STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVII**, Number 2, 2012

CEPSTRAL-BASED SPEAKER RECOGNITION

BALŢOI IULIA-MONICA, TODEREAN ANDREEA-MARIA, AND STERCA ADRIAN

Abstract. This paper presents a simple speaker identification and classification algorithm. The algorithm uses features from the frequency domain and simple Euclidian distance for comparison. More specifically, the features for each voice sample are 5 pitch estimators constructed using cepstral coefficients. The algorithm was tested in a trial test with a collection of 10 speakers and achieved acceptable results.

1. Introduction

Human speech is a complex signal and this complexity is due to the large number of characteristics of human speech which can be viewed on different levels: acoustic, semantic, linguistic, psychological. Every person has an unique voice and even when the same person speaks the same words, the resulting sounds are not identical. The field of speech analysis has received much attention from the scientific community since the 60s up to the present day. Among important directions in speech analysis research are speech recognition and speaker recognition. Speech recognition refers to translating a an utterance into computer text. Important applications for speech recognition are voice command interfaces for mobile phones or other electronic devices, providing comfort for persons with disabilities or a more natural way of recording text for writers etc. As examples of commercial speech recognition software we can name SIRI of Apple iOS [4], Google Voice Search [5] or Windows Speech Recognition integrated in Windows Vista and Windows 7.

Speaker recognition is the activity of recognizing the person who is speaking. Speaker recognition technology takes two forms, speaker verification and speaker identification. While speaker verification means discovering the best match of an unknown speaker's identity from a list of known speakers (i.e.

Received by the editors: May 10, 2012.

2010 *Mathematics Subject Classification.* 94A12, 68U99.

1998 *CR Categories and Descriptors.* H.5.5 [**Information Systems**]: Information Interfaces and Presentation – *Sound and Music Computing*; I.5.4 [**Computing Methodologies**]: Pattern Recognition – *Applications.*

Key words and phrases. speaker recognition, voice recognition, cepstral-based sound analysis.

comparing one voice sample to voice samples of other speakers), speaker identification means verifying that a speaker is who he/she claims he/she is (i.e. comparing the speaker's voice sample to a previously recorded verified voice sample). Speaker recognition, whether verification or identification can be text-dependent (where the speaker is asked to say a specific text and the program can take advantage of the phonetic invariability of the voice samples) or text-independent (in which case the speaker can say any phrase and it is much harder for the program to find similarities between voice samples of several speakers). Applications of speaker recognition are found in forensics, microphone surveillance and various forms of authenticating services.

Related to speech and speaker recognition, in the general field of sound recognition, is music recognition which has several forms: melody recognition like performed by the Shazam software [6] music genre classification [7] or instrument separation.

The paper continues with sound recognition fundamentals in the following section, then section 3 describes our speaker recognition algorithm which is evaluated in section 4 and the paper ends with conclusions in section 5.

2. Sound recognition fundamentals

Every speaker recognition system has two components: feature extraction and classifier or speaker model. The feature extraction part refers to extracting data from the raw speech signal that identifies and differentiates the speech signal among other sound signals. Feature extraction is inevitable because a time-domain signal contains too much redundant data to use it directly for classification. Good features should encapsulate the main energy of the signal and should not contain redundant information. They also should not exhibit to much variability when extracted from another voice sample of the same speaker.

In order to apply statistical techniques on the speech signal, the signal is usually separated into frames of several tens of milliseconds long, so that the signal in a frame becomes quasistationary. Then each frame is usually multiplied by a window function (e.g. Hamming window, Hanning window etc.) to reduce the spectral leakage from applying the Discrete Fourier Transform on a finite interval.

One of the most used features are DFT coefficients. The Discrete Fourier Transform [8] of a speech signal draws the spectral envelope of that signal. Usually, only the magnitude of the spectrum is used and the phase information of the DCT coefficients is ignored.

Another much used feature is cepstrum and cepstral coefficients. The Cepstrum [8] is calculated by taking the Fourier Transform of a signal, then absolute value of the coefficients, the logarithm and finally, the Inverse Fourier Transform. The resulting complex numbers are cepstral coefficients. Cepstrum has very good information-packing properties and the coarse spectral shape is modeled by the first cepstral coefficients, so not all coefficients from the frame must be considered. Cepstrum is also useful because a convolution of two signals in the time domain is equivalent to an addition of their cepstrum in the frequency domain.

Mel-frequency cepstral coefficients(MFCC) are another useful feature for speech recognition. MFCC are very similar to normal cepstral coefficients, but they approximate better the human auditory system's response due to the mel scale.

Other used features are linear frequency cepstral coefficients [9].

The classifier or the speaker model can be nonparametric where the classifier gets two feature vectors and it determines directly (without further tests) the similarity between them or it can be parametric where prior to determining the similarity between two feature vectors, the speaker model must be trained so that various parameters of the model can be fine-tunned for the specific speaker. As example of nonparametric classifiers we mention Dynamic Time Warping (DTW) and Vector Quantization (VQ) [1] and as examples of parametric speaker models we note Gaussian Mixture Models [11] and Hidden Markov Models [12]. Dynamic Time Warping is an algorithm that measures the similarity between two vectors of different time dimensions so that the similarity result is independent of small non-linear variations of the vectors in the time domain [1]. In Hidden Markov Models for speaker recognition, a Hidden Markov model is trained to match the speech utterance to some previously known utterance. In the training phase the model's parameters are adjusted so that they maximize the probability that the model outputs the training data.

For a very good overview of speaker recognition research please see [10].

3. Cepstrum-based algorithm for speaker recognition

The speaker recognition algorithm we present and evaluate in this paper uses the cepstrum transformation as the basis for the feature selection process and a simple euclidean metric for classification and matching. The algorithm is not text-depending meaning that it does not require the speaker to pronounce a specific phrase (although in the evaluation section, we have tested it using a specific text phrase). In order to recognize a speaker from a set of previously recorded speakers, the algorithm uses as input a voice sample of the unknown speaker and compares it to voice samples of the known speakers (previously recorded) and returns the one whose voice sample is the most similar to the voice sample of the unknown speaker. The features extracted from the voice sample are pitch estimators of that voice. More specifically, we extract 5 such pitch estimators.

The workflow of the feature extraction phase of the algorithm is depicted in Figure 1. After the silence period is removed from the beginning of the voice sample, we take 5 sample windows from the voice sample, each sample window having 2048 samples and having a 50% overlap period. The purpose of choosing 2048-long sample windows is to have a quasistationary signal in a single sample window and we use 5 such sample windows in order to capture information than is not contained in the first 2048-long sample window. Each of the 5 sample windows is then passed through a Hamming window and then 2048 cepstral coefficients are determined from each sample window. The Hamming window is used to reduce the spectral leakage generated by applying the Fourier transform to a finite time interval of samples for which the period exhibits discontinuity at the edges of the interval. The cepstrum is used because it has good information-packaging properties and is very good for pitch determination of speech. From the resulting 5 cepstral coefficient windows, we take from each window the first 256 coefficients containing the frequency values with the highest energy of the sound signal (i.e. lowest frequency components) and we compute the maximum from those 256 cepstral coefficients. This maximum is an estimator of the pitch of the initial 2048-long sample window. In the end, we obtain 5 maximal cepstral coefficients from each of the 5 sample windows.

In order to compare two voice samples, we first compute the 5 aforementioned features (i.e. 5 maximal cepstral coefficients) for each voice sample and compute the Euclidean distance between those 2 vectors of 5 features each. If the Euclidean distance is bellow a threshold, then the two voice samples are similar. The overall algorithm is outlined bellow:

```
Algorithm isSimilar(voicesample source, voicesample candidate):
  src_features[] = getFeatures(source);   //get the 5 source features
  cand_features[] = NULL;   // initialize vector of candidate features
  $(w_1, w_2, w_3, w_4, w_5)$ = getFiveSampleWindows(candidate);
  for i=1 to 5 do
       Hamming($w_i$);   // apply Hamming window on each sample window
       Cepstrum($w_i$);   // compute Cepstrum
       f = $max_{k=1..256} w_i[k]$;   // get max from the first 256 cepstral coef.
  cand_features[i] = f;   // add the feature to the candidate feature set
  end for;
```

```
if EuclidDistance(src_features, cand_features) $\leq$ threshold
    then return true;    // voice samples are similar
    else return false;
end if;
```

4. Evaluation of the algorithm

In order to evaluate the speaker recognition algorithm we have recorded voice samples from 10 speakers, 4 women and 6 men [2, 3]. Voice samples were recorded at a sample rate of 8000 Hz and 16 bit quantization. Each speaker was asked to say ”salut” twice. The first voice sample of each speaker was saved in a database and the second voice sample was used to compare it against all voice samples stored in the database. The speaker recognition algorithm was implemented in the Java programming language. The results are shown in Table 1. The first column of each line from the table depicts the ID of the speaker together with his/her voice sample (i.e. the second voice sample for all speakers) that is compared against the first voice sample of each of the 10 speakers shown in the top line of the table. In general, on line i, column j the similarity between the second voice sample of speaker i and the first voice sample of speaker j it is shown. The bolded numbers from each line shows the sample voice most similar to the one used for the current line in the table.

We can see from the table that the algorithm has an error rate of 30%. This error rate is good, but not great, but if we look more careful at the table, we see that the algorithm was very close to correctly identify speakers 5 and 8 (i.e. the difference between their own voice samples is very close to the minimum difference found). There were also several other tests performed, considering only the first 128 cepstral coefficients, but the algorithm depicted in Figure 1 was the most successful from the ones we have tested.

5. Conclusions and Future work

We have presented in this paper an algorithm for speaker identification and classification. The algorithm uses features from the frequency domain for classification, cepstral coefficients to be more specific. For the classification part, is uses a simple Euclidean metric. To assess the effectiveness of the algorithm we have tested it using 10 speakers and found an identification error of 30%. In order to get a more precise assessment, we should test the algorithm

Table 1. Evaluation results

-	S 1.1	S 2.1	S 3.1	S 4.1	S 5.1	S 6.1	S 7.1	S 8.1	S 9.1	S 10.1
S 1.2	**0.023**	0.108	0.078	0.051	0.063	0.261	0.041	0.057	0.061	0.034
S 2.2	0.087	**0.053**	0.129	0.106	0.115	0.271	0.082	0.065	0.118	0.056
S 3.2	0.082	0.152	**0.039**	0.081	0.051	0.268	0.081	0.105	0.065	0.105
S 4.2	0.043	0.138	0.091	**0.023**	0.048	0.325	0.087	0.097	0.040	0.087
S 5.2	0.056	0.150	0.090	**0.033**	0.049	0.320	0.085	0.098	0.041	0.106
S 6.2	0.034	0.235	0.187	**0.024**	0.224	0.085	0.187	0.206	0.230	0.086
S 7.2	0.035	0.105	0.071	0.039	0.036	0.281	**0.030**	0.040	0.041	0.141
S 8.2	0.062	0.086	0.069	0.077	0.062	0.266	**0.031**	0.035	0.079	0.067
S 9.2	0.053	0.141	0.086	0.042	0.040	0.291	0.074	0.084	**0.016**	0.105
S 10.2	0.042	0.124	0.078	0.082	0.071	0.321	0.072	0.056	0.043	**0.025**

on a larger database of speakers. Also, the classification phase of the algorithm is rather simplistic and can be improved by using learning techniques.

6. Acknowledgments

This work was partially supported by CNCSIS-UEFISCSU, through project PN II-RU 444/2010.

References

[1] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Academic Press, 4th edition, 2008.

[2] Baltoi Iulia, *Voice Recognition in the Frequency Domain*, Diploma thesis, Babes-Bolyai University, 2011.

[3] Toderean Maria, *Sound Classification in the Frequency Domain*, Diploma thesis, Babes-Bolyai University, 2011.

[4] ***, *Speech Interpretation and Recognition Interface*, http://www.apple.com/iphone/features/siri.html.

[5] ***, *Google Voice Search*, http://www.google.com/mobile/voice-search.

[6] ***, *Shazam*, http://www.shazam.com.

[7] G. Tzanetakis, P. Cook, *Musical Genre Classification of Audio Signals*, in IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, 2002.

[8] A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing*, 3rd edition, Prentice Hall, 2009.

[9] Zhou X., Garcia-Romero D., Duraiswami R., Espy-Wilson C., Shamma S., *Linear versus Mel-Frequency Cepstral Coefficients for Speaker Recognition*, in Automatic Speech Recognition and Understanding Workshop, 2011.

[10] Kinnunen T., Li H., *An Overview of Text-Independent Speaker Recognition: from Features to Supervectors*, in Speech Communication, 2009.

[11] Reynolds D., Rose R., *Robust text-independent speaker identification using Gaussian mixture speaker models*, in IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 7283, 1995.

[12] Naik J., Netsch L., Doddington G., *Speaker verification over long distance telephone lines*, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 524527, 1989.

University of Bucharest
E-mail address: giuliasw@yahoo.com

Babes-Bolyai University, Computer Science Department
E-mail address: tais0549@scs.ubbcluj.ro

Babes-Bolyai University, Computer Science Department
E-mail address: forest@cs.ubbcluj.ro

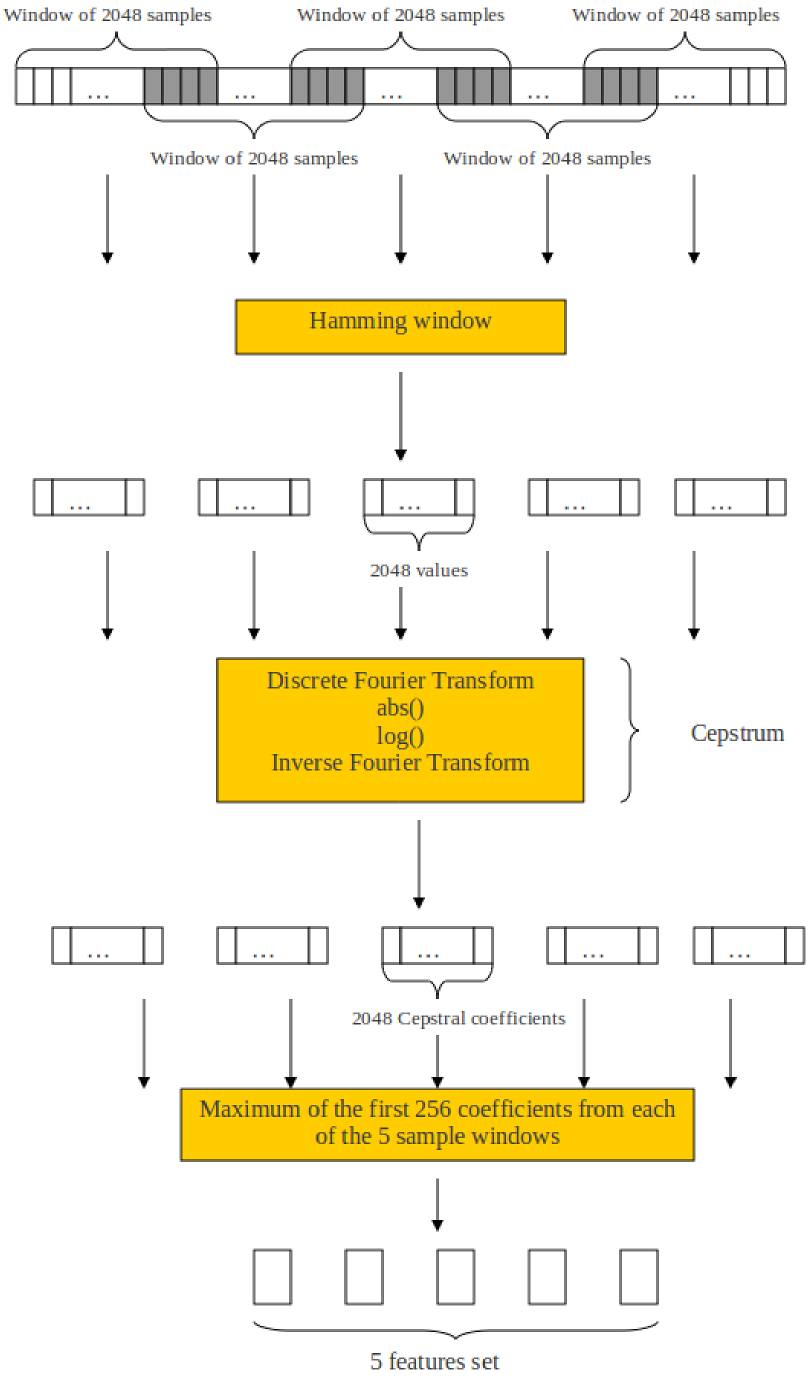

FIGURE 1. The feature extraction part of the speaker recognition algorithm