ELSEVIER

Speech Communication 28 (1999) 211–226

SPEECH COMMUNICATION

www.elsevier.nl/locate/specom

# Speaker Transformation Algorithm using Segmental Codebooks (STASC) [1]

Levent M. Arslan [2]

*Electrical and Electronics Department, Boğaziçi University, Bebek, 80815 Istanbul, Turkey*

Received 27 February 1998; received in revised form 17 December 1998; accepted 8 February 1999

**Abstract**

This paper presents a new voice conversion algorithm which modifies the utterance of a source speaker to sound-like speech from a target speaker. We refer to the method as Speaker Transformation Algorithm using Segmental Codebooks (STASC). A novel method is proposed which finds accurate alignments between source and target speaker utterances. Using the alignments, source speaker acoustic characteristics are mapped to target speaker acoustic characteristics. The acoustic parameters included in the mapping are vocal tract, excitation, intonation, energy, and duration characteristics. Informal listening tests suggest that convincing voice conversion is achieved while maintaining high speech quality. The performance of the proposed system is also evaluated on a simple Gaussian mixture model-based speaker identification system, and the results show that the transformed speech is assigned higher likelihood by the target speaker model when compared to the source speaker model. © 1999 Elsevier Science B.V. All rights reserved.

*Keywords:* Voice conversion; Speaker transformation; Codebook; Line spectral frequencies; Hidden Markov models; Time-varying filter; Overlap-add analysis

## 1. Introduction

There has been a considerable amount of research effort directed at the problem of voice transformation recently (Abe et al., 1988; Baudoin and Stylianou, 1996; Childers, 1995; Iwahashi and Sagisaka, 1995; Kuwabara and Sagisaka, 1995; Narendranath et al., 1995; Stylianou et al., 1995). This topic has numerous applications which include personification of text-to-speech systems, multimedia entertainment, and as a preprocessing step to speech recognition to reduce speaker variability. In general, the approach to the problem consists of a training phase where input speech training data from source and target speakers are used to formulate a spectral transformation that would map the acoustic space of the source speaker to that of the target speaker. The acoustic space can be characterized by a number of possible acoustic features which have been studied extensively in the literature. The most popular features used for voice transformation include formant frequencies (Abe et al., 1988; Narendranath et al., 1995), and LPC cepstrum coefficients (Lee et al., 1996). The transformation is in general based on codebook mapping (Abe et al., 1988; Acero, 1993; Baudoin and Stylianou, 1996; Lee et al., 1996).

[1] Speech files available. See www.elsevier.nl/locate/specom.

[2] Tel.: +90 212 2631540/1421; fax: +90 212 2872465; e-mail: arslanle@boun.edu.tr. The author was with Entropic Research Laboratory, Washington, DC.


PII: S0167-6393(99)00015-1

That is, a one-to-one correspondence between the spectral codebook entries of the source speaker and the target speaker is developed by some form of supervised vector quantization method. In general, these methods face several problems such as artifacts introduced at the boundaries between successive speech frames, limitation on robust estimation of parameters (e.g., formant frequency estimation), or distortion introduced during synthesis of target speech. Another issue which has not been explored in detail is the transformation of the excitation characteristics in addition to the vocal tract characteristics. Several studies proposed solutions to address this issue recently (Childers, 1995; Lee et al., 1996). In this study, we propose new and effective solutions to both problems with the goal of maintaining high speech quality.

## 2. Algorithm description

This section provides a general description of the Speaker Transformation Algorithm using Segmental Codebooks (STASC) algorithm. We will describe the algorithm under two main sections: (i) transformation of spectral characteristics, (ii) transformation of prosodic characteristics.

### 2.1. Spectral transformation

For the representation of the vocal tract characteristics of source and target speakers line spectral frequencies (LSF) are selected. The reason for selecting LSFs is that these parameters relate closely to formant frequencies (Crosmer, 1985), but in contrast to formant frequencies they can be estimated quite reliably. They have been used for a number of applications successfully in the literature (Hansen and Clements, 1991; Arslan et al., 1995; Arslan and Talkin, 1997; Crosmer, 1985; Laroia et al., 1991; Itakura, 1975; Pellom and Hansen, 1997). They have good interpolation properties and they are stable (Paliwal, 1995). In addition, they have a fixed dynamic range which makes them attractive for real-time DSP implementation. LSFs can be estimated by modifying the LPC polynomial, $A(z)$, in two ways: $P(z)$ and $Q(z)$ are obtained by augmenting $A(z)$'s PARCOR sequence with $a+1$ and $-1$, respectively. This results in the following two polynomials which have all their roots on the unit circle:

$$\begin{aligned} P(z) &= (1-z^{-1}) \\ &\quad \times \prod_{k=1,3,5,\ldots}^{P-1} (1 - 2\cos w_k (z^{-1} + z^{-2})), \\ Q(z) &= (1+z^{-1}) \\ &\quad \times \prod_{k=2,4,6,\ldots}^{P-1} (1 - 2\cos w_k (z^{-1} + z^{-2})), \end{aligned} \tag{1}$$

where $P$ is the LPC analysis order, and the angles of the roots, $w_k$, are LSFs. In STASC algorithm, codebooks of LSFs are used to represent the vocal tract characteristics of individual speakers. The codebooks can be generated in two ways.

The first method assumes that the orthographic transcription is available along with the training data. The training speech (sampled at 16 kHz) from source and target speakers are first segmented automatically using forced alignment to a phonetic translation of the orthographic transcription. The segmentation algorithm uses Mel-cepstrum coefficients and delta coefficients within an HMM framework and is described in detail in (Wightman and Talkin, 1994). The LSFs for source and target speaker utterances are calculated on a frame-by-frame basis and each LSFs vector is labeled using the phonetic segmenter. Next, a centroid LSFs vector for each phoneme is estimated for both source and target speaker codebooks by averaging across all the corresponding speech frames. The estimated codebook spectra for an example male source speaker and female target speaker combination from the database is shown in Fig. 1 when monophones are selected as speech units. A one-to-one mapping is established between the source and target codebooks to accomplish the voice transformation.

The second method does not require the phonetic translation of the orthographic transcription for the training utterances, however it assumes that both source and target speakers are speaking the same sentences during the training session. This method is a new method and it is referred to as "Sentence HMM" method. The method is as fol-

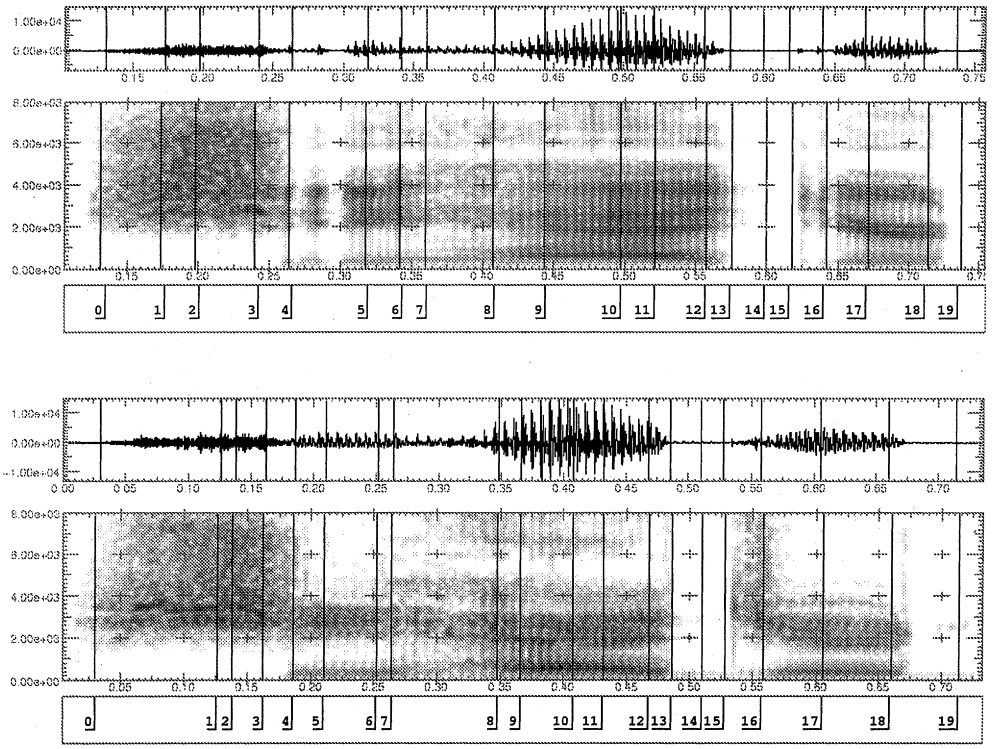

Fig. 1. The state alignments for source and target speaker utterances "She had your".

lows. First, template sentences are selected which are phonetically balanced to be uttered by the source and target speakers. After the training data are collected, silence regions at the beginning and end of each utterance are removed. Each utterance is normalized in terms of its RMS energy to account for differences in the recording gain level. Next, cepstrum coefficients are extracted along with log-energy and zero-crossing for each analysis frame in each utterance. Zero-mean normalization is applied to the parameter vector to obtain a more robust spectral estimate. Based on the parameter vector sequences sentence HMMs are trained for each template sentence using data from the source speaker. The number of states for each sentence HMM is set proportional to the duration of the utterance. The training is done using a segmental $k$-means algorithm followed by the Baum–Welch algorithm. The initial covariance matrix is estimated over the complete training dataset, and is not updated during the training since the amount of data corresponding to each state is not sufficient to make a reliable estimate of the variance. Next, the best state sequence for each utterance is estimated using the Viterbi algorithm. The average LSFs vector for each state is calculated for both source and target speakers using frame vectors corresponding to that state index. Finally, these average LSFs vectors for each sentence are collected to build the source and target speaker codebooks. In Fig. 2, the alignments to the state indices are shown for the utterance "She had your" both for source and target speaker utterances. From the figure, it can be observed that detailed acoustic alignment is achieved quite accurately using sentence HMMs. The transformation process will be explained in detail later in this section.

Another factor that influences speaker individuality is excitation characteristic. The LPC residual can be a reasonable approximation to the excitation signal. It is well known that the residual can be very different for different phonemes (e.g., periodic

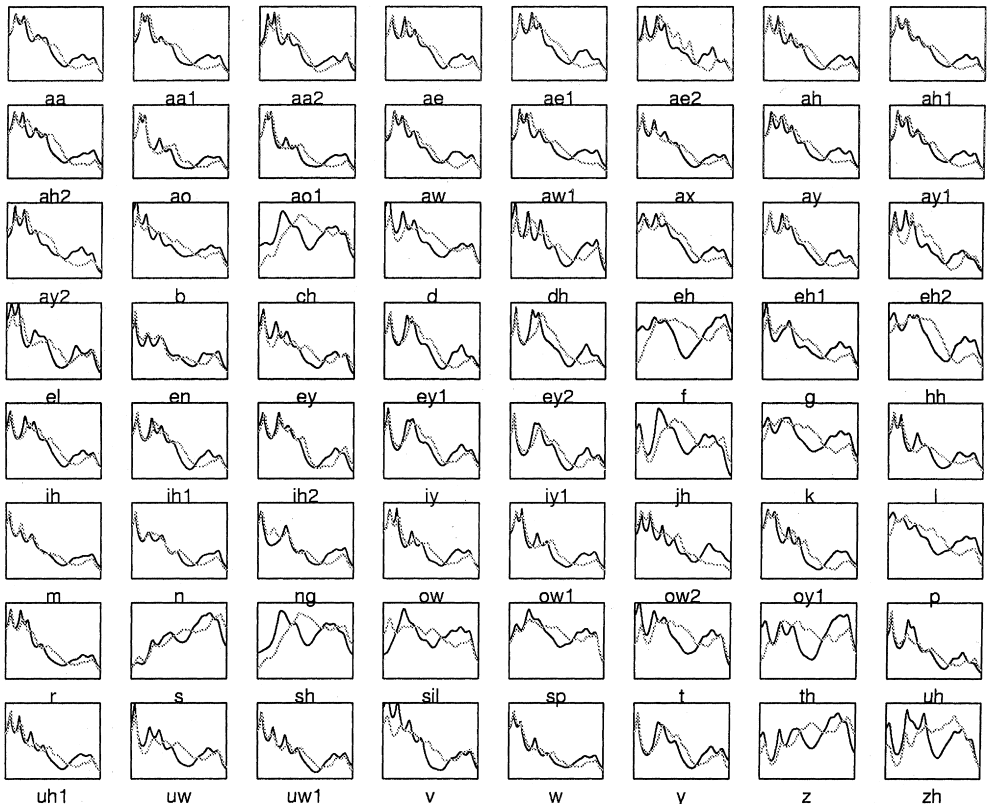


Fig. 2. Comparison of monophone codebooks derived from the source speaker (solid line) and the target speaker (dotted line).

pulse train for voiced sounds versus white noise for unvoiced sounds). Therefore, we formulated a "codebook based" transformation of the excitation spectrum similar to the one discussed above for the vocal tract spectrum transformation. Excitation spectrum codebooks are obtained as follows: using the segmentation information, the LPC residual signals for each speech unit (i.e., phoneme or state) in the codebook are collected from the training data. Next, a short-time average magnitude spectrum of the excitation signal is estimated for each speech unit both for the source speaker and the target speaker pitch synchronously. An excitation transformation filter can be formulated for each codeword entry using the excitation spectra of the source speaker and the target speaker. This method not only transforms the general excitation characteristics, but it estimates a reasonable transformation for the "zeros" in the spectrum as well, which are not represented accurately by the all-pole modeling. Therefore, this method resulted in improved voice conversion performance especially for nasalized sounds. It should be noted here that an attempt is being made here to only approximate the spectral characteristics of the LPC residual. Other salient properties of the glottal excitation and voice quality such as pulse jitter, aspiration and noise bursts have not been transformed.

The flow diagram of the STASC voice transformation algorithm is shown in Fig. 3. The incoming speech is first sampled at 16 kHz. Next, 20th order LPC analysis is performed to estimate the prediction coefficients vector $\boldsymbol{a}$.

Based on the source-filter theory, the incoming speech spectrum $X(\omega)$ can be represented as

$$X(\omega) = G_s(\omega)V_s(\omega), \tag{2}$$

where $G_s(\omega)$ and $V_s(\omega)$ represent source speaker excitation and vocal tract spectra, respectively for the incoming speech frame $x(n)$.

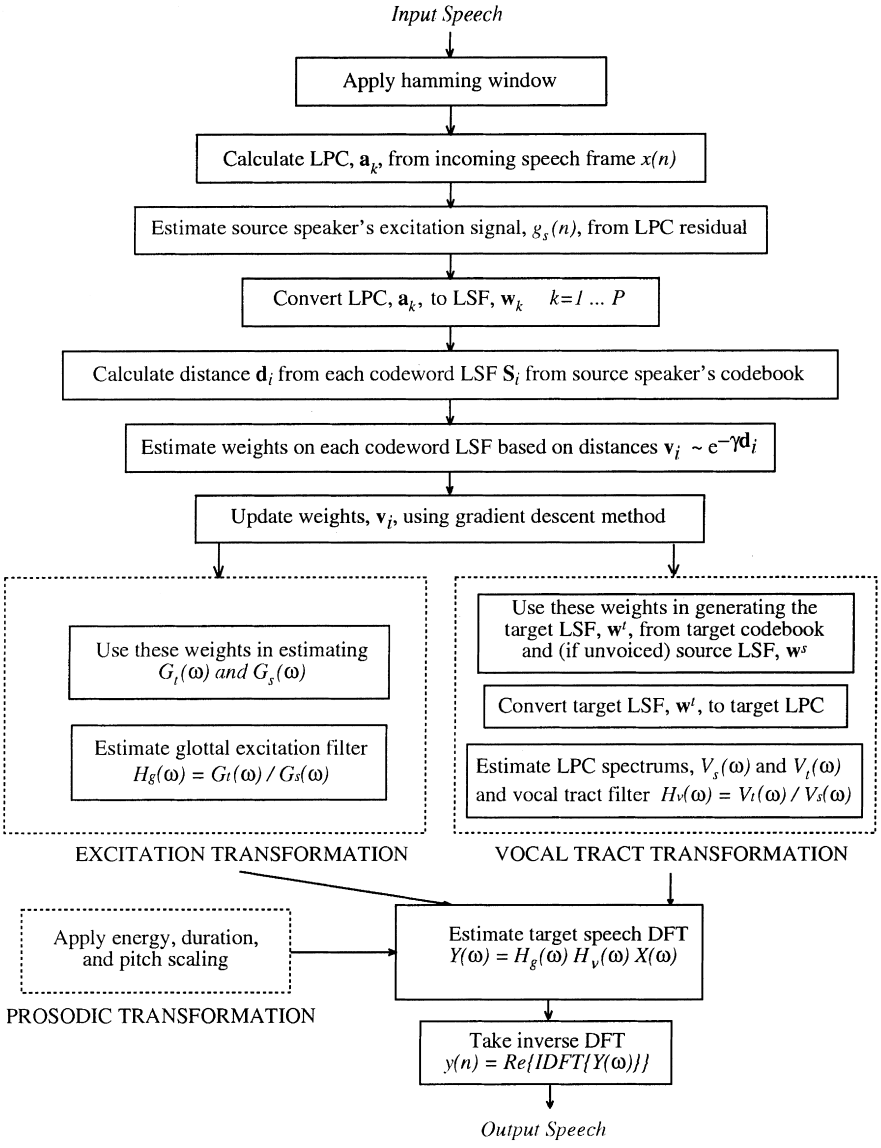


Fig. 3. Flow-diagram of STASC voice conversion algorithm.

The target speech spectrum $Y(\omega)$ can be formulated as

$$Y(\omega) = \left[\frac{G_t(\omega)}{G_s(\omega)}\right]\left[\frac{V_t(\omega)}{V_s(\omega)}\right]X(\omega), \quad (3)$$

where $V_t(\omega)$ and $G_t(\omega)$ represent codebook estimated target vocal tract and excitation spectra, respectively. This representation of the target spectrum can be thought of as an excitation filter, $H_g(\omega)$, followed by a vocal tract filter, $H_v(\omega)$:

$$Y(\omega) = H_g(\omega)H_v(\omega). \quad (4)$$

In the proposed algorithm, the source speaker vocal tract spectrum $V_s(\omega)$ is estimated from the LPC spectrum of the incoming speech frame:

$$V_s(\omega) = \left| \frac{1}{1 - \sum_{k=1}^{P} \boldsymbol{a}_k e^{-jkw}} \right|. \quad (5)$$

The LPC vector $\boldsymbol{a}$ can be derived in two ways: (i) directly from the incoming speech frame; or (ii) as an approximation from the source speaker codebook. For (ii), first the incoming LSFs vector can be approximated as

$$\tilde{\boldsymbol{w}}_k = \sum_{i=1}^{L} \boldsymbol{v}_i \boldsymbol{S}_{ik}, \quad k = 1, \ldots, P, \quad (6)$$

where $L$ is the codebook size, $\boldsymbol{S}_i$ is the $i$th codeword LSFs vector and $\boldsymbol{v}_i$ represents its weight. Next, $\tilde{\boldsymbol{w}}$ can be converted to the LPC vector $\boldsymbol{a}$ to be used in Eq. (5). In practice, we found that using method (i) for voiced segments and method (ii) for unvoiced segments produces best results in terms of output speech quality. For both methods (i) and (ii) the codebook weights need to be calculated for the target spectrum estimate $V_t(\omega)$. The codebook size $L$ is the total number of states for the sentence HMM-based STASC, and is the total number of available common phonetic units between source and target speaker utterances for the phonetic STASC algorithm. Average codebook size in our evaluations ranged from 50 to 2000 depending on the size of the training data. The typical segment size was 50 ms. The codebook weight estimation procedure is as follows.

### 2.1.1. Codebook weight estimation method

First, using Eq. (1), LSFs, $\boldsymbol{w}$, are derived from prediction coefficients. The line spectral frequency vector $\boldsymbol{w}$ is compared with each LSFs centroid, $\boldsymbol{S}_i$, in the source codebook and the distance, $\boldsymbol{d}_i$, corresponding to each codeword is calculated. The distance calculation is based on a perceptual criterion where closely spaced LSFs which are likely to correspond to formant locations are assigned higher weights (Laroia et al., 1991),

$$\boldsymbol{h}_k = \frac{1}{\operatorname{argmin}(|\boldsymbol{w}_k - \boldsymbol{w}_{k-1}|, |\boldsymbol{w}_k - \boldsymbol{w}_{k+1}|)}, \quad k = 1, \ldots, P,$$

$$\boldsymbol{d}_i = \sum_{k=1}^{P} \boldsymbol{h}_k |\boldsymbol{w}_k - \boldsymbol{S}_{ik}|, \quad i = 1, \ldots, L. \quad (7)$$

Based on the distances from each codebook entry, an expression for the normalized codebook weights can be obtained as (Arslan et al., 1995)

$$\boldsymbol{v}_i = \frac{e^{-\gamma \boldsymbol{d}_i}}{\sum_{l=1}^{L} e^{-\gamma \boldsymbol{d}_l}}, \quad i = 1, \ldots, L, \quad (8)$$

where the value of $\gamma$ for each frame is found by an incremental search in the range 0.2–2 with the criterion of minimizing the perceptual weighted distance between the approximated LSFs vector $\tilde{\boldsymbol{w}}$ and original LSFs vector $\boldsymbol{w}$. However, this set of weights may still not be the optimal set of weights that would represent the original speech spectrum. In order to improve the estimate of weights a gradient descent algorithm is employed. The previously estimated set of weights $\boldsymbol{v}_i$ are used as the initial seed to the gradient descent algorithm. The weights are constrained to have positive values after each iteration of the gradient descent algorithm to prevent unrealistic estimates. The codebook weight update algorithm can be summarized as follows.

### 2.1.2. Codebook weight update by gradient descent method

*Initialize*:
$E^0 = \infty$
$\eta^0 = 0.1$
$n = 1;$

*Loop*
$\boldsymbol{e} = \boldsymbol{h} \cdot (\boldsymbol{w} - \boldsymbol{S}\boldsymbol{v}^{n-1})$
$E^n = \sum_{k=1}^{P} |\boldsymbol{e}_k|$
$\boldsymbol{v}_i^n = \boldsymbol{v}_i^{n-1} + \eta^{n-1} \boldsymbol{e}^{\mathrm{T}} \boldsymbol{S}_i, \ i = 1, \ldots, L$
$\boldsymbol{v}_i^n = \max(\boldsymbol{v}_i^n, 0), \ i = 1, \ldots, L$

*if* $E^n < E^{n-1}$
$\eta^n = 2\eta^{n-1}$

*else*
$\eta^n = 0.1\eta^{n-1}$
$n = n + 1$

*until* $\|v^n - v^{n-1}\| < 1.0e^{-4} \|vn - 1\|$

where $\boldsymbol{S}$ is a $P \times L$ size matrix whose columns represent a codeword LSFs vector, and $\eta^n$ is a constant which controls the rate of convergence at iteration $n$. In our implementation, in order to reduce the amount of computation, $\eta$ is adjusted after each iteration based on the reduction in error $E^n$ with respect to $E^{n-1}$. If there is significant amount of reduction in error then $\eta$ is increased,

otherwise it is decreased. Typical ending values for $\eta$ are in the range 1.0e−3 and 1.0e−5.

It was also observed that only a few codebook entries were assigned significantly large weight values in initial weight vector estimate $\boldsymbol{v}^0$. Therefore, in order to save computational resources the gradient descent algorithm was performed on only 5 most likely codeword weights. Using the gradient descent method, a 15% additional reduction in average Itakura–Saito distance between the original and approximated spectra was achieved for the training data used in our experiments. The average spectral distortion (SD), which is a commonly used measure for spectral quantizer performance evaluation, was also reduced from 1.8 to 1.4 dB.

*2.1.3. Excitation spectrum mapping*

The estimated set of codebook weights can be utilized in three separate domains: (i) transformation of the excitation spectral characteristics, (ii) transformation of the vocal tract characteristics, (iii) transformation of prosodic characteristics. Although one may benefit from estimating a different set of codebook weights for each of the three domains mentioned above, in this study we chose to apply the same set of weights $\boldsymbol{v}$ mainly for computational reasons. For the transformation of the excitation spectrum, these weights are used to construct an overall filter which is a weighted combination of codeword excitation filters:

$$H_{\mathrm{g}}(\omega) = \sum_{i=1}^{L} \boldsymbol{v}_i \frac{\boldsymbol{U}_i^{\mathrm{t}}(\omega)}{\boldsymbol{U}_i^{\mathrm{s}}(\omega)}, \tag{9}$$

where $\boldsymbol{U}_i^{\mathrm{t}}(\omega)$ and $\boldsymbol{U}_i^{\mathrm{s}}(\omega)$ denote average target and source excitation magnitude spectra for the $i$th codeword, respectively.

*2.1.4. Vocal tract spectrum mapping*

The same set of codebook weights $(\boldsymbol{v}^i, i = 1, \ldots, L)$ are applied to target LSFs vectors $(\boldsymbol{T}_i, i = 1, \ldots, L)$ to construct the target line spectral frequency vector $\tilde{\boldsymbol{w}}^{\mathrm{t}}$:

$$\tilde{\boldsymbol{w}}_k^{\mathrm{t}} = \sum_{i=1}^{L} \boldsymbol{v}_i \boldsymbol{T}_{ik}, \quad k = 1, \ldots, P. \tag{10}$$

Next, target LSFs are converted to prediction coefficients, $\boldsymbol{a}^{\mathrm{t}}$, which in turn are used to estimate the target LPC vocal tract filter:

$$V_{\mathrm{t}}(\omega) = \left| \frac{1}{1 - \sum_{k=1}^{P} \boldsymbol{a}_k^{\mathrm{t}} \mathrm{e}^{-\mathrm{j}k\omega}} \right|. \tag{11}$$

In general, the weighted codebook representation of the target spectrum results in expansion of formant bandwidths due to the interpolation of LSFs. In order to cope with this problem a new bandwidth modification algorithm is proposed.

*2.1.5. Bandwidth modification method*

The bandwidth modification algorithm makes use of the knowledge that average formant bandwidth values of the target speech should be similar to the most likely target codeword. Once an estimate of bandwidths for the most likely target codeword is obtained, bandwidths of the target speech can be forced to be similar to this estimate by modifying the distance between line spectrum pairs representing each formant. The algorithm can be formulated as follows. First find the line spectral pair $\boldsymbol{w}_{j(i)}^{\mathrm{t}}$ and $\boldsymbol{w}_{j(i)+1}^{\mathrm{t}}$ in the estimated target LSFs vector $\boldsymbol{w}^{\mathrm{t}}$ that corresponds to each formant frequency location $\boldsymbol{f}_i^{\mathrm{t}}$, $i = 1, \ldots, 4$ (i.e., the smallest LSFs $> \boldsymbol{f}_i^{\mathrm{t}}$ and the largest LSF $< \boldsymbol{f}_i^{\mathrm{t}}$). Likewise, find the line spectral pair $\boldsymbol{w}_{k(i)}^{\mathrm{l}}$ and $\boldsymbol{w}_{k(i)+1}^{\mathrm{l}}$ in the most likely target codeword $\boldsymbol{w}^{\mathrm{l}}$. Next, for each formant frequency, $\boldsymbol{f}_i^{\mathrm{t}}$, in the estimated target spectrum estimate an approximation to the bandwidth, $\boldsymbol{b}_i^{\mathrm{t}}$, based on the corresponding LSFs distances. Estimate the most likely codeword bandwidths, $\boldsymbol{b}_i^{\mathrm{l}}$, in a similar fashion.

$$\begin{aligned} \boldsymbol{b}_i^{\mathrm{l}} &= \boldsymbol{w}_{k(i)+1}^{\mathrm{l}} - \boldsymbol{w}_{k(i)}^{\mathrm{l}}, \quad i = 1, \ldots, 4, \\ \boldsymbol{b}_i^{\mathrm{t}} &= \boldsymbol{w}_{j(i)+1}^{\mathrm{t}} - \boldsymbol{w}_{j(i)}^{\mathrm{t}}, \quad i = 1, \ldots, 4. \end{aligned} \tag{12}$$

Calculate average formant bandwidths, and find the bandwidth ratio $r$:

$$\bar{\boldsymbol{b}}^{\mathrm{l}} = \frac{1}{4} \sum_{i=1}^{4} \boldsymbol{b}_i^{\mathrm{l}}, \quad \bar{\boldsymbol{b}}^{\mathrm{t}} = \frac{1}{4} \sum_{i=1}^{4} \boldsymbol{b}_i^{\mathrm{t}}, \quad r = \frac{\bar{\boldsymbol{b}}^{\mathrm{l}}}{\bar{\boldsymbol{b}}^{\mathrm{t}}}. \tag{13}$$

Finally, apply the estimated bandwidth ratio to adjust the line spectral pairs around each formant:

$$\begin{aligned} \hat{\boldsymbol{w}}_{j(i)}^{\mathrm{t}} &= \boldsymbol{w}_{j(i)}^{\mathrm{t}} + (1 - r) \cdot (\boldsymbol{f}_i^{\mathrm{t}} - \boldsymbol{w}_{j(i)}^{\mathrm{t}}), \\ \hat{\boldsymbol{w}}_{j(i)+1}^{\mathrm{t}} &= \boldsymbol{w}_{j(i)+1}^{\mathrm{t}} + (1 - r) \cdot (\boldsymbol{f}_i^{\mathrm{t}} - \boldsymbol{w}_{j(i)+1}^{\mathrm{t}}), \\ & i = 1, \ldots, 4, \end{aligned} \tag{14}$$

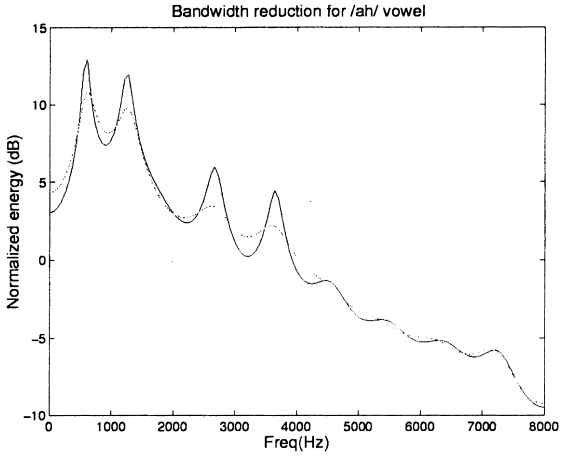


Fig. 4. Illustration of reduction of bandwidths using the proposed method. The light curve represents the normalized power spectrum for the /ah/ vowel before bandwidth modification, and the dark curve represents the power spectrum after bandwidth modification.

where $\hat{\boldsymbol{w}}^{\mathrm{t}}_{j(i)}$ and $\hat{\boldsymbol{w}}^{\mathrm{t}}_{j(i)+1}$ represent the line spectral frequency pair around $\boldsymbol{f}^{\mathrm{t}}_i$ after bandwidth modification. In order to prevent the estimation of unreasonable bandwidths the minimum bandwidth value is set to be $\frac{f^{\mathrm{t}}_i}{20}$ Hz.

In Fig. 4, a comparison of the target speech spectrum for the /ah/ vowel before and after the application of the proposed bandwidth reduction technique is shown.

#### 2.1.6. Combined output

The vocal tract filter and excitation filters are next applied to the magnitude spectrum of the original signal to get an estimate of the DFT corresponding to the target speech signal:

$$Y(\omega) = H_{\mathrm{g}}(\omega)H_{\mathrm{v}}(\omega)X(\omega). \tag{15}$$

Finally, inverse DFT is applied to produce the synthetic target voice,

$$y(n) = \mathrm{Real}\{\mathrm{IDFT}\{Y(\omega)\}\}. \tag{16}$$

### 2.2. Prosodic transformation

In the STASC algorithm a frequency domain pitch synchronous analysis synthesis framework is adopted in order to be able to realize both spectral and prosodic transformations simultaneously. In addition to the spectral transformation discussed in the previous section, pitch, duration and energy are modified to mimic target speaker prosodic characteristics. Analysis frame length is set to be constant for unvoiced regions. For voiced regions the frame length is set to two or three pitch periods depending on the pitch modification factor. It is observed that when the pitch modification factor is less than one, using a shorter frame length (i.e., 2 pitch periods) reduces artifacts introduced by the modification. [3]

#### 2.2.1. Pitch-scale modification

The pitch modification involves matching both the average pitch value and range for the target speaker. This is accomplished by modifying the instantaneous source speaker fundamental frequency [4] $f^{\mathrm{s}}_0(t)$ by a multiplicative constant $a$ and an additive constant $b$ at each time frame $t$:

$$f^{\mathrm{t}}_0(t) = af^{\mathrm{s}}_0(t) + b. \tag{17}$$

The value for $a$ is set so that the source speaker pitch variance, $\sigma^2_{\mathrm{s}}$, and target speaker pitch variance, $\sigma^2_{\mathrm{t}}$, match, i.e.,

$$a = \sqrt{\frac{\sigma^2_{\mathrm{t}}}{\sigma^2_{\mathrm{s}}}}. \tag{18}$$

Once the value for $a$ is set, the value for the additive constant $b$ can be found by matching the average $f_0$ values.

$$b = \mu_{\mathrm{t}} - a\mu_{\mathrm{s}}, \tag{19}$$

where $\mu_{\mathrm{s}}$ and $\mu_{\mathrm{t}}$ represent the source and target mean pitch values. Therefore, the instantaneous pitch-scale modification factor $\beta(t)$ can be set as

$$\beta(t) = \frac{af^{\mathrm{s}}_0(t) + b}{f^{\mathrm{s}}_0(t)} \tag{20}$$

in order to achieve the desired target speaker pitch value and range.

[3] The value of $\gamma$ basically controls the dynamic range of codebook weights. By setting this value to infinity one can force the selection of a single codebook entry from the codebook.

[4] Pitch and fundamental frequency are used interchangeably in this context though they differ in their definitions. Pitch is in fact the perceived fundamental frequency.

*2.2.2. Duration-scale modification*

The duration characteristics can vary across different speakers significantly due to a number of factors including accent or dialect. Although modifying the speaking rate uniformly to match the target speaker duration characteristics reduces timing differences between speakers to some extent, it is observed that this is not sufficient in general. In Fig. 5, a comparison of the duration statistics of monophones for the two speakers in our database are given. The speakers uttered the same sentences, therefore the effect of duration variation based on context was normalized in the comparison. It can be seen from the table that the proportion of average durations are quite different among different phonemes. For example, the average duration of /ah/ vowel is 100 ms for source speaker, and 67 ms for target speaker. On the other hand, for the /uh/ vowel the target speaker has a longer average duration (64 versus 37 ms). Although on the average the target speaker has 1.2 times longer average duration than the source speaker, there exists a significant number of phonemes that the target speaker uses shorter duration for.

Based on the previous set of results it can be concluded that the variation in duration characteristics between two speakers is heavily dependent upon context. Therefore, it is highly desirable to develop a method for automatically estimating the appropriate time-scale modification factor in a certain context. In the STASC algorithm, a codebook-based approach to duration modification is implemented. The phonetic codebooks used for the spectral mapping can also be used to generate the appropriate duration modification factor for a given speech frame. In order to accomplish this, first duration statistics are estimated for both the source speaker and the target speaker for all the speech units in the codebook. Then the same codebook weights developed for the spectral mapping can be used to estimate the appropriate time-scale modification factor $\gamma$:

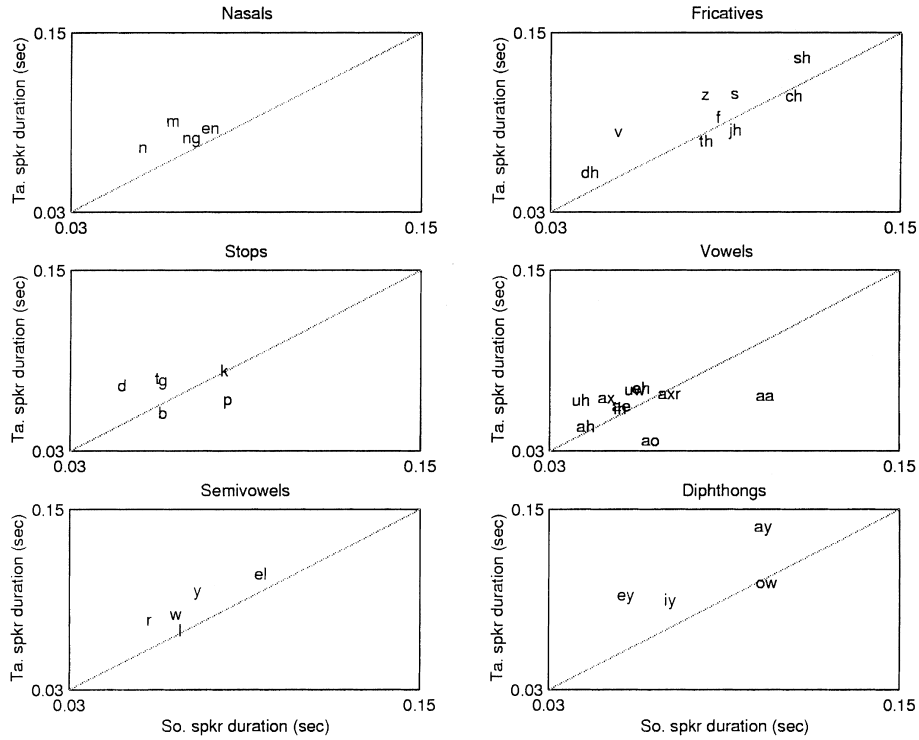


Fig. 5. Comparison of duration statistics between a source speaker and a target speaker.

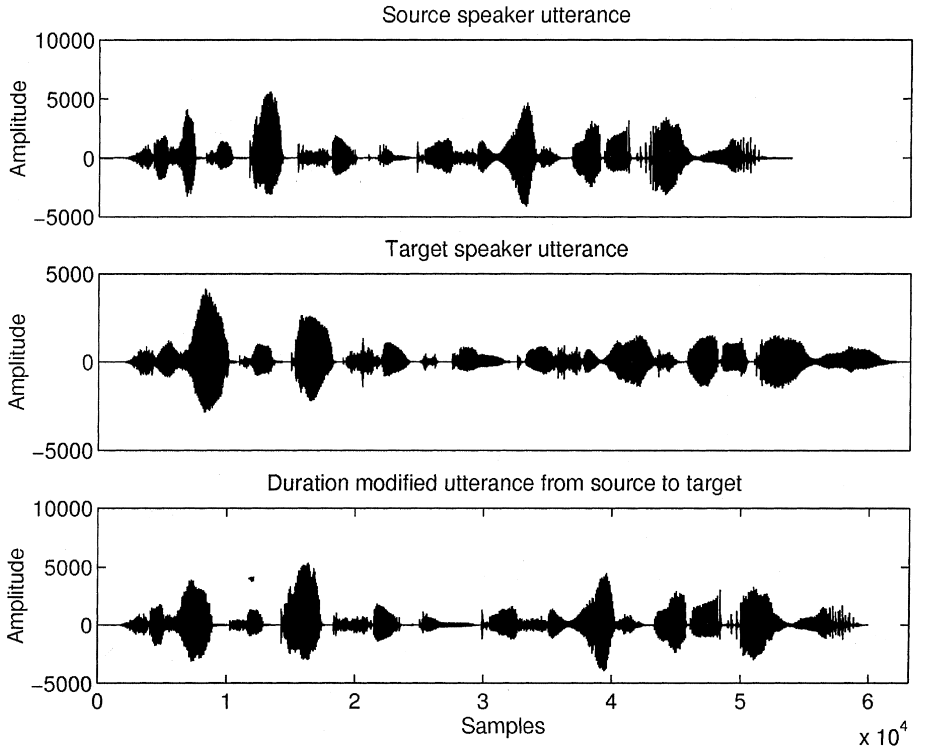


Fig. 6. Illustration of the duration modification algorithm on a TIMIT utterance. Top: source speaker; middle: target speaker; bottom: modified from source to match target duration characteristics.

$$\gamma = \sum_{i=1}^{L} \nu_i \frac{d_i^{\mathrm{t}}}{d_i^{\mathrm{s}}}, \tag{21}$$

where $d_i^{\mathrm{s}}$ and $d_i^{\mathrm{t}}$ represent average source and target speaker durations for the $i$th codeword entry.

In Fig. 6, the result of the duration modification is shown on a TIMIT sentence. In this case the duration statistics are generated from the label files corresponding to 10 sentences uttered by two speakers. [5] The top and middle plots in the figure correspond to the source and target speaker utterance "She had your dark suit in greasy wash water all year", respectively. The bottom plot in the figure shows the duration modified signal obtained from the source speaker utterance to match the duration statistics of the target speaker. Although this was a closed set example (i.e., the test utterance was one of the training examples) it serves well as an example to illustrate the motivation behind the proposed method. In order to take advantage of the context information triphones should be used as speech units. It should also be noted that the performance of the duration modification algorithm proposed here is dependent on whether the speakers were speaking the same utterances or not. If the utterances are the same, then the place of each phone in prosodic phrase will be the same. Therefore, a more accurate mapping can be achieved.

A major application for current time-scale modification algorithms is to slow down the speech for accurate transcription by humans. The problem with most of those systems is that they use a constant time-scale modification factor when changing the speaking rate. However, not all the phonemes are scaled to the same extent when a speaker modifies his/her speaking rate. Therefore,

[5] Only 2 sentences were common to both speakers, the remaining 8 sentences per speaker were unique.

the same approach proposed here for transforming duration characteristics across speakers can be applied to speaking rate modification algorithms if the statistics for slow, normal and fast speaking styles are generated prior to the application.

*2.2.3. Energy-scale modification*

In addition to pitch and duration, energy is another important component which characterizes the prosody of a speaker. In order to match target speaker's energy characteristics we applied a codebook-based energy mapping as well. The RMS energy is scaled with a variable $\eta$ at each time frame. The scaling factor can be expressed as follows:

$$\eta = \sum_{i=1}^{L} \boldsymbol{v}_i \frac{e_i^{\mathrm{t}}}{e_i^{\mathrm{s}}}, \tag{22}$$

where $e_i^{\mathrm{s}}$ and $e_i^{\mathrm{t}}$ denote average source and target speaker energies for the $i$th codeword.

Finally, the pitch-scale modification factor $\beta$, the time-scale modification factor $\gamma$, and the energy scaling factor $\eta$ are applied within a pitch–synchronous overlap–add synthesis framework to perform prosodic modification. [6]

The next section discusses the evaluations conducted to test the performance of the STASC algorithm.

## 3. Evaluations

In order to evaluate the performance of the STASC algorithm we performed both objective tests and subjective listening tests.

### 3.1. Objective tests

We propose two different methods to evaluate voice conversion performance. The first method uses a simple speaker identification (ID) system to estimate likelihoods for the source and target speakers. The second method employs sentence HMM-based alignments between the mimic and target utterances, and compares corresponding speech units in terms of their spectral and prosodic parameters. Using the second method, the performance of the phonetic and sentence HMM methods are also compared and the influence of the amount of training data on the STASC algorithm performance is also investigated.

*3.1.1. Speaker ID evaluation*

For the first objective test, we used a simple speaker identification system. The idea is that if we can make the speaker ID system select the target speaker after processing the source speaker utterance, it means that the voice conversion algorithm is performing well. Of course besides checking for the binary decision between the two speakers, one would like to have a confidence measure on the decision as well. For this reason, the log-likelihood ratio of the target speaker to that of the source speaker is adopted as an objective measure in our evaluations. The performance measure $\theta_{\mathrm{st}}$ can be formulated as

$$\begin{aligned}\theta_{\mathrm{st}} &= \log \frac{P(\boldsymbol{X}|\lambda_{\mathrm{t}})}{P(\boldsymbol{X}|\lambda_{\mathrm{s}})} \\ &= \log P(\boldsymbol{X}|\lambda_{\mathrm{t}}) - \log P(\boldsymbol{X}|\lambda_{\mathrm{s}}),\end{aligned} \tag{23}$$

where $\boldsymbol{X}$ is the observation vector sequence, $\lambda_{\mathrm{t}}$ is the target speaker model, and $\lambda_{\mathrm{s}}$ is the source speaker model. The speaker ID system employs 256 mixture Gaussian mixture models (GMM). The 24 dimensional feature vector used in the GMM system consists of 12 Mel-cepstrum coefficients and their delta coefficients. Initial vector quantization was done using binary split vector quantization method. This was followed by 2 iterations of forward–backward training. During data collection sessions 3 speakers were asked to read a different story to the tape recorder. The recorded speech was approximately one hour long for each speaker. Forty-five minutes of the recording was used as training data (both for speaker ID models and voice transformation codebooks), and fifteen minutes of speech was set aside for testing. Since each speaker read a different story, sentence HMMs were not used for this experiment. Triphone codebooks were generated

[6] How $\beta$ and $\gamma$ are used within a pitch–synchronous analysis–synthesis framework is described in detail in (Moulines and Charpentier, 1990).

Table 1
The speaker ID evaluation for voice conversion. Sp1: first speaker; Sp2: second speaker, Sp3: third speaker

| Test case | $\theta_{st}$ before conversion | $\theta_{st}$ 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 |

based on phonetic alignments. We can hear [7] an example transformation from the first speaker (Sp1) to the third speaker (Sp3) for this experimental set-up on Signal 1c. Signal 1a and Signal 1b correspond to (Sp1) and (Sp3) utterances, respectively.

The average likelihood of the Sp1 speech data for the first speaker model, $\log P(\boldsymbol{X}|\lambda_1)$, was −70.53. After using STASC for transformation to Sp2, Sp1 model likelihood reduced to −72.62, and the Sp2 model likelihood increased from −76.12 to −67.15. This is expressed in Table 1 in terms of log-likelihood ratio as an increase from −5.59 to +5.47. The transformation was not as successful for every speaker combination. For instance after conversion from Sp3 to Sp1 the likelihoods showed smaller differences ($\theta_{31}$: $-3.57 \rightarrow +0.47$). However, in all cases the likelihoods moved significantly in the right directions for source and target speakers (i.e., away from the source speaker, and towards the target speaker).

In Fig. 7, the illustration of the algorithm performance using speaker likelihood criterion on a sample test utterance is shown. Here, it can be seen that the voice conversion performance also depends on the context, and for some phonemes it is more successful, whereas it does not perform as well for others. Part of this can be explained by the fact that the same VQ indices are not forced to be used in speaker ID system, and another mixture combination from the source speaker may represent the target speaker characteristics in some cases. In order to eliminate this problem, we developed another objective measure which is described in Section 3.1.2.

[7] Speech files available. See www.elsevier.nl/locate/specom.

### 3.1.2. Sentence HMM-based objective evaluation

The second method uses sentence HMM alignment to force align the target utterance and the mimic utterance. Based on the state alignments the acoustic features corresponding to the same state indices for the target and mimic speech signals are compared. However, one drawback of this new scheme is that it requires access to target speaker's speech for the test utterance. We performed a series of experiments to measure the performance of both the sentence HMM-based and phonetic STASC algorithms in terms of objective measures in all the dimensions of speech that are considered in this paper: vocal tract spectrum, excitation spectrum, duration, F0, and RMS energy. We had a total of 15 min of speech from each of the three speakers (2 males, 1 female) with the same text. We used one of the male speakers as the source speaker, and trained codebooks for transformation to the other two speakers. We set aside 5 min of speech from the source speaker for testing. The remaining 10 min were used in the training. In order to understand the affect of the size of the training data on the performance of the STASC algorithm we used different duration lengths for training ranging from 10 to 600 s. In this experiment, since the speakers read the same text, we were able to use sentence HMM-based alignments to generate the codebooks. For comparison we also generated triphone codebooks based on phonetic alignments. The results are shown in Fig. 8. In the figure, the measures that correspond to time 0 show the distance between the unprocessed speech and target speech. The smaller distances imply better mimicking of the target speaker. Dark lines in each plot correspond to the sentence HMM based STASC, and light lines show the phonetic STASC algorithm performance. Each of the objective measures represented in the figure are described below.

*Cepstrum distance*. Average distance between the LPC-derived cepstrum vectors (first 8 cepstrum coefficients) of target and mimic utterance states:

$$d_{\text{ceps}} = \frac{1}{S}\sum_{s=1}^{S}(\boldsymbol{c}_s^{\text{t}} - \boldsymbol{c}_s^{\text{m}})^{\text{T}}\Sigma^{-1}(\boldsymbol{c}_s^{\text{t}} - \boldsymbol{c}_s^{\text{m}}), \tag{24}$$

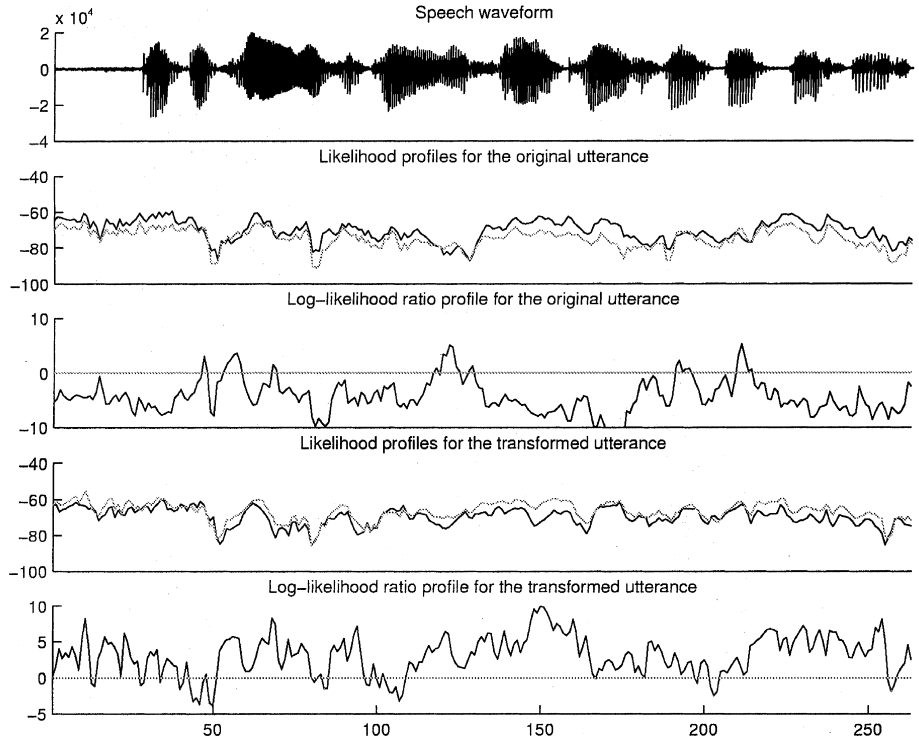


Fig. 7. Illustration of speaker conversion algorithm performance in terms of speaker ID system likelihoods across time (solid line: source speaker likelihood; light line: target speaker likelihood).

where $S$ is the total number of states in all sentence HMMs trained on all test utterances of the target speaker, $\boldsymbol{c}_s^{\mathrm{t}}$ and $\boldsymbol{c}_s^{\mathrm{m}}$ denote the average target and mimic cepstrum coefficient vectors, respectively for the $s$th state, and $\Sigma^{-1}$ is the global diagonal covariance matrix of all cepstrum vectors.

*Excitation spectrum distance*. Root mean square error difference between the normalized LPC residual spectra of target and mimic utterances.

$$d_{\mathrm{exct}} = \frac{1}{S}\sum_{s=1}^{S}\sqrt{\frac{1}{256}\sum_{k=1}^{256}(\boldsymbol{f}_s^{\mathrm{t}}(k) - \boldsymbol{f}_s^{\mathrm{m}}(k))^2}, \tag{25}$$

where $\boldsymbol{f}_s^{\mathrm{t}}$ and $\boldsymbol{f}_s^{\mathrm{m}}$ denote average normalized target and mimic excitation magnitude spectra for the $s$th state, respectively. Normalization of the spectra was performed prior to comparison in order to minimize the effect of gain level differences (i.e., $\sum_{k=1}^{256}\boldsymbol{f}^{\mathrm{t}}(k) = 1$ and $\sum_{k=1}^{256}\boldsymbol{f}^{\mathrm{m}}(k) = 1$).

*RMS energy distance* (dB). Average frame energy difference between target and mimic utterance states in dB.

$$d_{\mathrm{rms}} = \frac{1}{S}\sum_{s=1}^{S}(10\ \log_{10} E_s^{\mathrm{t}} - 10\ \log_{10} E_s^{\mathrm{m}}), \tag{26}$$

where $E_s^{\mathrm{t}}$ and $E_s^{\mathrm{m}}$ denote the target and mimic RMS energy for the $s$th state, respectively. Average gain levels for the target and mimic utterances are equalized prior to RMS energy estimation.

*F0 distance* (Hz). Average F0 difference between voiced states of target and mimic utterances.

$$d_{\mathrm{F0}} = \frac{1}{S_{\mathrm{v}}}\sum_{s=1}^{S_{\mathrm{v}}}(\mathrm{F0}_s^{\mathrm{t}} - \mathrm{F0}_s^{\mathrm{m}}), \tag{27}$$

where $S_{\mathrm{v}}$ is the total number of voiced states. $\mathrm{F0}_s^{\mathrm{t}}$ and $\mathrm{F0}_s^{\mathrm{m}}$ denote the target and mimic F0 values for the $s$th voiced state, respectively.

*Duration distance* (s). Average difference in corresponding sentence HMM state durations for target and mimic utterances.

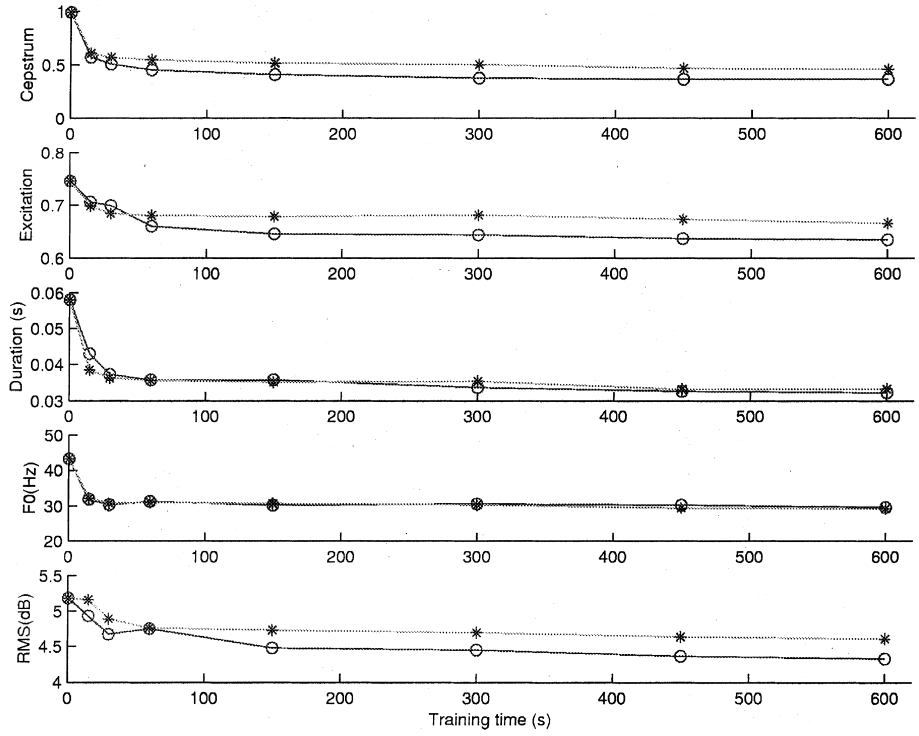


Fig. 8. Performance of the sentence HMM-based and phonetic STASC algorithms in terms of objective measures. Horizontal axis corresponds to the training duration. Vertical axis corresponds to the distance metric between mimic and target utterances in each of the 5 acoustic dimensions considered. The measures 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.

$$d_{\mathrm{dur}} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{3}\sum_{k=s-1}^{s+1}(\mathrm{dur}_k^{\mathrm{t}} - \mathrm{dur}_k^{\mathrm{m}}), \tag{28}$$

where $\mathrm{dur}_k^{\mathrm{t}}$ and $\mathrm{dur}_k^{\mathrm{m}}$ denote the target and mimic state durations for the $k$th state, respectively. Here each state duration is estimated based on the average of 3 states (i.e., each state together with its two neighbours) in order to reduce the effect of early/late transitions between states in the Viterbi alignment.

In general, sentence HMM-based STASC performs better when compared to phonetic STASC. The average acoustic distance between the transformed and target utterances decreases as more training data is used for STASC transformation. For example the average cepstral distance between original source speaker utterances and target speaker utterances reduced from 0.99 to 0.46 after 10 min of training data were used for the phonetic STASC algorithm. The sentence HMM-based STASC reduced the cepstral distance further down to 0.37. There is noticeable performance difference between the two methods in terms of cepstrum, excitation spectrum, and RMS energy parameters. In terms of duration and F0 parameters both methods performed at the same level. We can hear [8] on Signal 2a and 2b the original source and male target speaker utterances, respectively. Signal 2c corresponds to the output of the sentence HMM-based STASC algorithm after 10 min of training data were used. Here, a gradual transformation is applied in order to be able to illustrate another possible application of the STASC algorithm (i.e., voice morphing). As can be seen in Fig. 8 the performance of the algorithm

[8] Speech files available. See www.elsevier.nl/locate/specom.

levels off after several minutes of training data are provided. In fact, the F0 distance for both methods did not improve after 30 s of training data were provided. This is largely due to the fact that the degrees of freedom for the F0 model were very limited (mean and variance), and the model did not use any context information. Therefore, F0 is the most significant acoustic dimension where the STASC algorithm can benefit from further developments.

### 3.2. Subjective tests

We performed two listening experiments to test the performance of the STASC algorithm. The first test, forced choice ABX test, was designed to test how convincing the mimic was. The second test, intelligibility test, was performed to investigate whether the proposed algorithm was causing any degradation in intelligibility.

#### 3.2.1. Forced choice ABX test

In this experiment, we presented 20 stimuli A, B and X, and then asked, "is X perceptually closer to A or to B in terms of speaker identity?" A and B were short training utterances from source and target speakers, respectively (the order of assignment was randomized). X was the result of sentence HMM-based transformation from speaker A to speaker B. 10 stimuli were obtained from a male-to-male transformation, and the other 10 stimuli were obtained from a male-to-female transformation. Ten minutes of training data were used in both transformations. Each stimulus consisted of 2–3 word phrases from unseen test data. We used three inexperienced listeners in the experiments. The results are presented in Table 2. For the male-to-female transformation 100% of the time the listeners identified the transformed speech to be closer to the target speaker. As expected, the performance of the STASC algorithm was better for the male-to-female transformation (100%) when compared to the male-to-male transformation (78%).

Table 2
Forced-choice ABX listening test results (average number of times the listeners identified the transformed speech to be closer to the target speaker than to the source speaker)

| Test case | Number of times target is selected |
|---|---|
| Male → Female | 100% |
| Male → Male | 78% |

#### 3.2.2. Intelligibility test

While informal listening tests showed that the transformation of speaker characteristics was convincing, we wanted to test whether the transformation process introduced a degradation in intelligibility. This was necessary, since the most important application (i.e., text to speech personification) relies heavily on the level of intelligibility. The test material was 150 short nonsense sentences. One example of the sentences used in the test was "Shipping gray paint hands even". The main purpose of using nonsense sentences was to limit the ability of the listener to anticipate words based on context. Two conditions, transformed speech and natural speech, were presented to the listeners with random order. We used three inexperienced listeners to transcribe the words of the test material. Listeners were allowed to listen to each sentence up to three times. The standard NIST scoring algorithm (available through LDC from DARPA resource management continuous speech database) was then used to compare the utterance and transcribed phone strings. The phone sequences were determined by dictionary look-up. The transformation tested in this experiment was from a male speaker to another male speaker. The result of the experiment was surprising. The phone accuracy for the transformed speech (93.8%) was slightly higher than it was for natural speech (93.4%). The reason for the slight increase in intelligibility might be due to measurement noise. Another possible reason might be that the target speaker was more intelligible than the source speaker, and the transformation algorithm took advantage of that. Of course, the transformation between different speaker combinations may reveal different results. When the acoustic characteristics of two speakers are extremely different (e.g., male to female transformation), we may expect degradation in intelligibility. Our future plans include testing other speaker combinations.

## 4. Conclusion

In this study, a new voice conversion algorithm is developed. The algorithm is based on codebook mapping idea, however it uses a weighted average of codewords to represent each speech frame which results in smoother transition across successive frames. Both spectral and prosodic characteristics are transformed within the same framework which makes the algorithm computationally tractable. Part of the database used for training the codebooks did not have the same text across different speakers. The speakers were selected to be far apart from each other in terms of their speaking styles and acoustic spectrum. In spite of these factors, high quality speech which characterizes the target speaker was obtained after the STASC algorithm was employed for voice conversion. The performance of the algorithm was tested by objective tests and subjective listening tests. The objective evaluations verified that the target speaker characteristics are captured to a large extent after the STASC algorithm is employed. The subjective evaluations verified that convincing and high quality voice conversion can be achieved with the proposed algorithm.

## References

Abe, M., Nakamura, S., Shikano, K., Kuwabara, H., 1988. Voice conversion through vector quantization. In: Proceedings IEEE ICASSP, pp. 565–568.

Acero, A., 1993. Acoustical and Environmental Robustness in Automatic Speech Recognition. Kluwer Academic Publishers, Dordrecht.

Arslan, L.M., Talkin, D., 1997. Voice conversion by codebook mapping of line spectral frequencies and excitation spectrum. In: Proceedings EUROSPEECH, Rhodes, Greece, Vol. 3, pp. 1347–1350.

Arslan, L.M., McCree, A., Viswanathan, V., 1995. New methods for adaptive noise suppression. In: Proceedings IEEE International Conference on Acoust., Speech, Signal Processing, Detroit, USA, Vol. 1, pp. 812–815.

Baudoin, G., Stylianou, Y., 1996. On the transformation of the speech spectrum for voice conversion. In: Proceedings ICSLP, Philadelphia, USA, pp. 1405–1408.

Childers, D.G., 1995. Glottal source modelling for voice conversion. Speech Communication 16 (2), 127–138.

Crosmer, J.R., 1985. Very low bit rate speech coding using the line spectrum pair transformation of the LPC coefficients. Ph.D. Thesis, Elec. Eng., Georgia Inst. Technology.

Hansen, J.H.L., Clements, M.A., 1991. Constrained iterative speech enhancement with application to speech recognition. IEEE Trans. on Signal Processing 39 (4), 795–805.

Itakura, F., 1975. Line spectrum representation of linear prediction of speech signals. J. Acoust. Soc. Amer. 57 (S35(A)).

Iwahashi, N., Sagisaka, Y., 1995. Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks. Speech Communication 16 (2), 139–151.

Kuwabara, H., Sagisaka, Y., 1995. Acoustic characteristics of speaker individuality: Control and conversion. Speech Communication 16 (2), 165–173.

Laroia, R., Phamdo, N., Farvardin, N., 1991. Robust and efficient quantization of speech LSP parameters using structured vector quantizers. In: Proceedings IEEE ICASSP, pp. 641–644.

Lee, K.S., Youn, D.H., Cha, I.W., 1996. A new voice transformation method based on both linear and nonlinear prediction analysis. In: Proceedings ICSLP, Philadelphia, USA, pp. 1401–1404.

Moulines, E., Charpentier, F., 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Communication 9 (5/6), 453–467.

Narendranath, M., Murthy, H.A., Rajendran, S., Yegnanarayana, B., 1995. Transformation of formants for voice conversion using artificial neural networks. Speech Communication 16 (2), 207–216.

Paliwal, K.K., 1995. Interpolation properties of linear prediction parametric representations. In: Proceedings of the Fourth European Conference on Speech Communication and Technology (EUROSPEECH 95), Madrid, Spain.

Pellom, B.L., Hansen, J.H.L., 1997. Spectral normalization employing hidden Markov modeling of line spectrum pair frequencies. In: Proceedings of the IEEE International Conference on Acoust., Speech, Signal Processing, Vol. 2, Munich, Germany, pp. 943–946.

Stylianou, Y., Laroche, J., Moulines, E., 1995. High-quality speech modification based on a harmonic plus noise model. In: Proceedings EUROSPEECH, Madrid, Spain.

Wightman, C., Talkin, D., 1994. The Aligner User's Manual. Entropic Research Laboratory, Washington, DC, USA.