Glottal Mixture Model (GLOMM) for Speaker Identification on Telephone Channels

Paul M. Baggenstoss and Kevin Wilkinghoff and Frank Kurth
Fraunhofer FKIE, Fraunhoferstrasse 20
53343 Wachtberg, Germany
Email: p.m.baggenstoss@ieee.org, kevin.wilkinghoff@fkie.fraunhofer.de, frank.kurth@fkie.fraunhofer.de

***Abstract*—The Glottal Mixture Model (GLOMM) extracts speaker-dependent voice source information from speech data. It has previously been shown to provide speaker identification performance on clean speech comparable to universal background model (UBM), a state of the art method based on MFCC. And, when combined with UBM, the error rate was reduced by a factor of three, showing that the voice source information is largely independent of the information contained in the MFCC, yet holds as much speaker-related information. We now describe how GLOMM can be adapted for telephone quality audio and provide significant error reduction when combined with UBM and I-vector approaches. We demonstrate a factor of two error reduction on the NTIMIT data set with respect to the best published results.**

I. Introduction and Previous Work

Recent work in speaker identification relies on front-end processing that extracts short-time spectral information, usually using the MEL frequency cepstral coefficient (MFCC) features [1]. MFCC is the most prevalent short-term spectral feature despite the fact that it was developed for speaker-independent speech recognition. It is logical, then, that much speaker-related information is missing from MFCC. Attempts to add voice source information back into speaker identification systems to improve them have met limited success [1], [2], [3], [4], probably due to the difficulty and reliability of estimating the voice source waveform itself. We previously introduced the GLOMM method [5], which was based on detecting glottal events (glottal opening/closing) by detecting times of high linear prediction error. Rather than attempt to reconstruct the glottal source waveform accurately by separately modeling open and closed glottis regions [6], [7], GLOMM simply models the data as a recurrent pulse shape (located at the glottal event times) driving a (slowly) time-varying all-pole filter. The filter (linear prediction coefficients) and the recurrent pulse shape are estimated in alternation. In this paper, we review GLOMM and describe the algorithm changes that were required for the method to work on telephone audio.

II. Review of GLOMM

The most basic principle in voice source estimation is linear prediction [3]. The linear predictive coding (LPC) coefficients are readily estimated using classical methods. In addition to providing a good approximation to the vocal tract filter (VTF), the prediction error waveform is a first-order approximation to the voice source waveform (VSW), which approximates the derivative of the glottal flow [2]. Refer to Figure 1, which is a block-diagram of GLOMM. Parameters are shown in slanted paralellograms and processing is shown as boxes. There are two sections, a frame-based section, and a time-series based section. Any parameter or processing shown in the frame-based part is repeated individually for each frame. Conversion from frame-based to time-series is done using overlap-add.

We now describe the algorithm and note where it differs from the original GLOMM algorithm for clean speech [5]. Refer to Figure 1. The input data is first segmented into overlapped Hanning-weighted frames suitable for lossless re-combination with overlap-add. We then obtain an initial estimate of LPC coefficients in each frame using classical methods (autocorrelation/Levinson [8]). Using these LPC coefficients, the data in each frame is whitened in the frequency domain by multiplying the DFT of the data by the DFT of the LPC coefficients. Then, the negative frequency bins (those between N/2 and N) are set to zero before the inverse DFT is computed. This results in frames of complex (analytic) whitened time-series and is represented as "Hilbert transform" in the block diagram. The frames are then re-combined into a (complex) time-series using overlap-add to obtain the complex analytic linear prediction error time-series (LPETS), which brings us into the time-series-based part of the block diagram. We then take the absolute value of the complex LPETS, resulting in a real positive-valued envelope. Potential glottal events are found by peak-picking the LPETS envelope. Processing up to this point is the same as GLOMM for clean speech (See Section II.B of [5]).

Because glottal peaks in degraded audio are more difficult to locate, we implemented peak tracking. The purpose of peak tracking is to remove detections that are not consistent with the measured autocorrelation function. This is a new component of GLOMM and is explained in detail in Section III-B. Next, we collect glottal windows by extracting a window of LPETS centered at each glottal event (See Section III-C). Unlike GLOMM for clean speech which classified speakers by attempting to re-synthesize speech using the speaker's glottal parameters, the glottal windows themselves are the effective output of GLOMM for degraded speech. They are used much like MFCC features are used in speaker identification. But, since in the first pass, the glottal windows are only based on the initial LPC coefficients, which are biased, the GLOMM algorithm must iterate a few times before the glottal windows are suitable to be used for speaker identification. To account for phase distortions in degraded audio, we added an additional step of phase correction which is detailed in Section III-C.

The process of re-estimating the LPC coefficients begins with distilling the glottal windows into a set of parameters that

ISBN 978-0-9928626-7-1

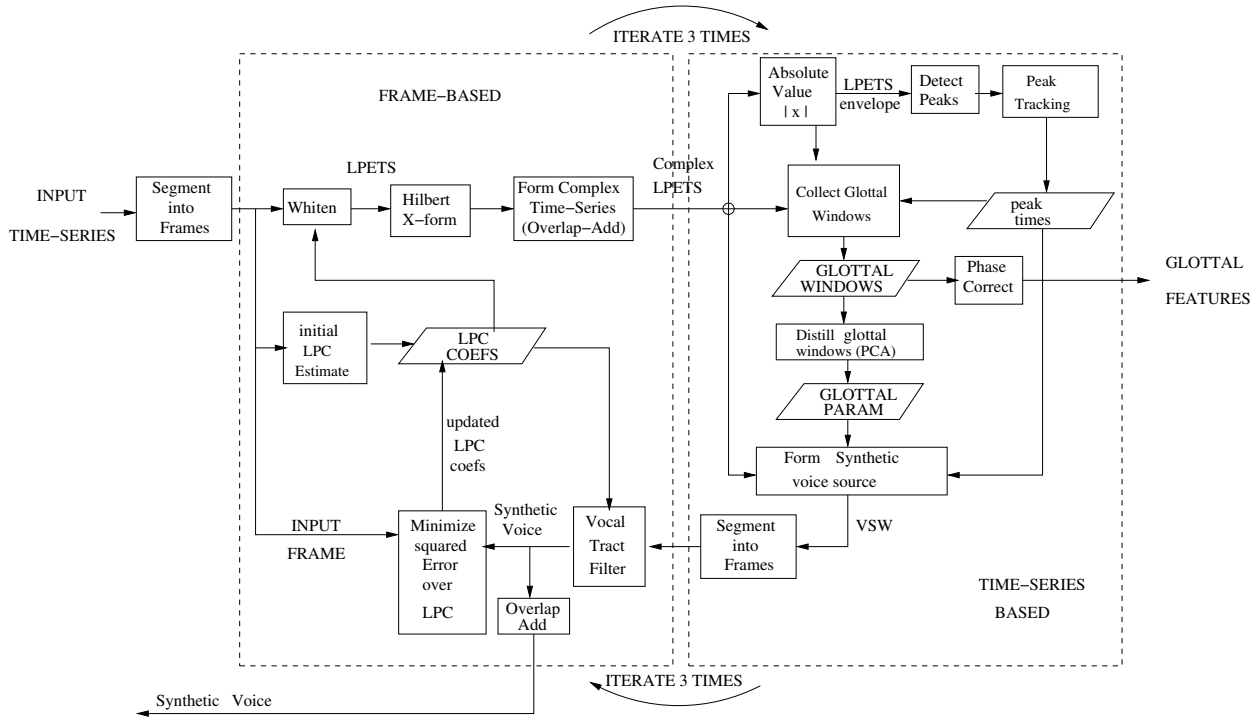

Fig. 1. GLOMM block diagram.

can be used to create a synthetic voice source waveform (See Section III-D). In the next step, we form a synthetic voice source by inserting a synthetic glottal pulse at the location of each detected glottal event. The synthetic voice source time-series is then segmented again into frames and then processed by the vocal tract filter, which is the all-pole filter formed from the LPC coefficients. The result is a synthetic approximation to the input time-series. The updated LPC coefficients in each frame are obtained by minimizing the modeling error between the synthetic voice source time-series and the input data. The steps of creating the synthetic voice source and optimizing the LPC coefficients is explained in Section III-E. Once the new LPC coefficients are obtained, the algorithm repeats with the "Whiten" block. This is repeated three times.

III. GLOMM FOR DEGRADED AUDIO

A. Effects of degraded audio

In telephone and other voice communications channels, the reliability of glottal pulse detection is reduced due to amplitide distortion, phase reversals, and high-pass filtering (loss of f_0 pitch fundamental). Despite these effects, the pitch frequency can still be be deduced from the spectral lines separated by f_0, and seen as peaks in the auto-correlation function (ACF) separated by $1/f_0$. Therefore, the glottal pulses, although weaker and not as sharp, are still present, and are repeating at the $1/f_0$ period. Our approach is to detect peaks with a lower threshold, and then reject those detections that are not consistent with the ACF. This idea is implemented by glottal peak tracking.

B. Glottal Peak Tracking by ACF validation

The purpose of peak tracking is to eliminate detections that are inconsistent with the measured auto-correlation function (ACF), using a method developed for the detection of repeating clicks from marine mammals [9]. To demonstrate peak tracking, we used a recording of degraded audio from the RATS corpus [10], "H" channel, which is a radio channel exhibiting amplitude compression. Refer to Figure 2. In Figure 2-(1), we see the LPETS envelope, with red circles drawn at the detected peaks. Figure 2-(3) is the normalized ACF for three frames spanning the illustrated time segment. A clear peak can be seen at 6 ms and a weaker ACF peak at 3.5 ms due to the interaction between the glottal closure and glottal opening. In Figure 2-(1), there are a total of 12 detected "glottal" peaks. One of them (number 10) is invalid. We have indicated with letter "A" the location of the "correct" peak, which is slightly after the invalid detection, but is more consistent with the surrounding periods. The valid detection was not detected by the peak-picking allgorithm, so cannot be recovered, but the false detection can be removed. In Figure 2-(2), there is an intensity plot of the 12×12 ACF pairing matrix $A_{i,j} = r(\tau_{i,j})/r(0)$, where $r(\tau_{i,j})$ is the ACF at the lag time equal to the time spacing between the detections. It is used to approximate the probability that detection "i" can be paired with detection "j". For example, notice that a bright value is indicated for pairing detection 4 with detection 6. This value is determined by looking at the difference in time between detections 4 and 6, then going to the ACF plot and choosing that ACF value. The difference in time between detections 4 and 6 is 6 ms, which corresponds to an ACF value of 0.5. This is also true of detections 5 and 7, and detections 6 and 8, etc.

We use an optimal assignment algorithm to find the

ISBN 978-0-9928626-7-1 © EURASIP 2017

globally-best pairings. To do this, we define an assignment "error" for pairing detections i and j [9], which is set to $E_{i,j} = -\log\left(r(\tau_{i,j})/r(0)\right) = -\log(A_{i,j})$. The algorithm [9] finds the unique set of pairings that globally minimize the total error. In Figure 2-(4), we see the automatically-generated pairings (dotted lines). Here you see one sequence created from detections 1,3,5,7,9,11, and another from detections 4,6,8,12 - one set is glottal openings, and the other is glottal closings. Note that the invalid detection (number 10) has been skipped. Instead, detections 8 and 12 have been connected because their time separation coincides with an ACF lag of 12 ms, which has a high ACF value. All detection that are not paired will be dropped.

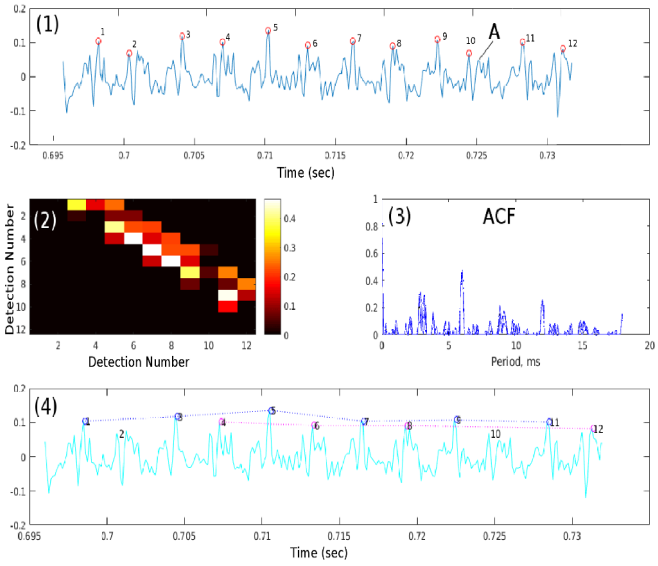

Fig. 2. Illustration of click tracking using a section of data from RATS "H" channel. (1) LPETS envelope with detections, (2) normalized ACF values for each detection pair, (3) normalized ACF, (4) LPETS envelope with associated detection groups.

C. Glottal Window Collection and Phase Correction

Glottal windows are time windows of LPETS centered on each detected glottal pulse. All glottal windows are vernier time-corrected so that the waveform has an amplitude peak precisely at the $Q+1$-th sample, where $2Q+1$ is the total window length (Section II.C of [5]).

For degraded audio, we added the step of phase correction. To phase correct, we divided each complex glottal window by its value at the $Q+1$-th sample, so it will always have precicely value 1 there, with imaginary part zero. In Figure 3, we show glottal windows from speaker "fadg0" (real part) without phase correction. Each glottal pulse is shown as an intensity-modulated column in the vertical direction. Note that here the glottal window half-size is $Q = 52$ and the glottal pulses all have a (amplitude) peak at sample 53. There appears to be a wide variation in the glottal pulse shape. However, on closer inspection, we see this is due to phase reversals that are common in the older analog two-wire telephone circuits used to create NTIMIT. For the clean TIMIT corpus, such phase variations were not noticed. In Figure 4, we see the corresponding phase-corrected glottal windows (real part). The glottal windows look all of a sudden much more consistent. And, when comparing with Figure 5, which is from speaker "falk0", we see that the two speakers have distinct characteristics. It is logical, then, that the glottal windows themselves can serve as a feature, similar to MFCC.

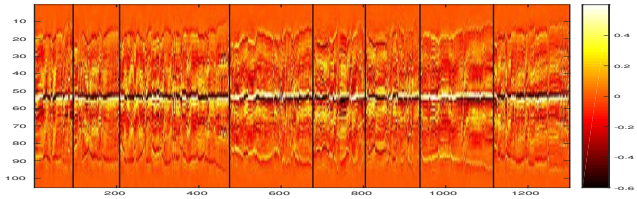

Fig. 3. Glottal windows from speaker "fadg0" - no phase correction.

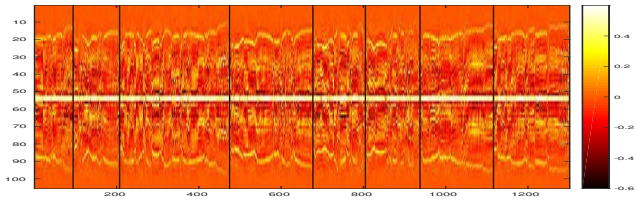

Fig. 4. Glottal windows from speaker "fadg0" - with phase correction.

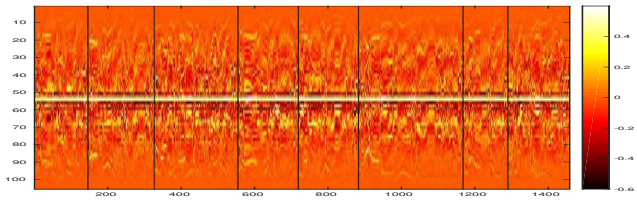

Fig. 5. Glottal windows from speaker "falk0" - with phase correction.

D. Distilling Glottal Windows

By the word "distilling", we mean the process of extracting a set of parameters that describe the collected glottal windows. In the original GLOMM algorithm, distilling included PCA analysis and GMM clustering (See Section II.C of [5]). We have since dropped the GMM clustering step in favor of just doing PCA. Suppose there are K detected glottal pulses. Let $\mathbf{Z}$ be the $(2Q+1)\times K$ matrix of complex glottal windows (real part of $\mathbf{Z}$ is shown in Figures 4 and 5). Let matrix $\mathbf{U}$ be the $(2Q+1)\times P_s$ matrix formed from the largest P_s singular vectors of $\mathbf{Z}$, where $P_s = 2$. Matrix $\mathbf{U}$ constitutes the "distilled" glottal parameters.

E. Synthetic Voice Source and LPC optimization

To create the synthetic voice source waveform (VSW), we place a "synthetic" pulse at the time of each detected glottal event, with amplitude consistent with the LPETS. Therefore, the synthetic VSW approximates the LPETS using predetermined pulse shapes that are obtained from distilled glottal windows. GLOMM for clean speech [5] constructed the VSW from the cluster centers obtained from glottal pulse clustering.

ISBN 978-0-9928626-7-1 © EURASIP 2017

For degraded audio, we have removed glottal pulse clustering and have replaced it with the simpler PCA. Let $\mathbf{z}_k$ be a time-window from LPETS centered at detected glottal pulse k (i.e. a single column of $\mathbf{Z}$). Then, $\mathbf{s}_k = \mathbf{U}(\mathbf{U}'\mathbf{z}_k)$ is $\mathbf{z}_k$ projected onto $\mathbf{U}$ and constitutes the "synthetic" replacement for $\mathbf{z}_k$. The entire VSW time-series is constructed by the superposition of $\mathbf{s}_1 \ldots \mathbf{s}_K$, each time-shifted to the corresponding detection time. The completed VSW is then segmented into frames.

In the final step, the LPC coefficients in each frame are individually optimized so that the VSW passed through the vocal tract filter matches the corresponding input data frame as close as possible (model-based measure of fit). Further details are given in Section II.E of [5].

F. Classifier Approach

The original GLOMM classification method is described in Section II.G of [5] and used the model-based measure of fit mentioned in the previous section as a classification statistic. In GLOMM for degraded audio, we do not use the GLOMM algorithm as a classifier. Rather, we use the extracted phase-corrected glottal windows (real part) as features, much like MFCC is used. We then apply can apply speaker/channel separation methods and dimension reduction methods including UBM [11], JFA [12], and I-Vector [13].

IV. Experiments

We conducted experiments using classifiers based on three feature extraction methods: GLOMM, MFCC, and PITCH, which are described in the next sections. We then combined the outputs of the three classifiers linearly.

A. GLOMM feature extractor and classifier

The GLOMM algorithm is described in [5] and its modifications for degraded audio are described above. We used the following parameter settings (See [5] for variable definitions):

1) LPC model order $P = 12$.
2) FFT size (at 8 kHz sample rate) $N_{FFT} = 336$.
3) Glottal data window length = 13 milliseconds, which for 8000 Hz sample rate results in a glottal window half-size of $Q = 52$.
4) Glottal window PCA dimension 2.
5) Null assignment cost for Jonker-Volgenant algorithm $p = 2.4$ (See eq. (1) and Section III.B in [9]).

To classify using the GLOMM features, we trained a 28-component UBM on the glottal window data from all speakers (training files). We then enrolled the UBM using data from each speaker.

B. MFCC classifier

We extracted 19-dimensional MFCC with the HTK toolkit [14] using a 25 millisecond window with 10 millisecond frame rate. The HTK configuration parameters are TARGETRATE = 100000, WINDOWSIZE = 250000, PREEMCOEF = 0.96, CEPLIFTER = 22, NUMCHANS = 20, NUMCEPS = 19, DELTAWINDOW = 3, ENORMALISE = F, SOURCERATE=1250, SAVECOMPRESSED = T, SAVEWITHCRC = F, USEHAMMING = T, LOFREQ = 300, HIFREQ = 3400, TARGETKIND = MFCC . Using these MFCC features, we tried three MFCC classifiers: UBM [11], JFA [12], and I-Vector [13]. The parameters of the classifiers were individually optimized. The UBM classifier used a 192-component mixture. The JFA had an eigenvoice dimension of 75 and an eigenchannel dimension of 14. The I-vector classifier used an I-vector dimension of 225, with a 68-dimensional LDA space.

C. PITCH classifier

To estimate pitch, we divided the data into overlapped Hanning-weighted frames, then extracted pitch information from each frame by peak-picking the autocorrelation function (ACF) for the highest peak in the range of human pitch. From each frame, we measured pitch period τ and amplitude $a = r_\tau / r_0$, which was less than 1. To train on a given speaker, we fit a 1-dimensional Gaussian mixture (essentially a smoothed histogram) to the values τ, using a as a sample weight. Since each speaker presumably had a different distribution of pitch estimates, we were not concerned with errors associated with choosing the ACF peak at $2f_0$ or $f_0/2$. To classify using pitch, we evaluated the GMM on the pitch data extracted from the training utterance, adding up the log-likelihood weighted by the sample weights a.

D. Combined Classifier

We combined the classifiers using $L = L_{\mathrm{MFCC}} + \beta_1 L_{\mathrm{GLOMM}} + \beta_2 L_{\mathrm{PITCH}}$, and measured classification error as a function of β_1 and β_2.

E. NTIMIT Data

The NTIMIT speech recognition corpus is a re-processing of the TIMIT corpus through telephone circuits. It consists of 630 male and female speakers, each having 10 utterances, averaging about 3 seconds each, and divided into eight "SX" and "SI" utterances and two "SA" utterances. In the speaker identification experiments, we trained on all eight "SX" and "SI" utterances and tested on the "SA" utterances. We used no voice activity detection, always using the complete utterances. All data was down-sampled to 8 kHz, since in NTIMIT there is no activity above 4 kHz.

We conducted a 10-speaker and a 630-speaker experiment. In the 10-speaker experiment, we used 500 independent trials. In each trial, we chose 10 speakers at random from the 630 available. Then we formed a 10×10 classification experiment, testing each of the 2 test utterances from each speaker. There were therefore 20 individual classification decisions per trial, or 10,000 individual classification decisions total. In the 630-speaker experiment, there was one trial with 2 test utterances per speaker, so a total of $2 \times 630 = 1260$ decisions.

F. Results

For 10-speakers, the individual classifiers attained the following classification error in percent): MFCC/UBM 5.73%, MFCC/JFA 5.36%, MFCC/I-vector 4.81%, PITCH 30.82%, GLOMM/UBM 16.51%. The combined classifier (MFCC/I-vector + PITCH + GLOMM) attained 2.2% (See Figure 6, left). Just MFCC/I-vector + PITCH got 2.97%, and MFCC/I-vector + GLOMM attained 2.61%. For 630 speakers, the

ISBN 978-0-9928626-7-1 © EURASIP 2017

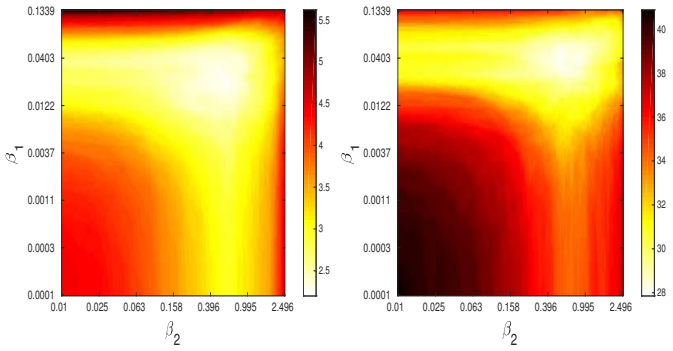

Fig. 6. Combination results GLOMM + PITCH + UBM. Left: Ten speakers. Right: 630 speakers.

combined classifier (MFCC/I-vector + PITCH + GLOMM) attained 27.85% error (See Figure 6, right). Just MFCC/I-vector + PITCH got 33.9%, and MFCC/I-vector + GLOMM attained 30.4%. We compare these results with existing results available in the literature [15], [16], [17], [18] in Figure 7. We can conclude that the MFCC-only results are comparable with the existing results, both at 10 and 630 speakers. The combined classifiers greatly out-perform the published results.

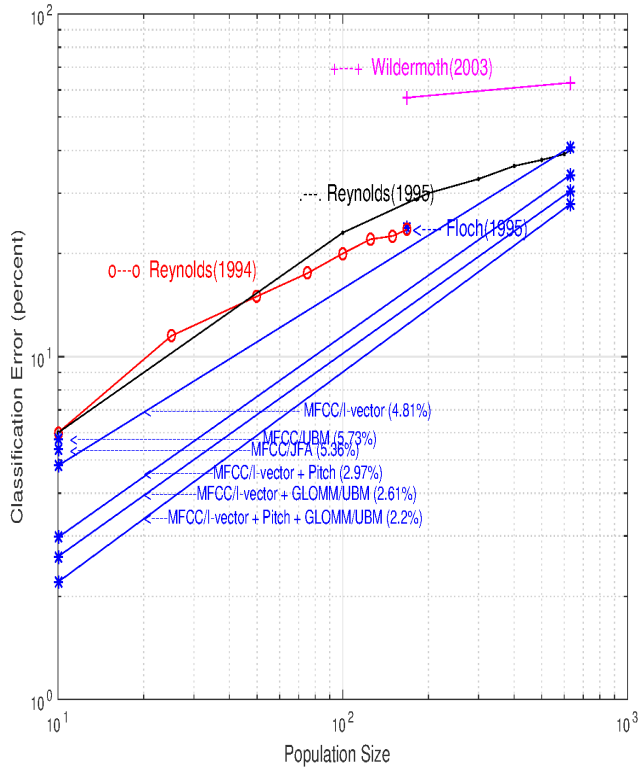

Fig. 7. Speaker identification results on NTIMIT: Comparison with published results

V. Conclusions

In this paper, we have made GLOMM robust against distortions present in telephone quality audio using glottal window phase correction and glottal pulse tracking. In doing so, we simplified the GLOMM classifier, using the extracted glottal windows much like MFCC features. We have shown that a hybrid classifier that linearly combines classifiers based on GLOMM, PITCH, and MFCC features attains 2.2% error for a 10-speaker population, at least a factor of two better than the best previously-published results. At 630 speakers, it attained 27.85% error, bettering the performance of 39% error reported by Reynolds.

References

[1] J. Gonzalez-Rodriguez, "Evaluating automatic speaker recognition systems: An overview of the nist speaker recognition evaluations (1996-2014)," *Loquens*, vol. 1, no. 1, 2014.

[2] M. Plumpe, T. Quatieri, and D. Reynolds, "Modeling of the glottal flow derivative waveform with application to speaker identification," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, pp. 569–586, Sep 1999.

[3] R. R. Rao, V. K. Prasad, and A. Nagesh, "Performance evaluation of statistical approaches for text- independent speaker recognition using source feature," *Computer Science and Networking*, vol. 2, pp. 8–13, Aug 2010.

[4] B. Wildermoth and K. K. Paliwal, "Use of voicing and pitch information for speaker recognition," in *Proc. 8th Australian International Conf. Speech Science and Technology, Canberra*, 2000.

[5] P. M. Baggenstoss, "Combining the glottal mixture model (GLOMM) with UBM for speaker recognition," in *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 2156–2160, Aug 2016.

[6] T. Ananthapadmanabha and B. Yegnanarayana, "Epoch extraction from linear prediction residual for identification of closed glottis interval," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 309–319, Aug 1979.

[7] J. Walker and P. Murphy, "Advanced methods for glottal wave extraction," in *Proceedings of International Conference on Nonlinear Analyses and Algorithms for Speech Processing*, Springer, Berlin, Heidelberg, 2005.

[8] S. Kay, *Modern Spectral Estimation: Theory and Applications*. Prentice Hall, 1988.

[9] P. M. Baggenstoss, "The jonker-volgenant algorithm applied to click-train separation," *J. Acoust. Soc. Am.*, no. 135, 2014.

[10] D. Graff, K. Walker, S. Strassel, X. Ma, K. Jones, and A. Sawyer, "The rats collection: Supporting hlt research with degraded audio data," *LREC*, pp. 1970–1977, 2014.

[11] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[12] P. Kenny, "Joint factor analysis of speaker and session variability : Theory and algorithms," *CRIM Technical report CRIM-06/08-13*, 2005.

[13] N. Dehak, P. Kenny, R. Dehak, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, May 2011.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book, Version 3.4*. Cambridge University Engineering Department, 2006.

[15] J.-L. L. Floch, C. Montacié, and M.-J. Caraty, "Speaker recognition experiments on the ntimit database," in *Eurospeech. 1995*, 1995.

[16] B. R. Wildermoth and K. K. Paliwal, "Gmm based speaker recognition on readily available databases," in *Microelectronic Engineering Research Conference, Brisbane, Australia. 2003*, 2003.

[17] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," in *ESCA Workshop on Automatic Speaker Recognition, Identification, and Verification, Martigny, Switzerland*, Apr 1994.

[18] D. A. Reynolds, "Large population speaker identification using clean and telephone speech," *IEEE Signal Processing Letters*, vol. 2, pp. 46–48, Mar 1995.

ISBN 978-0-9928626-7-1 © EURASIP 2017