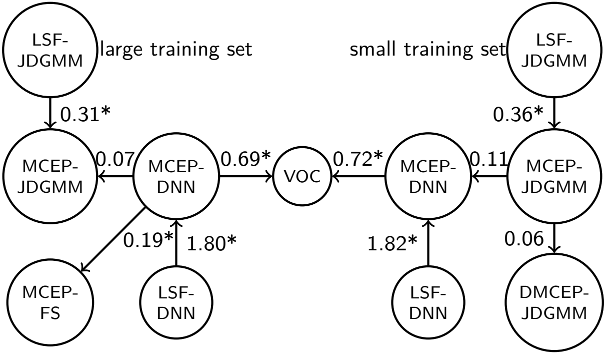


Figure: Speech quality (the direction of arrows show the better performer and asterisks show stat. significance)

Conversion Accuracy

- same-different speaker recognition test
- listeners hear two stimuli A and B with *different* content, and are then asked to indicate wether they thought that A and B were spoken by the *same* or by two *different* speakers, using a five-point scale consisting of +2 (definitely same), +1 (probably same), 0 (unsure), -1 (probably different), and -2 (definitely different).

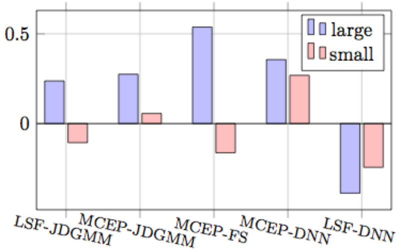


Figure 7: Conversion accuracy

Experiment [2]: frequency warping

- We used 70 Harvard sentences spoken by two male (M1, M2) and two female speakers (F1, F2)
- Two cross-gender (M1→F1, F2→M2) and two intra-gender (M2→M1, F1→F2) conversions
- train set: 46 of sentences
- development set: 4 sentences
- test set: 20 sentences
- GEN: the generative system, we trained a JDGMM to map 18^{th}-order LSF source parameters to same-order LSF target parameters, in an impulse/noise-excited LPC
- FOR: the transmutative system, we trained a JDGMM formant predictor from source cepstrum which predicts both source and target formant frequencies used as cues for frequency warping.

[2] Seyed Hamidreza Mohammadi and Alexander Kain. Transmutative voice conversion. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 6920–6924. IEEE, 2013.

Experiment [2]: Speech quality

- CMOS test on AMT
- A total of 35 listeners evaluated the test sentences.

	FOR-GEN	GEN-NAT	FOR-NAT
all	-0.43(1.4)	1.83(0.4)	1.57(1.1)

- The two-tail *t*-test shows a statistically significance difference between FOR-NAT and GEN-NAT ($t(188) = 2.24$, $p = 0.026$).

Experiment [2]: conversion accuracy

- same-difference speaker recognition test
- The experiment was administered to 44 listeners on AMT
- Each listener judged 40 sentence pairs, 10 trials for each of the three conversions.

NAT-	NAT	FOR	GEN	LSF
same	1.39	-0.38	0.12	1.04
diff	-1.32	-0.68	-0.22	-1.08
all	1.36	0.14	0.17	1.06

- A two-tail *t*-test show an statistically insignificant difference between the NAT-FOR and NAT-GEN ($p = 0.24$).

Experiment [3]: Conversational to Clear

- apply voice conversion approaches to the task of style conversion
- this experiment: Conversational -> Clear
- Clear (CLR): when talking to a listener who is impaired in their understanding either due to hearing loss, the presence of background noise, or both.
- Conversational (CNV): the speaking style which is intended for a normal hearing listener
- In this experiment: map short-term speech spectra from CNV speech -> CLR speech spectra,

[3] Seyed Hamidreza Mohammadi, Alexander Kain, and Jan PH van Santen. Making conversational vowels more clear. In INTERSPEECH, 2012.

Experiment [3]: Intelligibility test

- Corpus of 242 American English sentences, uttered by one male speaker.
- Each sentence was recorded in two styles, CNV and CLR.
  - CNV speech: the speaker was asked to speak as if talking with a friend at a natural pace.
  - CLR speech: the speaker was asked to "enunciate consonants more carefully and with greater effort than in CNV speech and avoid slurring words together" [Hefler 1998].
- This CVC word was located in a neutral carrier phrase immediately following the phoneme /d/ (e. g. "I know the meaning of the word moon").
- We recorded each sentence twice in both speaking styles, resulting in a total of 242 words $\times$ 2 styles $\times$ 2 renditions $=$ 968 tokens.

Experiment [3]: Training the system

- A total of 49 CVC words from the test set were used for a perceptual listening test.
- We created seven conditions for each word:
  - CNV,
  - CLR,
  - LSF-vocoded CNV (VCNV),
  - LSF-vocoded CLR (VCLR),
  - CNV with mapped spectrum (MAP-S),
  - CNV with CLR "oracle" duration (MAP-D) and
  - CNV with mapped spectrum and CLR duration (MAP-SD),
- a total of 49 words $\times$ 7 conditions $=$ 343 stimuli.
- All conditions of the same word were loudness-normalized.
- Babble noise with signal-to-noise ratios (SNR) of $+3$ dB and -2 dB were added to each stimulus

Experiment [3]: Intelligibility test

- 49 AMT listeners/test $\times$ 2 SNR levels $=$ 98 listeners,
    - all of whom had approval ratings of at least 90% and were located in the USA.
- Each listener was presented with 49 stimuli in a balanced test design.
- We asked the participant to listen to the word in noise and select one of the vowel classes based on what they heard.
- The intelligibility rate of each condition is shown in Table

Configuration	-2 dB SNR	$+3$ dB SNR
CLR	74.92%	80.46%
VCLR	71.42%	78.71%
MAP-SD	56.26%*	58.60%†
MAP-S	49.85%	59.76%†
MAP-D	48.10%	56.26%
VCNV	45.18%	52.47%
CNV	45.48%	56.55%

Conclusion

- VC methods:
  - Deep Neural Networks
  - Frame Selection + Trajectory Generation algorithm
  - Frequency warping using predicted formants
- VC Applications:
  - Transforming speaker identity [1,2]
  - Making Conversational Vowels more Clear [3]

Thank you!

Questions?