International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Voice Conversion Based On Hidden Markov Model

Arpitha.D[1], Chaitra.C.N[2], Manasa.M[3], Chaitra.N[4]
Department of ECE, BNMIT Bangalore, India [1,2,3]
Assistant Professor, Department of ECE, BNMIT Bangalore, India [4]

Abstract

Voice morphing which is also referred to as voice transformation and voice conversion is a technique to modify a source speaker's speech utterance to sound as if it was spoken by a target speaker. There are many applications which may benefit from this sort of technology. In the paper, we propose a new method of voice conversion which uses Hidden Markov Model (HMM) for the training. HMM is used to represent the phonetic structure of training speech and to generate the training pairs of source and target speakers by mapping the HMM states between source and target speeches. Then, HMM codebook is generated to create the mapping function for the voice conversion.

Keywords

Voice conversion, MFCC, Hidden Markov model, Codebook

1. Introduction

Voice conversion modifies the voice produced by a source speaker to be perceived by listeners as the voice of a different speaker, the target speaker. There are basically three inter-dependent issues that must be solved before building a voice morphing system. Firstly, it is important to develop a mathematical model to represent the speech signal so that the synthetic speech can be regenerated and prosody can be manipulated without artifacts. Secondly, the various acoustic cues which enable humans to identify speakers must be identified and extracted. Thirdly, the type of conversion function and the method of training and applying the conversion function must be decided. By far, most of current voice conversion methods are based on text-dependent corpus, which means the training set has to be extracted from parallel utterances of both source and target speakers. However, the preparation of parallel speech database is very inconvenient in real-life applications. In addition, it is unlikely to have parallel utterances of both source and target speakers in some applications, e.g. cross-lingual voice conversion. (The cross lingual voice conversion problem refers to the replacement of a speaker's timbre or vocal identity in a recorded sentence, assuming that the source speaker and target speaker use different languages. This problem differs from typical voice conversion in the sense that the mapping of acoustical features cannot depend on time-aligned recordings of source and target speakers uttering the same sentences.) Thus, voice conversion systems must be able to identify the individual characteristics of the source speaker’s voice and replace them with those of the target speaker’s voice without losing information or modifying the message that is being transmitted. Voice conversion has a wide variety of applications, including the design of multi-speaker speech synthesis systems, the customization of speaking devices, the design of speaking aids for people with speech impairments, film dubbing using the original actors’ voices,the creation of virtual clones of famous people for videogames, and masking identities in chat rooms.

A new branch of voice conversion has grown over the last decade around speech synthesis and conversion systems based on hidden Markov models (HMMs) [4]. Given an input speech signal and a previously trained HMM set, such systems return sequence of parameter vectors. In this case, voice conversion is performed by adapting the HMMs themselves to the target speaker [1] [3]. Hidden Markov Model (HMM) has been successfully applied to speech recognition systems for its excellent ability of characterizing the spectral parameter sequence and modeling phonetic structure[2]. When modeling speech, each HMM corresponds to a phonetic unit with phonetic signification. Considering these facts, the HMM is used to represent the phonetic structure of training speech. The transformation between source and target characteristics is accomplished by establishing a mapping between the source and target HMM states from the generated training pairs. Model similarity is used to measure the correlation between source and target states for state alignment. Then, HMM state mapped codebooks are generated to create the mapping function for the voice conversion.

The training of parameters of speech signal gives better results when compared to training of speech signal itself. One such technique to represent the speech parameters is MFCC and the same is used in this paper. The Mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.

The paper is organized as follows. In Section 2, new technique of voice conversion based on HMM using codebook is derived. Experimental results are presented in Section 3. Finally, conclusion is drawn in Section 4 and the Section 5 gives future scope.

2. Proposed Method

Voice morphing is a conversion from one speaker's voice to another speaker's voice. For example, when speaker A utters something, morphing makes it to be perceived as if it was uttered by speaker B. In this paper, we make use of HMM to train the speech samples. Then for a given source test sample, we achieve conversion with the help of a codebook for mapping.

The proposed method uses the block diagram shown in Fig.1 to perform voice conversion. It consists of two stages, namely the training and conversion.

2.1. Training Part

A single database has been considered to store both the source and the target speech samples.

2.1.1 Noise removal

This is used to remove the silence period and the background noise of a speech sample.

Here, we first calculate the mean and the standard deviation of a portion of the speech signal. This is done to characterize the background noise [5]. Then, from the first to the last sample of the recorded signal, the Mahalanobis distance of each sample i.e. $|x-\mu|/\sigma$ is checked with a pre-determined threshold value. If it is greater, we treat it as a voiced sample otherwise as an unvoiced sample. All the voiced samples are marked as '1' and the unvoiced as '0'. The speech signal is divided into non overlapping windows of convenient size (e.g. 10ms).The numberof 1's and 0's in each window is compared. If the number of 1's is greater than the 0's then the window is marked as '1'. We consider the voiced part according to the window marking and discard the unvoiced.

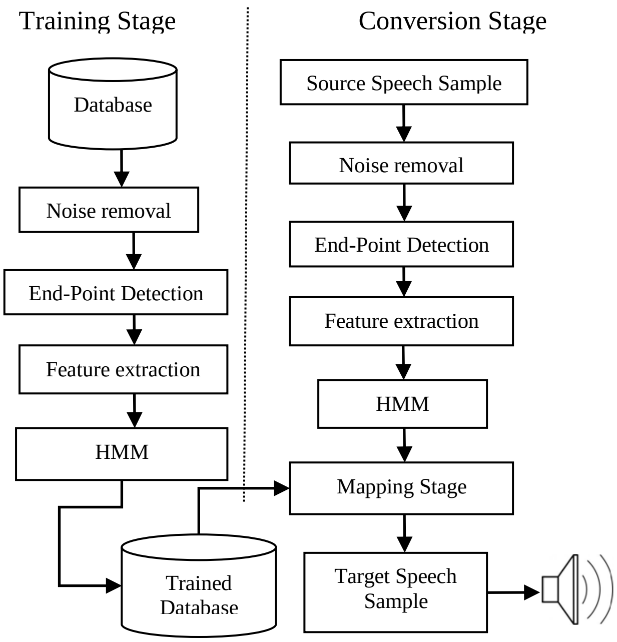

Fig. 1: Block diagram for voice conversion

2.1.2 End point Detection

The algorithm described below is divided into two parts.

1. Find out the maximum signal value in the signal. Divide all the samples in the signal with the max value to normalize the speech signal.
2. Set the threshold value 'Th', which is used to find the end points of the signal.

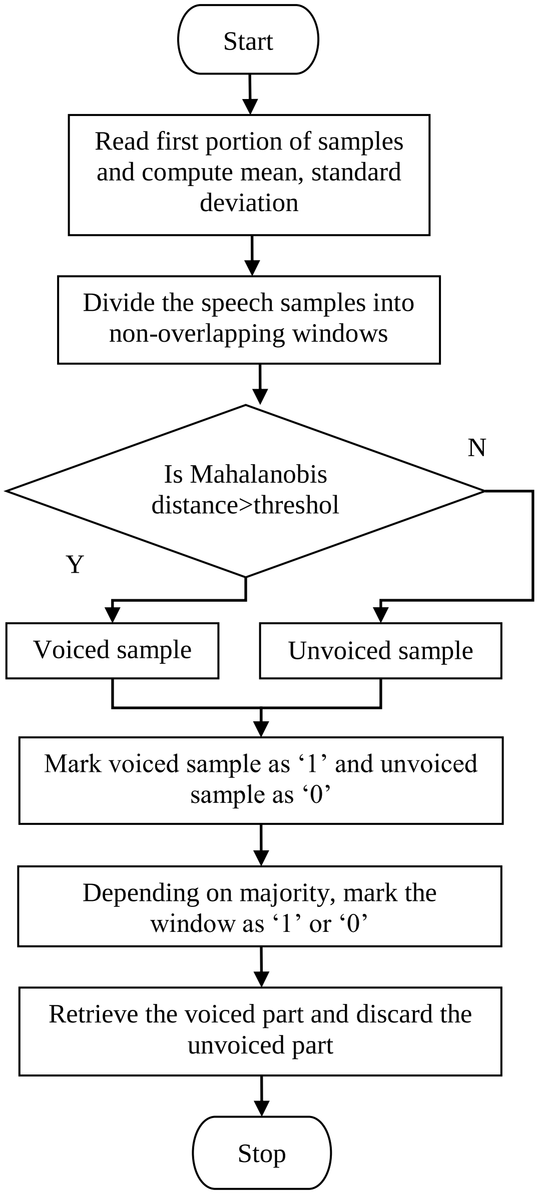
Start
Read first portion of samples and compute mean, standard deviation
Divide the speech samples into non-overlapping windows
Is Mahalanobis distance>threshol
N
Y
Voiced sample
Unvoiced sample
Mark voiced sample as '1' and unvoiced sample as '0'
Depending on majority, mark the window as '1' or '0'
Retrieve the voiced part and discard the unvoiced part
Stop

Fig. 2: Flowchart for noise removal

Step 1: Find the number of samples present in the signal for which end point detection should be performed.

Step 2: For fixed frame sizes calculate the number of frames that can be extracted.

No. of frames = No. of samples/frame size (1)

Step 3: Check if the No. of frames is an integer. If not then convert it to an integer and perform zero padding appropriately. If it is an integer then continue with the next step.

Step 4: Find the new length of the signal after zero padding.

Step 5: Divide the signal into frames with the frame size as defined before.

Step 6: Find the energy of each frame.

$$E = \Sigma \mid \text{signal frame} \mid^2 \qquad (2)$$

Step 7: Find the maximum of the energy value and normalize the energy spectrum.

$$E=E/\max \qquad (3)$$

Step 8: Set a threshold value and find the starting point using it. If the energy level of a particular sample is greater than the threshold then that is considered as the starting point.

Step 9: Once the starting point is obtained, reverse the energy spectrum and find the ending point similarly as starting point.

Step 10: Reconstruct the signal by considering the frames between starting and ending point.

2.1.3 MFCC (Mel-scale Frequency Cepstral Coefficient)

For speaker recognition, the most commonly used acoustic features are MFCC**.** Feature extraction is done to extract the relevant information from the speech signal which provides a good model of the speech signal. These features give a fairly good representation of the vocal tract characteristics. MFCC takes human perception sensitivity with respect to frequencies into consideration, and therefore are best for speaker recognition. MFCC of a speech signal is obtained using the process shown in Fig.4.

Framing: The input speech signal is segmented into frames with optional overlap. Usually the frame size (in terms of sample points) is equal to power of two in order to facilitate the use of FFT. If this is not the case, we need to do zero padding to the nearest length of power of two.

Windowing: Each frame has to be multiplied with a window in order to keep the continuity of the first and the last points in the frame.

Fast Fourier Transform (FFT): Spectral analysis shows that different timbres in speech signals correspond to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame. When we perform FFT on a frame, we assume that the signal within a frame is periodic, and continuous when wrapping around.

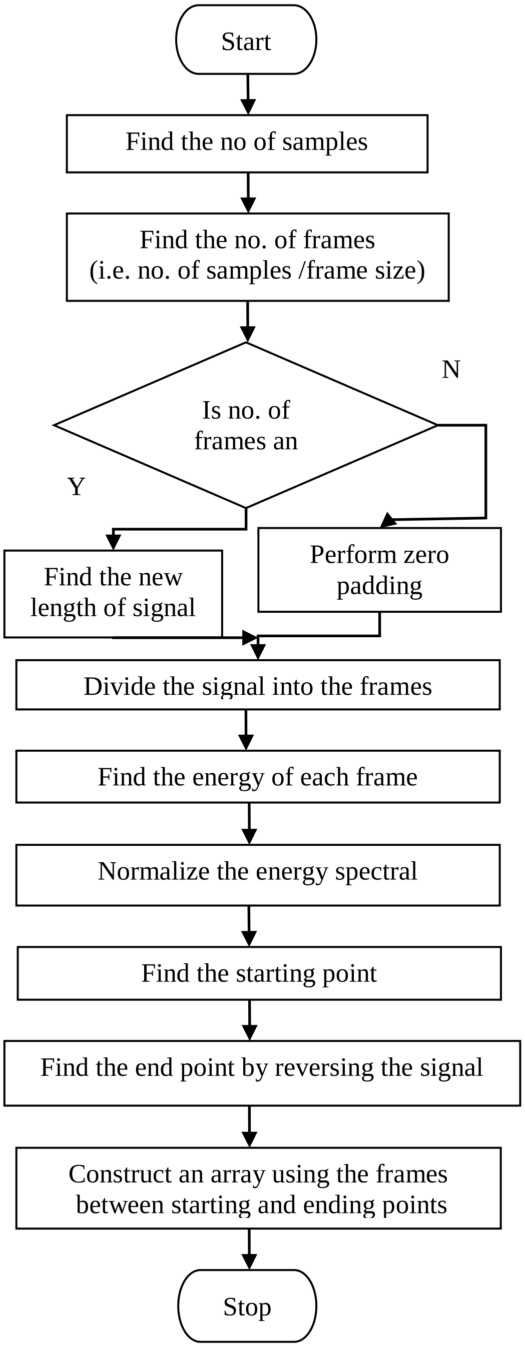
Start
Find the no of samples
Find the no. of frames
(i.e. no. of samples /frame size)
Is no. of
frames an
N
Y
Find the new
length of signal
Perform zero
padding
Divide the signal into the frames
Find the energy of each frame
Normalize the energy spectral
Find the starting point
Find the end point by reversing the signal
Construct an array using the frames
between starting and ending points
Stop

Fig.3: Flowchart for end-point detection

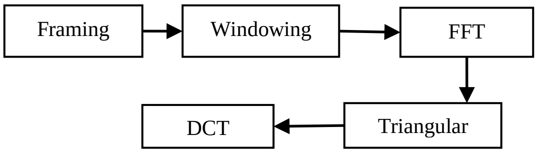

Fig. 4: Feature extraction using MFCC

If this is not the case, we can still perform FFT but the discontinuity at the frame's first and last points is likely to introduce undesirable effects in the frequency response. To deal with this problem, we have two strategies:

a. Multiply each frame by a Hamming window to increase its continuity at the first and last points.
b. Take a frame of a variable size such that it always contains an integer multiple number of the fundamental periods of the speech signal.

Triangular Bandpass Filters: We multiply the magnitude frequency response by a set of 20 triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel frequency, which is related to the common linear frequency ‘f’ by the following equation:

$$mel\ (f) = 1125*\ln(1+f/700) \qquad (4)$$

Mel-frequency is proportional to the logarithm of the linear frequency, reflecting similar effects in the human's subjective aural perception.

Discrete cosine transform or **DCT**: In this step, we apply DCT on the log energy ‘E_k’ obtained from the triangular bandpass filters to have ‘L’ mel-scale cepstral coefficients. The formula for DCT is shown below

$$C_m = S_{k=1}^{N} \cos\ [m*(k-0.5)*p/N]*E_k,\ m=1,\ 2,\ L \quad (5)$$

Where, N is the number of triangular bandpass filters, L is the number of mel-scale cepstral coefficients. Usually we set N=20 and L=12. Since we have performed FFT, DCT transforms the frequency domain into a time-like domain called quefrency domain. The obtained features are similar to cepstrum, thus it is referred to as the mel-scale cepstral coefficients, or MFCC. MFCC alone can be used as the feature for speech recognition. For better performance, we can consider additional features.

2.1.4 Hidden Markov Model (HMM)

HMM training is done to assign the extracted feature vectors to the HMM states. This can be done in two steps

i. We use the ‘Baum-Welch’ algorithm for training. It basically assigns at what probability a feature vector is emitted from a HMM state. We first assign the initial probabilities to all the model parameters, until the training converges, it adjusts the probabilities of the HMM parameters.

ii. We use 'Viterbi' algorithm to assign a feature vector to a particular state. Given a sequence, this algorithm calculates the most likely path through the Hidden Markov model specified by the model parameters.

The HMM parameters of all the speech signals are stored in the trained database.

2.2 Conversion Part

The source test sample is given as an input. The representation phonetic structure is same as in the training part.

2.2.1 Mapping Stage

This stage performs two tasks. First we need to recognize the speech sample from the trained database. The states sequence of the test sample is compared with the state sequences in the codebook. This is done using Euclidean distance. The second task is to morph using a mapping function.

3. Experimental Results

We have considered different morphing scenarios. They are:

Phoneme	Word
1. Female to Female	5. Female to Female
2. Female to Male	6. Female to Male
3. Male to Male	7. Male to Male
4. Male to Female	8. Male to Female

For phoneme database we have considered 5 speakers and 10 phonemes with only one utterance for each speaker. The following are the phonemes considered

1. 'a' as in bat
2. 'ai' as in bait
3. 'ee' as in keep
4. 'i' as in bit
5. 'o' as in bob
6. 'b' as in bet
7. 'oo' as in book
8. 'p' as in pet
9. 't' as in ten
10. 'uy' as in buy

For word database we have considered 5 speakers and 5 words with five utterances for each speaker. The following are the words considered

1. One
2. Two
3. Three
4. Four
5. Five

3.1 Female to Female Phoneme

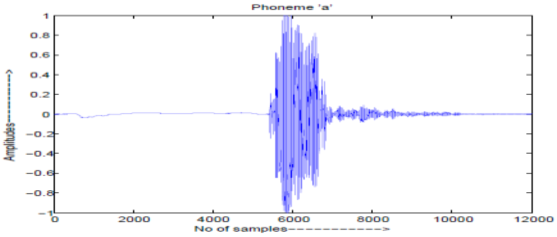

Fig.5a: Original signal

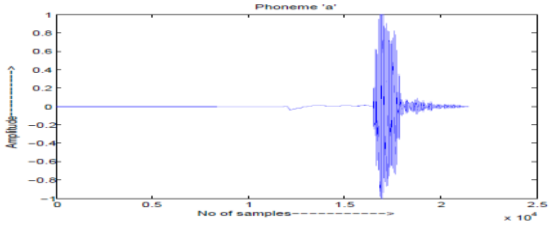

Fig.5b: Speech signal after Background Noise removal

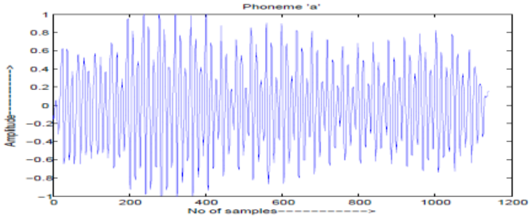

Fig.6a: Speech signal after End Point Detection

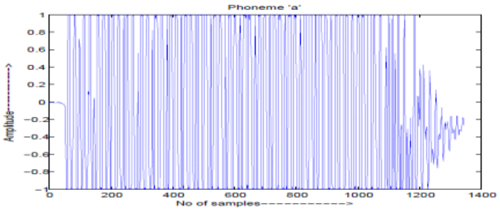

Fig.6b : Morphed signal

3.2 Male to Female word

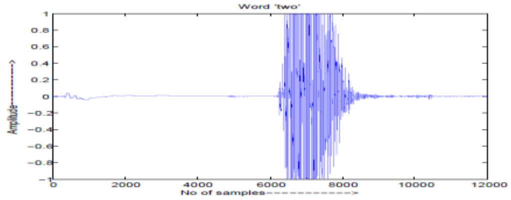

Fig.7a: Original signal

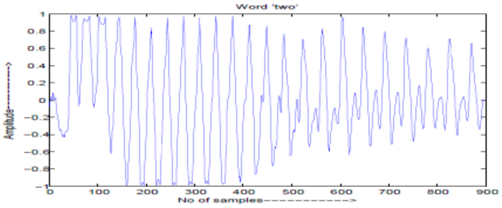

Fig.7b : Morphed signal

4. Conclusion

In this paper, we proposed a new method for voice conversion based on codebook technique. This approach uses HMM to present the phonetic structure of training speech and achieve the conversion by mapping states between source and target HMMs. Here we make use of end point detection for background noise removal. Feature extraction is done by obtaining the MFCC. The experimental results prove that this approach is indeed efficient.

5. Future Scope

In this method, transformation needs speech samples of both the speakers. The work can be extended to a database which contains phonemes of different languages. This would enable cross lingual transformation. As a further step, the same principles of HMM codebook mapping for voice modeling can be used to make the morph text independent by concatenating the phonemes of a certain language. The HMM models for voice are strong enough to allow for text independent morphing, which entails HMM models for the speech characteristics rather than phonemes. The concept of real time processing is of much relevance. The appropriate models for noise suppression and for voice timbre can lead to real time morph. Such a morph will be a breakthrough.

References

[1] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," IEEE Trans. Audio, Speech, Lang. Process., vol. 17, no. 1, pp. 66–83, Jan. 2009.

[2] Text-Independent Voice Conversion Based On State Mapped Codebook .Meng Zhang, Jianhua Tao, Jilei Tian, Xia Wang ICASSP 2008.

[3] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Voice characteristics conversion for HMM-based speech synthesis system," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 1997, pp. 1611–1614.

[4] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis using HMM with dynamic features," in Proc. IEEE Int. Conf. Acoust.,Speech, Signal Process., 1996, pp. 389–392.

[5] A New Silence Removal and Endpoint Detection Algorithm for Speech and Speaker Recognition Applications G. Saha, Sandipan Chakroborty, Suman Senapati.

[6] Rabiner, Lawrence R., and Ronald W. Schafer. Digital processing of speech signals. Vol. 100. Englewood Cliffs: Prentice-hall, 1978.

[7] "Speech and Audio Signal processing", Ben Gold and Nelson Morgan, John Wiley, 2007 edition.

[8] Chapman, Stephen. Essentials of MATLAB programming. Cengage Learning, 2008.

Arpitha.D, was born on 18-Dec-1990. She graduated as a Bachelor of Engineering from BNMIT.

Manasa.M was born on 1-Feb-1991. She graduated as a Bachelor of Engineering from BNMIT.

Chaitra.N was born on 20-Aug-1980. She graduated as a Bachelor of Engineering BMSCE.She did her M.tech in RVCE.She is currently working as an Assistant Professor in BNMIT.
.

Chaitra.C.N. was born on 15-Apr-1991. She graduated as a Bachelor of Engineering from BNMIT.