Pseudo Pitch Synchronous Analysis of Speech With Applications to Speaker Recognition; A Review

Chidiogo A. Ike-Egbuonu

**Abstract**

**Mel Frequency Cepstral Coefficients (MFCC) are used in speaker and speech recognition systems. Unfortunately, variations in pitch cause variations in these features and if not handled correctly, will lead to high errors in the recognizers. Also, the parsing of the raw, input speech signal into frames with a constant frame size does not encourage alignment with natural pitch cycles and using multiple frame sizes is not computationally ideal. This leads to the addition of ‘extra’ samples or artifacts in the power spectrum. This in turn affects the results of the Mel cepstral computation. R.D Zilca, B. Kingsbury, J. Navratil and G.N. Ramaswamy introduce three *Pseudo Pitch Synchronous* (PPS) signal processing procedures that attempt to address the above problems. Their objective is to keep a constant frame size and still align the frames to their natural pitch cycles while not truncating the pitch cycles in the process. To test these three procedures (depitching, syncpitching and padpitching), experiments were performed on NIST (National Institute of Standards and Technology) speaker recognition tasks. The results of the experiments show a better performance of the recognizers when the PPS procedures are fused with traditional procedures than with either procedure individually. There was also improvement in the results of speech recognition experiments for extremely low signal-to-noise ratio data.**

1. **Introduction**

Speaker and speech recognition technologies today have various applications in different fields. In the security filed, speaker recognition provides access to secure systems. Speech recognition can be employed by the military in battle management and training of air traffic controllers. As such, it is beneficial to have these technologies working as accurately as possible.

Speaker and speech recognition systems (recognizers) typically use the Mel Frequency Cepstral Coefficients (MFCC) from consecutive frames of the speech signals along with their time derivatives and a statistical classifier [1]. The process of obtaining the MFCCs is explained in [1] and when calculated, the MFCCs usually include some harmonic structure of the fundamental frequency which can increase the error rate of the recognizers. This is very noticeable when the speakers are high-pitched (typically female) and when the speaker’s pitch varies between enrollment (recording and creation of voice model) and testing (or verification: comparing sample speech to voice model). One way to improve the accuracy of recognizers as it concerns the variability in pitch is to train the system with a large and diverse collection of speakers. While this may work, it is not a perfect solution as there is always the unexpected that cannot be trained for. Another solution is that instead of preparing the system to handle the errors, we can redesign the system to reduce or eliminate ‘pitch-induced feature variability’ [1] completely.

The authors of the paper have presented three Pseudo Pitch Synchronous (PPS) processing algorithms to do just that. The algorithms will generally ‘accept individual raw speech frames, perform an adjustment of the fundamental frequency value on the LPC (Linear Predictive Coding) residual and produce time domain pitch-adjusted frames. The adjustment may attempt to produce a power spectrum completely free of pitch harmonics and with no interfering pitch harmonics, the MFCCs will be more accurate and lead to a better recognizer. Another option is for the adjustment to better align the speech frame boundaries to the raw speech signal and thereby reduce spectral estimation artifacts. [1]

In this paper, I will first review general aspects of the PPS algorithms and then describe them individually presenting the rationale behind the algorithm and the results of experiments done with the algorithm. In the end, I will present the authors’ conclusion and suggested future work, some review notes/questions and my conclusion.

1. **PPS Algorithms: General**

Three PPS algorithms will be described: depitch, syncpitch and padpitch. The input of the algorithms is a time domain speech signal and the output is also a time domain speech signal which unfortunately, is not continuous since speech frames typically overlap. The general procedure of the algorithms is shown in Figure 1 and is performed independently for each frame.

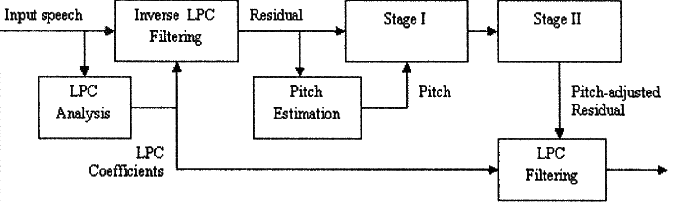


Figure 1: Block Diagram for the PPS Algorithms [1]

*Step 1:*

The input speech frame is windowed with a Hamming window. The Hamming window has its maximum amplitude as 1 and is non-zero between 0 and N-1 samples. It is used to minimize the height of the maximum side lobe. After this, the speech frame undergoes linear prediction analysis to derive the LPC coefficients. [1]

*Step 2:*

Using the LPC coefficients, the residual signal is calculated by filtering the output of Step 1 above using the inverse LPC filter. [1] The LPC algorithm finds an all-pole filter to fit the spectrum of the input signal frame and it attempts to minimize the error between the spectrum of the input and the frequency response of the filter. The inverse LPC filter extracts the LPC residual signal which represents the glottal in the voice tract. [2].

*Step 3:*

Pitch estimation (pitch and voice detection) is performed on the residual signal using a variant of the autocorrelation method [3]. Only frames of the residual signal classified as ‘voiced’ undergo this process. It is done in two major steps as shown in Figure 2 [1]:

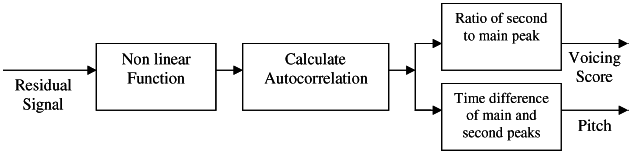


Figure 2: Block Diagram for Pitch Estimation [1]

First, a nonlinear function is applied to the residual signal performing center clipping and compression as shown in Figure 3. The threshold, TH (or clipping level) of the clipping function is calculated for each frame using (2) below and is based on the full scale as determined by (1). The center clipping function basically eliminates (or zeros-out) the center/middle amplitudes of a signal leaving only the maximum and minimum points which carry most of the information in the signal. The nonlinear function (NLF) is given by (3). [1]

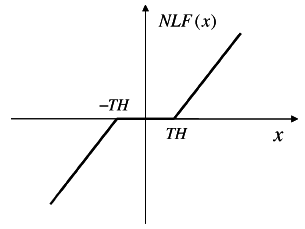


Figure 3: Clipping Function Used in Pitch Estimation [1]

|  |  |
| --- | --- |
| *FS = max {|H|,|L|}* (1)  *TH = α . FS* (2)  *NLF(x) =* (3)  FS = Full scale of frame.  H = maximum sample value in residual frame.  L = minimum sample value in residual frame.  α = constant fraction, chosen to be 0.07 in this case.  Secondly, the sample autocorrelation function is computed. Autocorrelation is the cross-correlation of a signal with itself and is used to find repeating patterns like the main periodic signal in a mixture of signal and noise. Using the method, one would take the input series, split it half and step through the computation each time shifting the second half by 1 sample. The resulting autocorrelation function has the general shape as shown in Figure 4. It has a main peak which signifies the residual signal energy and additional peaks (local maxima) corresponding to other modes of periodicity. [1].    Figure 4: General Shape of Autocorrelation Function in Pitch Detector [1]  (Autocorrelation Function Value vs. Number of Samples)  The outputs of the pitch and voice detector are the voicing score and the pitch. The voicing score is the ratio between autocorrelation values at the secondary peak and at the main peak. The pitch lag is the difference in number of samples between the main peak and the second largest peak. The more random the residual signal, the more the autocorrelation function resembles an impulse function and the closer the voicing score is to 0 (0/1 = 0). The more periodic the residual signal, the more it will resemble a cosine function and the closer the voicing score is to 1 (1/1 = 1). [1]  *Step 4:*  This is labeled as ‘Stage I’ in Figure 1 and it varies for the different PPS algorithms. It has to do with determining the pitch cycles in samples. | *Step 5:*  This is labeled as ‘Stage II’ in Figure 1 and it varies for the different PPS algorithms. It involves interpolating or zero padding and a cyclic shift for the pitch cycles. Interpolation involves low pass filtering of the residual signal and this could lead to loss of parts of the signal.   1. **Depitch PPS Algorithm** 2. Stage I and Stage II [4]   Stage I of the depitch algorithm is to “extract a single pitch cycle consisting of *p* samples from the center of the frame, where *p* is the estimated pitch cycle in samples.”  Stage II of the depitch algorithm is to “interpolate the *p* samples to fit the size of one frame, *N* (up-sample from *p* to *N*).” To up-sample, add *N-1* zeros between each sample in the residual signal and then filter with a sinc, low-pass filter with cut-off frequency at . This stage leads to some loss because some samples are ignored if they are not in the center of the frame. However, there is overlap between the frames so in the end; a minimal number of samples are ignored. Interpolating involves a low pass filter and this leads to some spectral distortion in the output speech spectrum.   1. Rationale and Motivation   The first objective for using the depitch algorithm in speaker recognition is to improve the accuracy for the subset of ‘high-pitched speakers and pitch mismatched trials that normally perform significantly worse than average’ as mentioned in the introduction. [1] When successful, depitching will lead to a better distribution of individual errors in the subset while not compromising the overall average accuracy. Secondly, this method provides insight into the role of pitch information in auto-speaker recognition. [1]  In speech recognizers, depitching will force all voiced frames to have the same pitch frequency and thus greater reduce issues with variable pitch.   1. Effect of Depitching on Speech Power Spectrum [1]   After depitching, the resulting power spectrum is smoother than the original spectrum (Figure 5). This is because the effects of the fundamental frequency fluctuations have been removed. Therefore, “when using depitching as a front-end processing stage in speaker recognition, we [can] expect an improvement in the false rejection rate”. This means that speakers will still be recognized if their pitch changes between enrollment and testing as the feature vectors will not be changed. This is good however; it also means that more imposters are likely to be accepted as the difference in their pitch will no longer raise a flag as it would have before depitching.   1. Speaker Verification Experiments   The experiments were conducted using the NIST 2002 speaker recognition evaluation corpus [4]. The subset used includes 191 female speakers, 23309 trials using 2128 test sentences. This is because most of the pitch-related problems occurred with high-pitched (mostly female) |
| Figure 5: Effect of Depitching on The Power Spectrum [1]  (a) Original (b) Depitched | |
| speakers. The two systems used for the experiment were the baseline system or ‘nondep’ and the depitch system.  To evaluate the systems’ performances, we use the EER (equal error rate) and DCF (decimal cost function). The lower the EER and DCF, the better the system. Several experiments were run in an attempt to get the best data set. The results below are from data sets on which the TNORM (triangular norm) has been run.   |  |  |  | | --- | --- | --- | | Procedure | DCF | EER (%) | | Nondep | 39.3 | 10.2 | | Depitch | 54.9 | 16.0 | | Fusion of nondep and depitch | 40.1 | 10.2 |   Table 1: Speaker Recognition Experiment Results for Depitch PPS [1]  The results show that there is no real performance improvement when the traditional and depitch methods are combined.   1. **Syncpitch PPS Algorithm** 2. Stage I and Stage II [5]   Stage I of the syncpitch algorithm is to “calculate the maximal number of pitch cycles, *nmax,* that would not exceed the frame size *N*. Then, extract *nmax* x *p* samples of the residual signal, where *p* is the estimated pitch cycle in samples.”  Stage II of the syncpitch algorithm is to “interpolate the *nmax* samples to fit the size of one frame (up-sample from *nmax* to *N*). Perform a cyclic shift such that the frames edges have the lowest energy.” This stage leads to some loss because some samples are ignored in each frame but the overlapping of frames helps minimize loss. There is still some spectral distortion with the use of the low pass filter | however, not as much distortion as in the depitch algorithm because *nmax* is closer to N than *p* is. The cyclic shift should not typically have any effect on the features because it is supposed to provide continuity and change the phase spectrum only. Due to windowing however, the shift has a small effect on the features.   1. Rationale and Motivation   Syncpitch is a milder version of depitch in terms of filtering. Depitch forces a single pitch cycle in every frame (*p*) while syncpitch up-samples to the closest possible integer number of pitch cycles. Syncpitch addresses mainly the alignment of the pitch cycles and maintaining most of the original frequency information. Its cutoff frequency is . [1]   1. Effect of Syncpitching on Speech Power Spectrum [1]   After syncpitching, the resulting power spectrum has the same spectral structure as the original spectrum. The syncpitch method as mentioned before, is a mild version of the depitch method and is thus, not expected to have much spectral distortion and this is shown in Figure 6.   1. Speaker Verification Experiments   The experiments were conducted using the NIST 2002 speaker recognition evaluation corpus [4]. The subset used 330 target speakers and 39105 test trials (male and female). All the systems used TNORM normalization and |
| Figure 6: Effect of Syncpitching on The Power Spectrum [1]  (a) Original (b) Depitched  a set of 234 imposter models. Two baseline systems were used with corresponding syncpitch systems and the results are given below. Baseline I was trained on NIST 2001 cellular development set and NUST 1996 evaluation set run through GSM coders. [7] [8]. Baseline II was trained on NIST 2001 cellular development set and NIST 1990 landline evaluation set run though CDMA coder.   |  |  |  | | --- | --- | --- | | Procedure | DCF | EER (%) | | Baseline II | 32.7 | 8.8 | | Syncpitch I | 36.7 | 9.6 | | Fusion of systems | 31.4 | 8.6 |   Table 2a: Speaker Recognition Experiment Results for Syncpitch PPS with Baseline I System [1]   |  |  |  | | --- | --- | --- | | Procedure | DCF | EER (%) | | Baseline II | 35.24 | 9.2 | | Syncpitch II | 36.3 | 9.0 | | Fusion of systems | 33.2 | 8.3 |   Table 2b: Speaker Recognition Experiment Results for Syncpitch PPS with Baseline II System [1]  The fact that the baseline systems perform better than the syncpitch systems may be because of the loss of high frequency energy in Stage II’s interpolation. Overall though, the combination of the two systems results in improved performance.   1. **Padpitch PPS Algorithm** 2. Stage I and Stage II [1]   Stage I of the syncpitch algorithm is to “calculate the maximal number of pitch cycles, *nmax,* that would not exceed the frame size *N*. Then, extract *nmax* x *p* samples of the residual signal, where *p* is the estimated pitch cycle in samples.”  Stage II of the syncpitch algorithm is to “set the remaining (*N - nmax* x *p*) samples to zero and perform a cyclic shift such that the frame edges have the lowest energy (zero-padding).” | The padpitch algorithm uses zero-padding instead of interpolating to eliminate the loss from the low pass filter.  The cyclic shift should not typically have any effect on the features because it is supposed to provide continuity and change the phase spectrum only. Due to windowing however, the shift has a small effect on the features.   1. Rationale and Motivation   Padpitch is an attempt to eliminate the need for a low pass filter due to interpolating by zero padding instead. Samples are zeroed out instead of being completely ignored as in the other PPS algorithms.   1. Effect of Padpitching on Speech Power Spectrum [1]   Padpitching eliminates the interpolation process and just performs zero padding. This means there is no loss from the filter and as can be seen in Figure 7, the processed spectrum is closer to the original than in the other cases. The difference between the two spectral images is due to the reduction in the artifacts that normally result from the difference in the natural pitch cycle and the frame size.   1. Speaker Verification Experiments   The experiments were conducted using the NIST 2002 speaker recognition evaluation corpus [4]. For the first trial, no TNORM score normalization was used. The results were not good:   |  |  |  | | --- | --- | --- | | Procedure | DCF | EER (%) | | Nondep | 42.6 | 10.7 | | Padpitch | 57.5 | 13.3 | | Fusion of systems | - | - |   Table 3: Speaker Recognition Experiment Results for Padpitch PPS[1]  As the table shows, the difference in performance between the 2 systems is very significant. As a result, further testing was not carried out.   1. **Conclusion**   The paper introduced three PPS procedures for pitch adjustment. The goal was to adjust the pitch without changing the frame size by either removing pitch information (depitch), interpolating pitch cycles (syncpitch) or zero padding incomplete pitch cycles (padpitching).  It was discovered that the depitch and syncpitch methods, when compared with traditional methods, showed some improvement. The depitch method showed more improvement than the syncpitch method (over 20% as compared to 10%).  Intuitively, I would have expected the padpitch method to be the best as it has little energy loss and its resulting power spectrum closely matches the original spectrum. This was not the case however and no explanation was given for the performance of the padpitch method. |
| Figure 7: Effect of Padpitching on The Power Spectrum [1]  (a) Original (b) Depitched | |
| 1. **REVIEW NOTES**   The paper discusses the use of the PPS algorithms for both speaker and speech recognition but I have focused on the speaker recognition aspects. This is because the authors concluded, from experimentation, that the PPS procedures were more useful in speaker recognizers than in speech recognizers.  In Step 3 of the PPS algorithms, a pitch estimator/detector is used however; the accuracy of the pitch detector was not formally assessed. It is expected that some doubling and halving will occur with the autocorrelation function but it is expected that the PPS algorithms will not be very sensitive to these errors since the frames overlap. Also, the main goal is to make integer pitch cycles available for analysis and this goal is not compromised.  One thing to note is that all the experiments were not performed on the same data set in order to compare the performance of the PPS algorithms to one another. It would have been insightful to know how the algorithms perform on a subset of all female speakers and on a subset of male and female speakers. The syncpitch method was used on a mixed subset but since it is less ‘lossy’ than the depitch method, I expect it would perform better than the depitch method on a subset of only female speakers. | Due to the large degradation in DCF and EER, no further tests were performed using the padpitch method. This was not expected as the padpitch method should have performed better than the depitch and syncpitch methods since it did not have any interpolation and filter loss to deal with. The authors did not explain the reason for this but I believe it could have to do with the pitch mismatch in the residual signal for which the sample amplitude was zero (after zero padding). The authors suggested frame specific testing for future work. In the case of testing the algorithms on only a subset of frames with large errors, it would be interesting to note how the procedures perform.  From the experiments already on record, it can be assumed that the syncpitching method will be most effective depending on the data set used. |

**REFERENCES**

[1] R.D Zilca, B. Kingsbury, J. Navratil and G.N. Ramaswamy, “Pseudo Pitch Synchronous Analysis of Speech With Applications to Speaker Recognition”, *IEEE Transactions on*

*Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 467-478, March, 2006.

[2] K. I. Nordstrom, P. F. Driessen, and G. A. Rutledge, "Influence of the LPC filter upon the perception of breathiness and vocal effort", *IEEE Int.Symposium on Signal Processing and Information Technology (ISSPIT06)*, Vancouver, BC, Canada, August 2006.

[3] W. Hess, *Determination of Speech Signals*. Berlin, Germany: Springler Verlag, 1983

[4] R.D Zilca, B. Kingsbury, J. Navratil and G.N. Ramaswamy, “Depitch And The Role of Fundamental Frequency in Speaker Recognition,” in *Proc. ICASSP*, 2003.

[5] R.D Zilca, B. Kingsbury, J. Navratil and G.N. Ramaswamy, “Syncpitching: A Pseudo Pitch Synchronous Algorithm for Speaker Recognition,” in *Proc. Eurospeech*, 2003.

[6] [Online]. Available: http://www.nist.gov/speech/tests/spk/index.htm

[7] G.N. Ramaswamy, J. Navratil, U.V. Chaudhari and R.D. Zilca, “The IBM System for the NIST-2002 Cellular Speaker Verification Evaluation,” in Proc. ICASSP, 2003.

[8] G.N. Ramaswamy, U.V. Chaudhari and R.D. Zilca, “The Sphericity Measure for Cellular Speaker verification’” in Proc. ICASSP, 2002.