

# “SYNCPITCH”: A PSEUDO PITCH SYNCHRONOUS ALGORITHM FOR SPEAKER RECOGNITION

Ran D. Zilca, Jiri Navratil, and Ganesh N. Ramaswamy  
IBM T. J. Watson Research Center  
Yorktown Heights, NY  
{zilca, jiri, ganeshr}@us.ibm.com

## ABSTRACT

Pitch mismatch between enrollment and testing is a common problem in speaker recognition systems. It is well known that the fine spectral structure related to fundamental frequency manifests itself in Mel cepstral features used for speaker recognition. Therefore pitch variations result in variation of the acoustic features, and potentially an increase in error rate. A previous study introduced a signal processing procedure termed *depitch* that attempts to remove pitch information from the speech signal by forcing every speech frame to be pitch synchronous and include a single pitch cycle. This paper presents a modification of the depitch algorithm, termed *syncpitch*, that performs pseudo pitch synchronous processing while still preserving the pitch information. The new algorithm has a relatively moderate effect on the speech signal. System combination of syncpitch with a baseline system is shown to improve speaker verification accuracy in experiments conducted on the 2002 NIST Speaker Recognition Evaluation data.

## 1 INTRODUCTION

Most current speaker recognition systems use Mel Frequency Cepstral Coefficients (MFCC) and their time derivatives as the signal processing front end. MFCC are extracted from the power spectrum of the speech signal, which includes fundamental frequency harmonics [1, 2]. The presence of the pitch harmonics in the power spectrum depends on the value of the pitch frequency, the number of pitch cycles in a single frame, and the number of samples that belong to a partial pitch cycle in a frame (i.e. truncated pitch cycle). Apart from the pitch frequency value, the above factors are artifacts introduced by the decision to process the speech signal frame by frame, rather than genuine properties of the signal. In speaker recognition evaluations conducted by the National Institute of Standards and Technology (NIST) in recent years, it was found that the performance of automatic speaker recognition systems degrades with higher pitch frequency, and with stronger “pitch mismatch” (i.e. pitch variation between enrollment and testing) [3, 4, 5]. This degradation may be attributed to the presence of this pitch-related harmonic fine structure and artifacts. Previous work [6] has proposed a signal-domain algorithm termed *depitch* that suppresses the presence of fundamental frequency information in the speech signal, and results in a power spectrum that is free from the typical pitch-related harmonic structure. In addition to reducing the presence of pitch

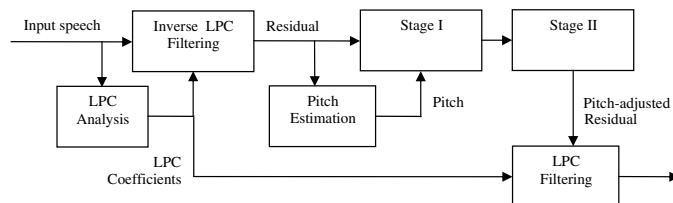


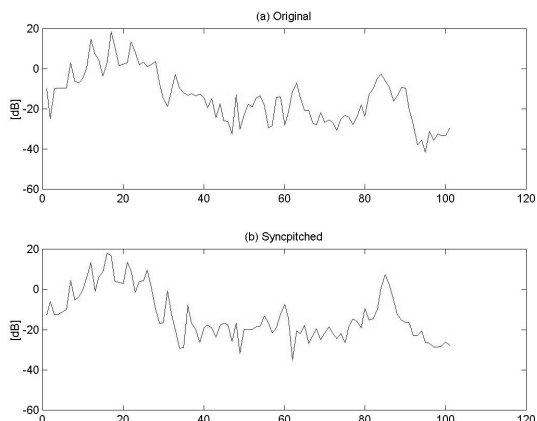
Figure 1. Block Diagram of the Depitch and Syncpitch Methods

harmonics, the depitch procedure also results in pitch synchronous processing, since exactly one pitch cycle fits in a single processing frame. This is obtained extracting a single pitch cycle from the residual signal, and interpolating it to fit the frame size.

This paper describes a new pitch adjustment algorithm termed *syncpitch*. The new algorithm does not suppress the presence of pitch information in the speech signal, but does modify the signal in every frame to include an integer number of pitch cycles. This approach results in more accurate spectral estimation for voiced frames in the spectral analysis stage of the MFCC computation. The syncpitch procedure is designed to reduce the presence of the aforementioned artifacts.

## 2 The Syncpitch Algorithm

Syncpitch uses the same basic idea as depitch, where the pitch adjustment is performed in the residual domain rather than in the original waveform domain. A general block diagram that describes the two algorithms is shown in figure 1. The procedure is always performed independently for each individual frame. It accepts input speech and produces “syncpitched” or “depitched” time-domain speech. However, since speech frames typically overlap, the resulting time domain signal is not continuous. The input speech frame is first windowed and undergoes Linear Prediction analysis to estimate Linear Prediction Coefficients (LPC). We use a Hamming window in our experiments. The LPC coefficients are then used to calculate the residual signal by filtering the original speech using the inverse LPC filter. Pitch estimation follows, performed on the residual signal, using a variant of the autocorrelation method. The pitch estimation stage uses a nonlinear function prior to computing the autocorrelation function, in order to reduce the influence of noise. The function performs both saturation, and zero-out of values



**Figure 2. Effect of Syncpitch on Spectrum Example Number 1**

close to zero. Then, the pitch cycle is estimated to be the time difference between the main autocorrelation peak and the closest secondary peak [7]. This pitch estimation method also results in a voicing score, given by the ratio of the secondary to main peak energies in the autocorrelation function. The residual signal is then processed according to stages I and II to perform some form of pitch adjustment. Stages I and II are described below for depitch and syncpitch separately. Finally, the output of stage II (i.e. the “pitch adjusted” residual signal) is filtered using the LPC filter to synthesize a speech signal in the original waveform domain.

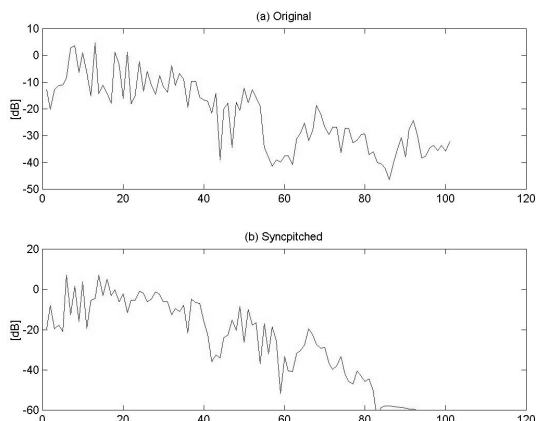
For the previously suggested depitch procedure [6], stage I involves extracting a single pitch cycle from the center of the frame, and stage II performs interpolation of the pitch cycle to fit exactly the size of one frame. For syncpitch, stages I and II are as follows:

**Stage I:** Calculate the maximal number of pitch cycles,  $n_{\max}$ , that would not exceed the frame size  $N$ . Then, extract  $n_{\max} \cdot p$  samples of the residual signal, where  $p$  is the estimated pitch cycle in samples.

**Stage II:** interpolate the  $n_{\max}$  samples to fit the size of one frame (up-sample from  $n_{\max}$  to  $N$ ). Perform a cyclic shift such that the edges have the lowest energy.

We note the following regarding the algorithms:

1. Stage II is lossy for the two algorithms, as some samples of the residual signal are ignored in every frame. However, since we use significant overlap between frames very few samples are eventually ignored in the entire signal.
2. Interpolation involves low pass filtering of the residual signal in stage II, resulting in some spectral distortion in the output speech spectrum. Depitch will typically involve more aggressive filtering comparing to syncpitch since  $n_{\max}$  is closer to  $N$  than a single pitch cycle. Loosely speaking, syncpitch is less lossy than depitch.



**Figure 3. Effect of Syncpitch on Spectrum Example Number 2**

3. Voicing detection is performed for each frame and only frames classified as “voiced” undergo the procedure. The voicing detection is performed by thresholding the score generated by the autocorrelation-based pitch detector.

4. The cyclic shift is meant to provide better continuity, for example for the purpose of listening. It is not meant to have any effect on the resulting features, as it changes only the phase and not the power spectrum. However, since the resulting processed frame is typically windowed, the shift does have some effect on the features.

### 2.1 The Effect on of Syncpitch on the Power Spectrum

The previous study [6] demonstrated that the presence of fine spectral structure related to the fundamental frequency can be removed using depitch. When syncpitch is used this fine structure is maintained. Yet, the spectral distortion resulting from having a truncated pitch cycle in the frame is reduced. Two examples of the syncpitch-processed power spectra of a single frame are shown in figures 2 and 3 respectively. From the figures it is clear that the pitch-related harmonic spectral component is still present. However, the power spectrum changes after syncpitch processing. It could be argued that the syncpitched spectrum exhibits clearer peaks. Also, as clearly seen in figure 3, the interpolation performed in stage II is lossy, and the signal is low-pass filtered. This is less evident for the frame shown in figure 2, where  $n_{\max} \cdot p$  was close to  $N$ , in a higher cut-off frequency of the interpolation low pass filter.

## 3 EXPERIMENTS

### 3.1 Experimental Configuration

Experiments were conducted using the NIST 2002 Speaker Recognition Evaluation corpus [4]. The task is text independent speaker verification on cellular telephony speech data. The dataset includes 330 target speakers and 39105 test trials (male and female). A detailed description of the 2002

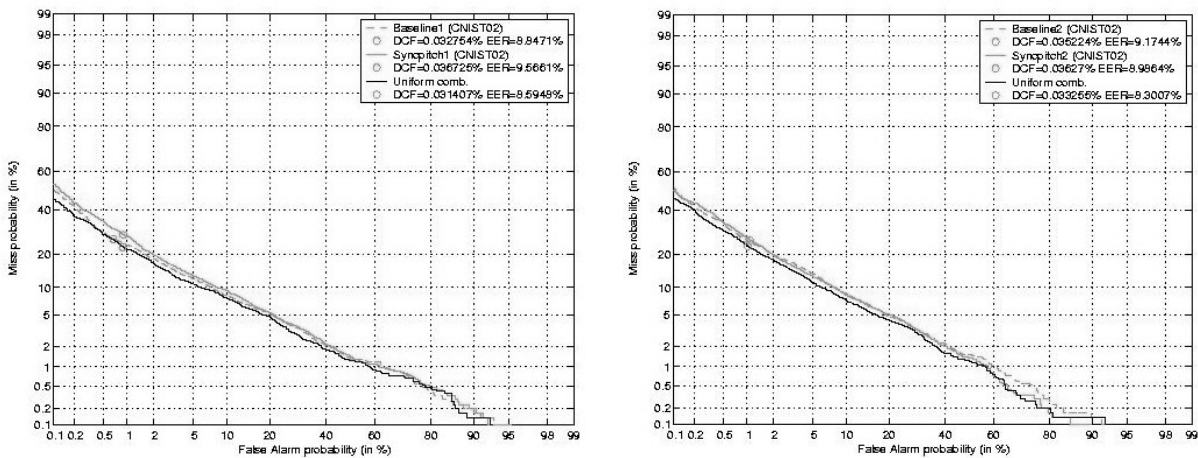


Figure 4. Performance of Syncpitch and Baseline Systems

NIST speaker recognition evaluation can be found in [4]. All systems share the following common attributes: A diagonal covariance Gaussian Mixture Model (GMM) with 2048 components, that is adapted for each speaker using MAP adaptation of the Gaussian means from a Universal Background Model (UBM). The Maximum Likelihood Linear Transform (MLLT) is also used [9]. For signal processing, 19 MFCC's and their time derivatives were extracted using 24 Mel filters from 25 msec frames with 40% overlap. The feature vectors were subject to feature warping [8] that transforms each individual element such that it is a standardized Gaussian within a 3 seconds window. Training data for the UBM's was different for each system, as described below. The UBM's were trained using the clustering procedure described in [9] followed by 12 EM iterations. A more detailed description of the signal processing and modeling details may be found in [10]. All systems use T-norm score normalization [11] with a set of 234 imposter models.

Four systems were implemented and tested for the purpose of testing the usefulness of the syncpitch method. Two systems serve as a baseline system, while the other two are identical except that all of the speech data is preprocessed using syncpitch (background, target, and testing):

**Baseline1:** UBM trained on the following:

1. NIST 2001 cellular development set (consisting mostly of GSM cellular calls recorded in the United States)
2. NIST 1996 evaluation set, consisting of landline telephone speech, run through three different GSM coders (full rate, half rate, enhanced half rate). The use of cellular coders on landline telephone speech is meant to match the background data to the expected training and test data [e.g. 10, 12].

**Syncpitch1:** Same as baseline 1, only syncpitched data (background, target, and test data).

**Baseline2:** UBM trained on the following:

1. NIST 2001 cellular development set, consisting mostly of GSM U.S. cellular calls
2. NIST 1996 landline evaluation set, run through a CDMA cellular coder (U.S. IS-95 standard).

**Syncpitch2:** Same as baseline 2, only syncpitched data (background, target, and test data).

### 3.2 Results

We measure the performance of the systems by their Equal Error Rate (EER) and Detection Cost Function (DCF). Both of them represent two points on the DET plot, or two different values of acceptance threshold. While the EER corresponds to the mid point of the DET plot, where the false rejection and false acceptance errors are the same, the DCF corresponds to a point in the upper-left (high security) region of the DET plot, and it gives more weight to false acceptance errors than false rejection errors. The definition of the DCF may be found in [4]. For convenience, all the DCF values in this paper were multiplied by 10<sup>3</sup>. A preliminary experiment on the 2002 NIST evaluation dataset (with a simple configuration that does not use T-norm [11]) found syncpitch to be worse than a corresponding baseline system (DCF/EER of 50.1/11.7% comparing to 42.2/10.7%). Adding T-norm score normalization to both baseline and syncpitch systems has closed some of this performance gap, as described in detail below. The results are shown in figure 4. Baseline1 obtains a DCF of 32.7 and an EER of 8.8%, while syncpitch1 performs slightly worse: 36.7 and 9.6%. However, the combination of the two systems (obtained by simple averaging the utterance level scores) provides a DCF of 31.4, and an EER of 8.6%, a 4% relative improvement in DCF, and 3% in EER. It is important to note that this improvement is achieved with no additional data, simply by processing all the

existing data using syncpitch. The baseline2 system obtains a DCF of 35.24, and EER of 9.2%. Syncpitch 2 has a slightly better EER, 9.0%, and a slightly worse DCF, 36.3. The simple average combination yields a relative improvement of 6% in DCF and 10% in EER (33.2, 8.3%). Again, the improvement is obtained using the combination without adding additional data. The fact that the baseline systems outperform the syncpitch systems may be attributed to the loss of high frequency energy in Stage II of the process (interpolation). However, a simple average combination was found to be beneficial, indicating that for some trials the syncpitch system outperforms the baseline system, and that the errors are relatively uncorrelated. It is probable that many errors of the syncpitch system are related to the interpolation loss described in section 2 above and that for those same trials the baseline systems perform better. On the other hand, it is also probable that many errors of the baseline systems are related to the spectral distortion that syncpitch is designed to reduce, resulting in error decorrelation between baseline and syncpitch.

#### 4 CONCLUSIONS

This paper describes the syncpitch algorithm that allows “pseudo pitch synchronous” speech processing in a system that uses a fixed frame size. In particular, we used the syncpitch algorithm for text independent speaker verification on cellular telephony speech. A previous study that used the depitch algorithm has found that removing pitch information hurts speaker verification performance, though it may alleviate the high false rejection encountered “goat speakers”. It was also found that for female speakers the use of depitch may cause errors to distribute more evenly across verification trials. The syncpitch algorithm presented in this contribution is a modification of the depitch algorithm that does not remove the pitch information, and was thus expected to provide better performance. However, the two tested syncpitch systems performed consistently worse across the DET curve comparing to their corresponding baseline systems. This finding implies that the number of errors caused by the lossy syncpitch procedure is higher than the number of errors that it reduces. More importantly, a simple average score combination between the syncpitch and baseline systems yielded a performance improvement of up to 10% relative, implying that the errors that syncpitch reduces are on different trials than the ones for which it creates errors.

The syncpitch procedure can be useful in an operational system, since it does not require additional data, and works at the frame level. Two UBM’s need to be used (original + syncpitched), and every frame for both training and verification needs to be generated twice (original + syncpitch). The results shown in this paper indicate a consistent improvement in accuracy. Using a combination of syncpitch with a conventional system on the 2002 NIST evaluation complete dataset, a best DCF of 31.4 and a best EER of 8.3% were obtained (for two separate systems). Future work will focus on modifying syncpitch to account for the lossy processing while interpolating the residual signal.

Also, the usefulness of the depitch and syncpitch methods for speech recognition tasks will be assessed.

#### REFERENCES

- [1] T. F. Quatieri, R. B. Dunn, and D. A. Reynolds, “On the Influence of Rate, Pitch, and Spectrum on Automatic Speaker Recognition Performance,” ICSLP 2000.
- [2] T. F. Quatieri, R. B. Dunn, D. A. Reynolds, J. P. Campbell, and E. singer, “Speaker Recognition using G.729 Speech CODEC Parameters,” ICASSP 2000, Istanbul.
- [3] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, “The NIST Speaker Recognition Evaluation – Overview, Methodology, Systems, Results, Perspective,” *Speech Communication*, Vol. 31, No. 2-3, pp. 225-254, June 2000.
- [4] <http://www.nist.gov/speech/tests/spk/>
- [5] A. Martin and M. Przybocki, “The NIST 1999 Speaker Recognition Evaluation – An Overview,” *Digital signal Processing*, Vol. 10, No. 1, pp. 1-18.
- [6] R. D. Zilca, J. Navratil, and G. N. Ramaswamy, “Depitch and the Role of Fundamental Frequency in Speaker Recognition,” *ICASSP 2003*, Hong Kong.
- [7] W. Hess, “*Pitch determination of speech signals*”, Springer Verlag, Berlin, 1983.
- [8] J. Pelecanos and S. Sridharan, “Feature warping for robust speaker verification”, Proc. Speaker Odyssey 2001 workshop, June 2001.
- [9] J. Navratil, U. V. Chaudhari, and G. N. Ramaswamy, “Speaker Verification using Target and Background Dependent Linear Transforms and Multi-System Fusion,” *Eurospeech 2001*, Aalborg, Denmark.
- [10] G. N. Ramaswamy, J. Navratil, U. V. Chaudhari, and R. D. Zilca, “The IBM System for the NIST-2002 Cellular Speaker Verification Evaluation,” *ICASSP 2003*, Hong Kong.
- [11] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score Normalization for Text-Independent Speaker Verification Systems,” *Digital Signal Processing*, Vol. 10, No. 1, pp. 42-54, 2000.
- [12] R. D. Zilca, U. V. Chaudhari, and G. N. Ramaswamy, “The Sphericity Measure for Cellular Speaker Verification,” *ICASSP 2002*, Orlando, FL.