A Software Kit for Automatic Voice Descrambling

K. Riedhammer, M. Ring and E. Nöth
Lehrstuhl für Informatik 5 (Mustererkennung)
Univ. Erlangen-Nürnberg, GERMANY
sikoried@cs.fau.de

D. Kolb
MEDAV GmbH
Gräfenberger Str. 32-34, 91080 Uttenreuth, GERMANY
http://www.medav.de

***Abstract*—Voice scrambling is widely used to add privacy to the radio communication of various authorities – but is also used by criminals to evade prosecution. In this article, we consider various analog voice scrambling techniques such as fixed frequency inversion, splitband inversion and rolling code scramblers. We explain how to break them using automatically extracted measures and scoring algorithms, and evaluate the proposed system using simulated data. While the simple inversion can be easily broken, the more advanced techniques require additional work prior to unsupervised automatization; the presented user interface allows the user to refine the automatic results to obtain a high quality solution.**

I. INTRODUCTION

Many authorities including police, fire department, tow trucks, military and alike, use voice scrambling to add privacy to their radio communications. Although voice scrambling does not provide security due to its simple implementation, the manual decipherment is a tedious task, thus one could think of voice scrambling to add *temporal* security – which may be sufficient for time-critical operations. In general, "a rule of thumb is 60:1 'grunt time to clear speech time'."[1] Unfortunately, voice scrambling is not only used by authorized personnel but also by villains taking part in organized crime such as drug dealing and man hunt, making it hard for authorities to succeed in surveillance and raids.

In this article, we take on the widespread analog voice scrambling, a symmetric and frequency based modulation of the speech signal. Our vision is an automatic descrambler that acts as a one-fits-all adapter to analog voice scrambling that allows to listen to the clean speech in real time. Beside the use to aid real time reconnaissance in ongoing operations, an automatic descrambler can be used in conjunction with large-scale surveillance assets like broad-band radio scanning and automatic speech recognition.

The remainder of this paper is organized as follows. Sec. II introduces to analog voice scrambling and its variants. Sec. III and IV describe our key contributions that is how to model a "good" clean speech signal and how to exploit that for automatic descrambling. The graphical user interface described in Sec. IV-C allows the user to work with the algorithms and make manual corrections to the descrambled solution. We evaluate the proposed algorithms in Sec. V and conclude with a discussion and outlook on future work in Sec. VI.

[1] J. Walker, former US DOD employee; source: MX-COM document #20830062.002

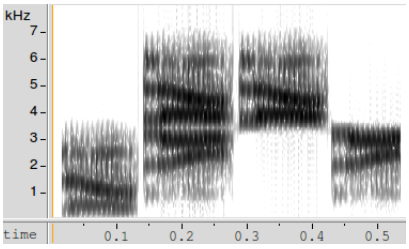

Figure 1. Spectrogram of a single word; from left to right: original, ring modulated (3400 Hz), additional high-pass and additional low-pass. The mirror effect of the ring modulation is clearly visible.

II. ANALOG VOICE SCRAMBLING

A. Frequency Inversion

The most basic scrambling technique is based on a fixed frequency inversion (*ring modulation*) that shifts all frequencies by a certain modulation frequency. This simple transform can be expressed as a multiplication in the time domain as

$$s(t) = f(t) \cdot \cos(2\pi t f_{\text{inv}} / f_{\text{SR}}) \tag{1}$$

where t is the time index, f_{inv} is the selected modulation frequency and f_{SR} is the sampling frequency. As the ring modulation introduces a mirror-like effect (everything that "falls out on top" is inserted head first at the bottom of the spectrum – hence *ring* modulation), typically a low- or high-pass filter at the modulation frequency is applied after the modulation. A high-pass filter results in a funny voice, often known from characters in animated films like Donald Duck, but a low-pass filter retains the inverted and typically unintelligible part (hence, *speech inversion*); Fig. 1 shows the spectrogram of a single word in its original, ring modulated (3400 Hz), and subsequent high- and low-pass filter.

The ring modulation is a symmetric process, i.e., it can be reversed by the same transformation; this makes it very easy to implement in both software and hardware. The most salient parts of the human voice frequency spectrum is between 100 and 4000 Hz; that range typically covers the first 3 formants which are crucial for intelligibility[2]. To break the scrambling, one needs to find the right descrambling frequency, which is similar to tuning in the right frequency in an AM/FM radio. Thus, analog voice scrambling only provides privacy, but not security. Example devices for ham radio use include the Ramsey SS70C speech scrambler/descrambler kit and the Kenwood TK 3170.

[2] Telephone codecs typically cover 300-3400 Hz (a-law, μ-law)

B. Splitband Inversion

To add a little bit to the privacy achieved by voice inversion, security companies came up with a slight modification, the *splitband* inversion. In contrast to the single frequency as with regular inversion, splitband inversion is configured by a frequency triple. The input signal is first split into a lower and upper frequency band using a split frequency f_s; then, the lower (upper) band is inverted using f_l (f_u); finally, the two signals are added together to obtain the output signal. As with the simple inversion, this process is symmetric, i.e., the same configuration is used for scrambling and descrambling. Popular devices include the MX-COM VSB chips and alike.

C. Rolling Code

Both single and splitband inversion can be wire-tapped fairly simple by a human operator setting the right frequencies. Rolling code (RC) scrambling significantly increases the privacy by a fairly simple principle: In a fixed time interval, the inversion frequencies are changed following a prior negotiated protocol; the chips synchronize using short high frequency and energy bursts. Depending on the hardware manufacturer and module, configurations may change every 80-500 msec; frequencies and their time-order may have an arbitrary length. Popular devices include the Transcrypt Int'l 400 series and the Selectone ST-25. This RC principle can be applied to both single and splitband inversion, making it rather hard for humans to decode in reasonable time.

D. Simulation Framework

Unfortunately, scrambled communications data is often strictly classified, and privacy laws and the limited accessibility of scrambling devices make it difficult to acquire authentic recordings. Thus, we implemented a simulation framework for the scrambling techniques described above to work on automatic descrambling algorithms. The de/scrambler can be configured for single and splitband inversion; additionally, a RC governor can be configured at the desired burst (frequency, energy), time interval and configuration protocol to simulate scrambling modules available on the market.

III. Modeling Speechiness

Due to the simple design of the analog scrambling process, the descrambling problem comes down to finding the right inversion frequency. Interestingly, the closer the guessed inversion frequency is to the correct value, the more natural and intelligible is the speech. While selecting an inversion frequency which is wrong by several 100 Hz results in unintelligible gibberish, a close guess may already result in a clearly understandable speech. Thus, a crucial step in finding a proper inversion frequency is to come up with a measure of *speechiness*, i.e., how good or natural a (descrambled) speech signal sounds.

A. Statistical Model

In related work on speech quality, we could show that statistical models can be used to describe and estimate inherent properties of speech such as age and gender [1] and intelligibility [2]. Based on these findings, we build a model by extracting features from the speech signal and computing a probability of being "proper" speech, i.e., that the selected inversion frequency was indeed (close to) correct.

We extract shifted delta coefficients [3] on top of Mel-frequency cepstral coefficients (MFCC), both well established features in speech and language recognition. In short, the feature extraction pipeline is

1) Apply Hamming window (25 ms length, 10 ms shift).
2) Compute power spectrum using 512 point FFT.
3) Apply Mel-filterbank (300-3400 Hz), 26 filters distributed over the Mel scale with 50% overlap; apply *log* for compression.
4) Apply DCT to compute model spectrum; output short-time energy and coefficients 1-6.
5) Compute and stack delta-frames (SDC 7-1-3-7), by

$$\Delta c_j(t) = c(t + iP + d) - c(t, +iP - d) \qquad (2)$$

where j is the cepstral coefficient in the t-th window, $d = 1$ the delta size, $P = 3$ the block shift between the deltas, and $i = 0 \dots (k-1), k = 7$ is the index of the shifted delta.

Speech and silence are modeled using Gaussian mixture models (GMM) defined as

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{N} c_i \mathcal{N}(\mathbf{x}; \mu_i, \mathbf{\Sigma}_i) \qquad (3)$$

where $\mathbf{x}$ is a feature vector and Θ are the parameters, i.e., weights c_i, means μ_i and covariances $\mathbf{\Sigma}_i$ of each mixture i; here, we use $N = 32$ Gaussians with diagonal covariance. The parameters are learned from training data in an iterative scheme, beginning from an initial estimate.

The means and variances are trained in a discriminative way maximizing the maximum mutual information (MMI) criterion

$$\mathcal{L}_{\text{MMI}}(\Theta) = \sum_{r=1}^{R} \log \frac{p_\Theta(\mathbf{x}_r|s_r)^{K_r} P(s_r)}{\sum_k p_\Theta(\mathbf{x}_r|k)^{K_r} P(k)} \qquad (4)$$

where s_r is the correct class label (voice, silence) of segment r, and the segment probability $p_\Theta(\mathbf{x}_r|s)$ is computed using the current parameters Θ; the scaling coefficient K_r is typically set to a value related to the segment length, e.g., $k_r = C/T_r$ where T_r is the length of segment r and C is a constant, e.g., 2; for detailed update formulas refer to [4].

After the re-estimation of the means and variances, the individual mixture weights c_i are estimated by maximizing the maximum likelihood objective function

$$\mathcal{L}_{\text{EM}}(\Theta) = \prod_t p(\mathbf{x}_t|\Theta) \qquad (5)$$

which also has a closed form solution as

$$\hat{c}_i = \frac{1}{N} \sum_t \gamma_t(i) = \frac{1}{N} \sum_t \frac{c_i \mathcal{N}(\mathbf{x}_t; \mu_i, \mathbf{\Sigma}_i)}{\sum_j c_j \mathcal{N}(\mathbf{x}_t; \mu_j, \mathbf{\Sigma}_j)} \quad . \qquad (6)$$

The ML step is repeated for five times, the MMI-ML routine ten times.

The speech and silence model are now used to compute a *statistical* speechiness score of a possible descrambling

attempt.

$$\tau_{\text{stat}} = \frac{1}{\sum_{t=1}^{T} \chi(\mathbf{x}_t)} \sum_{t=1}^{T} \chi(\mathbf{x}_t) \cdot p(\mathbf{x}_t|\Theta_{\text{voice}}) \tag{7}$$

where $\mathbf{x}_t$ is the feature vector, and

$$\chi(\mathbf{x}_t) = \begin{cases} 1 & \text{if } p(\mathbf{x}_t|\Theta_{\text{voice}}) > p(\mathbf{x}_t|\Theta_{\text{silence}}) \\ 0 & \text{else} \end{cases} \tag{8}$$

is the decision function to filter out silence frames, as these would have similar probability regardless of the chosen inversion frequency. This results in an average probability of each (non-silence) frame being a proper speech frame, thus, the larger the value, the more speech-like the underlying speech signal is.

B. Cepstral Peak Prominence

Beside a purely statistical measure, we extract a solely acoustic value from the presented speech signal to indicate speechiness. Human voice production is basically a two-step process; the primary excitation signal is generated by air flowing through the vocal folds. This signal is then further modulated by the vocal tract, i.e., the trachea, mouth, tongue and nasal cavities. In case of voiced (e.g., vowel) segments, the vocal folds generate a signal with a fundamental frequency (F0), which also manifests as harmonics throughout the whole spectrum. We exploit this natural property of speech; if the signal was properly descrambled, then the F0 and it's harmonics must be clearly identifiable. If we chose a wrong inversion frequency, the main F0 is shifted, thus, the resulting signal will have little harmonics for that phony F0[3]. The value and strength of the F0 and its harmonics can be found as a peak in the cepstrum, making it easy to detect.

This measure of cepstral peak prominence [5] was successfully used to evaluate the intelligibility of pathologic voices. Here, we adapt the idea and compute a more coarse estimate. Similar as with the SDC, we apply the same window function and FFT to the input signal. The Mel-filterbank contains 30 filters that cover 0-4000 Hz, equally spaced on the Mel-scale and with 50% overlap. After the DCT, we consider the first 10 coefficients, excluding the zeroth (an energy correlate).

The final per-frame measure $\delta(\mathbf{x}_t)$ is the absolute distance of the peak to the regression line fit to the remaining coefficients. Experiments show that this distance is rather small for proper speech frames, thus we consider the inverse of the absolute distance. Similar as with the statistical scoring, we also compute an average *acoustic* speechiness score that discards silence frames.

$$\tau_{\text{acou}} = \left(\frac{1}{\sum_{t=1}^{T} \chi(\mathbf{x}_t)} \sum_{t=1}^{T} \chi(\mathbf{x}_t) \cdot \delta(\mathbf{x}_t) \right)^{-1} \tag{9}$$

C. Combining Measures

The two above measures can be combined to compensate individual shortcomings. While the statistical measure can be fooled by something that "just looks like speech" by chance, the acoustical measure might be misleading if the explored frequencies were too far off the correct solution. Furthermore, one or the other might be less reliable in adverse channel conditions.

[3]Unless a harmonic of the F0 was chosen as inversion frequency.

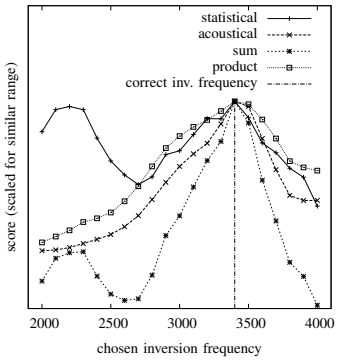

Figure 2. Speechiness values (scaled and normalized for same maximum) for a speech signal scrambled at 3400 Hz and descrambled with frequencies from 2000 Hz to 4000 Hz in 100 Hz steps.

We consider two score combinations; first, the measures can be combined as a weighted sum as

$$\tau_{\text{sum}} = w \cdot \tau_{\text{stat}} + v \cdot \tau_{\text{acou}} \tag{10}$$

where w and v are the individual weights; these can be used to transform the measures to a similar numeric range or to put emphasis on one measure. We chose $v = 0.01$ and $w = 1$ to achieve a similar numeric range of the two measures.

Similar, we can combine the two values as a product utilizing the fact that the probabilities should be high, but the peak prominence low.

$$\tau_{\text{prod}} = \tau_{\text{stat}} \cdot \tau_{\text{acou}}^{-1} \tag{11}$$

Fig. 2 shows the statistical, acoustical and combined speechiness values for an example file that was originally scrambled at 3400 Hz and then descrambled at frequencies from 2000 Hz to 4000 Hz, at a 100 Hz interval. The fact that the statistical measure shows peaks around 2200 Hz confirms, that something may appear as speech that can be ruled out by the acoustical measure.

IV. Automatic Descrambling

The automatic descrambling can be a problem of variable difficulty. In the best case, the make and model of the used scrambling module are known, and thus are the possible scrambling configurations – most chips have only a limited number of scrambling configurations with associated frequencies. In the worst case, nothing about the used language or scrambling device is known – but the statistical model is trained to recognize certain languages, and the scrambler might introduce too much noise to the acoustical features.

A. Stationary Scrambling

If the voice scrambling method is stationary, i.e., the inversion configuration is unchanged throughout the recording, finding the best inversion frequency is a straight forward task. The τ measures are computed for the whole recording; the possibly best candidate is the frequency associated with the

maximum τ value. Using a list sorted by descending τ, we can also produce a set of best guesses.

Interestingly, the experiments show that splitband inversion can be treated as regular inversion. Although the resulting voice quality is clearly lower, it seems sufficient to catch one of the two bands with a proper inversion frequency. The resulting voice will sound unnaturally high pitched or dark in timbre, as the other subband is missing. The advantage is that we only need to estimate one frequency instead of guessing the right frequency triple. However, if the scrambling module is known in advance, the possible configurations can be evaluated.

B. Rolling Code

For RC scramblers, the descrambling process is a two-step process. The segments of constant scrambling configurations need to be identified before each segment is individually descrambled.

Typically, RC devices use a high-frequency and -energy burst to synchronize the transmitter's and receiver's scrambling configuration. We identify these bursts using a sliding Hamming window (as with the feature extraction) and a threshold for the short-time energy. The segmentation is further heuristically constrained to minimize false alarms; the burst length is typically 30-50 msec, and they should appear rather regularly. The burst segmentation is a rather simple task that can be completed with a high reliability.

The second step is analog to the stationary descrambling, assuming that the configuration remains the same throughout the segment.

C. Guided Manual Descrambling

Although the automatic descrambling can produce good results, it still contains errors and, especially for RC, results in sub-optimal quality due to errors. We implemented an interface that allows the user to work with the recording in question; Fig. 3 shows an overview of the program. Starting from an initial automatic burst segmentation (in case of RC) and descrambling attempt, the user can modify the burst segmentation and the descrambling configurations for each segment. Zoom and pan for the spectrograms as well as the audio play-back functions allow the user to quickly assess the recording and produce a high quality descrambled version. The program is implemented in Java using the parts of the Java Speech Toolkit [6] and is thus platform independent.

V. Evaluation

We use a subset of the CallFriend [7] corpus, namely the training and test sets for the languages Arabic, Mandarin, German, Farsi, Spanish and Vietnamese. The corpus is a collection of phone call recordings in the above languages with varying topics. The Brno Phoneme Recognizer [8] was used to obtain a speech/non-speech segmentation of the speech data which is necessary for the training of the statistical model. To evaluate the automatic descrambling algorithms, we simulate voice scrambling for a subset of the German CallFriend test data; we use a 30 second chunk of each of the 40 available speakers.

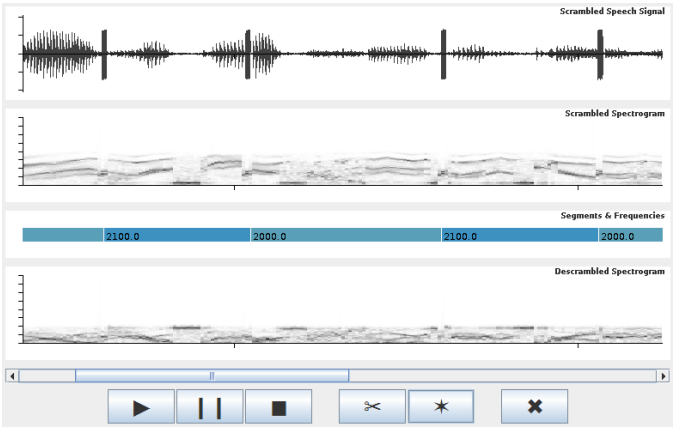

Figure 3. Screen shot of the descrambler interface; from top to bottom: original signal and spectrum, rolling code segmentation, descrambled signal spectrum, controls. Segmentation and per-segment descrambling configuration can be automatically computed and manually refined.

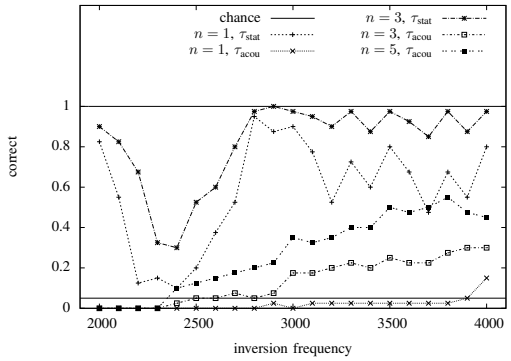

Figure 4. Recognition rate of the correct descrambling frequency by reference inversion frequency considering the $n = 1$ and $n = 3$ best guesses; the combinations are not displayed as they could not improve over the statistical measure. $n = 5$ for τ_{acou} shows that it is typically only off by little.

A. Frequency Inversion

To examine the descrambling strategy performance for the whole spectrum, we explore the de/scrambling frequencies from 2000 Hz up to 4000 Hz, with 100 Hz steps. Each recording is first inverted using a fixed frequency; the descrambler tries to find the correct frequency within the full range (2000, 2100, 2200, ..., 4000), resulting in a 5% chance of guessing the right frequency. Fig. 4 indicates a satisfactory performance using τ_{stat} for inversion frequencies above 2600, especially when considering the best three estimates; if the estimate is wrong, it is on average ca. 300 Hz off for that measure.

B. Splitband Inversion

We evaluate the splitband inversion for each of the first 16 frequency triples of the MX-COM VSB (refer to doc. #20830062.002). The automatic descramblers are provided with the same list as possible frequencies, resulting in a 6.25% chance of guessing the correct triple. Fig. 5 shows the classification performance for the individual scrambling configurations. Unfortunately, the task seems rather more

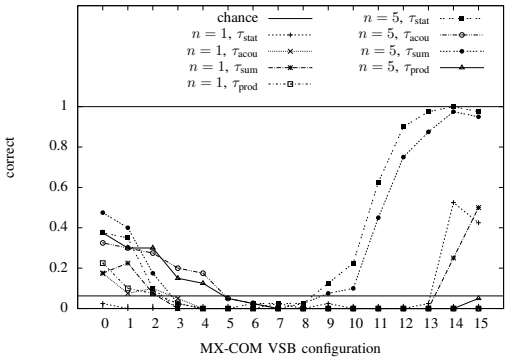

Figure 5. Recognition rate of the correct descrambling configuration by reference scrambling configuration using the $n = 1$ and $n = 5$ best guesses; the combination measures show improvements for some configurations.

difficult; the main reason is the proximity of the scrambling configurations in terms of frequency – most configurations differ by about 100-130 Hz, leading to very similar acoustic and statistical scores. Interestingly, the combination measures could show major improvements for some configurations, indicating possible future improvements.

C. Rolling Code

The RC descrambling and subsequent user interactions were not yet evaluated. We expect that the segmental descrambling needs to be improved to work on the RC chunks; the segments are typically very short (80-500 msec), making it difficult to extract reliable spectral and acoustic features.

VI. Discussion

Basic voice scrambling by frequency inversion is implemented as a ring modulation with a subsequent low-pass filter. The proposed descrambling approach is a brute-force attack; the signal is descrambled with a list of frequencies, and statistical and acoustic measures are used to identify the most promising attempt. The list of candidate frequencies is either manually selected, e.g., if the scrambling device and thus its possible settings are known, or systematically sampled; here, we chose frequencies in 2000, 2100, 2200, ..., 4000 Hz, as these can be used without discarding too much of the actual speech spectrum, and an error of 100 Hz still results in a well understandable voice. While the basic inversion frequency can be reliably detected, there is still a rather low performance for frequencies between 2200 and 2600 Hz. We suspect this due to the relatively large cutout in the spectrum resulting in similar τ values. The proposed setup can be run in about real time on typical desktop machines; the availability of clusters would allow a more detailed analysis. For future work, we are interested in fine-tuning the descrambling frequency; starting from a rough estimate, the frequency can be adjusted to maximize the harmonics in voiced segments. Instead of comparing different descrambling attempts, the inversion frequency could also be estimated directly from the scrambled signal using similar statistic and acoustic features.

Splitband scrambling is a more challenging task; depending on how much is known in advance, an automatic descrambling should be possible following some optimizations. The rather poor classification results can be explained by the proximity of the individual scrambling configurations. This proximity is also the reason why these results should be evaluated by human listeners – the guesses may be close enough to result in intelligible speech. Prior knowledge about the signal bandwidth and possibly used scrambling device can help to narrow the search space to a few frequency combinations.

The RC descrambling evaluation turned out to be tricky; although the bursts can be reliably selected, the segmental descrambling gives us a hard time. Furthermore, picking slightly wrong inversion frequencies already significantly decreases the overall intelligibility, as the pitch and formants may have an abrupt change at segment boundaries. Future work needs to address a homogeneous pitch and formant contour over segment boundaries to ensure a good intelligibility. As a by-product, this constraint may narrow down the search space for the possible inversion frequencies, resulting in a better segmental descrambling.

The presented user interface helps to compensate the shortcomings of the automatic system. The segmentation and individual scrambling configurations can be changed and immediately validated by the user to obtain high quality results.

Finally, the performance needs to be validated on real(istic) data, ideally real voice transmissions with known scrambling configurations; unfortunately, these are hard to get due to privacy and homeland security laws. Furthermore, real data may include transmissions of variable quality and include noise, squelch triggers, data carrier and dialer tones which all introduces further challenges.

Acknowledgments

This work is supported by the European regional development fund (ERDF) under STMWVT grant IUK-0906-0002 in cooperation with the Medav GmbH.

References

[1] T. Bocklet, A. Maier, J. Bauer, F. Burkhardt, and E. Nöth, "Age and gender recognition for telephone applications based on gmm supervectors and support vector machines," in *Proc. IEEE Int'l Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 1605–1608.

[2] T. Bocklet, K. Riedhammer, E. Nöth, U. Eysholdt, and T. Haderlein, "Automatic Intelligibility Assessment of Speakers After Laryngeal Cancer by Means of Acoustic Modeling," *J Voice*, 2011, *online preprint*.

[3] B. Bielefeld, "Language identification using shifted delta cepstrum," in *Proc. 14th Annual Speech Research Symposium*, 1994.

[4] P. Matejka, , L. Burget, P. Schwarz, and J. Cernocky, "Brno university of technology system for nist 2005 language recognition evaliation," in *Proc. 2005 NIST Language Recognition Evaluation*, 2005.

[5] J. Hillenbrand and R. Houde, "Acoustic Correlates of Breathy Vocal Quality: Dysphonic Voices and Continuous Speech," *J Speech Hear Res*, vol. 39, pp. 311–321, 1996.

[6] S. Steidl, K. Riedhammer, T. Bocklet, F. Hönig, and E. Nöth, "Java Visual Speech Components for Rapid Application Development of GUI based Speech Processing Applications," in *Proc. Annual Conference of the Int'l Speech Communication Association (INTERSPEECH)*, 2011, pp. 3257–3260.

[7] A. Canavan and G. Zipperlen, "LDC CallFriend Corpus," 1996.

[8] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition," in *Proc. Annual Conference of the Int'l Speech Communication Association (INTERSPEECH)*, 2005, pp. 2237–2240.