Segmentation of Speech and Humming in Vocal Input

Adam J. SPORKA, Ondřej POLÁČEK, Jan HAVLÍK

Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic

adam@sporka.eu, polacond@fel.cvut.cz, xhavlik@fel.cvut.cz

Abstract. *Non-verbal vocal interaction (NVVI) is an interaction method in which sounds other than speech produced by a human are used, such as humming. NVVI complements traditional speech recognition systems with continuous control. In order to combine the two approaches (e.g. "volume up, mmm") it is necessary to perform a speech/NVVI segmentation of the input sound signal. This paper presents two novel methods of speech and humming segmentation. The first method is based on classification of MFCC and RMS parameters using a neural network (MFCC method), while the other method computes volume changes in the signal (IAC method). The two methods are compared using a corpus collected from 13 speakers. The results indicate that the MFCC method outperforms IAC in terms of accuracy, precision, and recall.*

Keywords

Non-verbal vocal interaction, speech, MFCC, neural network, segmentation, multi-layer perceptron.

1. Introduction

Modern care uses a wide palette of approaches and technologies, commonly called assistive technology (AT), for meeting various needs of patients. The main aim of AT is to support people with various kinds of disability – physically disabled persons, mental handicapped persons, or the elderly – in their daily life. The crucial goal is to provide care which enables people to maintain their independence, social contacts, and daily habits, thereby postponing their admission to institutional care. Devices such as walkers, wheelchairs, accessible computer input devices, and telemonitoring systems are examples of assistive technology.

Many people are limited by motor impairment, which may have been caused by a variety of events, often as a consequence of an injury or paralysis after a stroke, or as a result of a neural disease such as Parkinson's, amyotrophic lateral sclerosis (ALS), or polyneuropathy. These people are limited in their daily activities. In the advanced stages of their disease, they are unable to use or to control everyday devices such as locks, phones, TV remote controllers, and computers, which are items that require fine motor coordination. A common approach for helping these patients is through speech remote control.

However, speech control is not suitable for inputting continuous data (such as continuous control of a mouse cursor or a game input device) [1]. An alternative can be provided by non-verbal vocal input (NVVI), which can be described as an interaction method in which sounds other than speech produced by a human are used. Several approaches have been described in the literature, using either the pitch of a tone, the length of a tone, volume, or vowels to control user interfaces.

This interaction method has already attracted significant attention within the research community. According to Igarashi and Hughes [2], the method has the following advantages in comparison with speech recognition – cross-cultural and language independence, continuous control, and relatively simple recognition [1].

Non-verbal vocal interaction cannot be considered a replacement for speech interaction, as the expressive capabilities of NVVI are rather limited. However, NVVI complements traditional speech recognition systems by continuous control. Igarashi and Hughes [2] suggested using the length of a tone produced after an utterance to emulate a one-dimensional joystick with immediate feedback. This would be very useful for example for moving a mouse. The user can say "move up, *mmm*" and the cursor moves up while *"mmm"* continues.

Currently, NVVI and speech recognition systems exist separately. In order to combine these two approaches so that the scenarios described above can be implemented, we need a method that analyzes the input audio signal and determines the segments containing verbal utterances and non-verbal commands. These segments will be further processed by existing speech and NVVI recognizers.

2. Related Work

Processing vocal input is a traditional area in the field of signal processing. Numerous works have been published in recent years, but most of them concern speech process-

ing, and only a small proportion deal with processing non-verbal sounds. The vital part of each speech recognizer is a speech/non-speech detection that selects parts of an input audio signal to be processed by the recognizer. A considerable amount of work exists on the speech detection. For example, Martin et al. [3] used linear discriminant analysis applied to mel-frequency cepstral coefficients, while Shafran et al. [4] used non-parametric estimation of the background noise spectrum using minimum statistics of the smoothed short-time Fourier transform. Zibert et al. [5] combined cepstral and phoneme recognition features to improve the accuracy of speech/non-speech segmentation. Several works also exist on speech/music discrimination, such as Scheirer et al. [6] or Kim et. al [7].

Significant work has been done on controlling user interfaces by pitch-based voice commands [1], [8]. A common control by pitch uses humming (producing a tone at the lips with the mouth closed, "hmmmm"), which has been evaluated by users as more convenient than whistling. The methods proposed in this paper will therefore classify the extracted segments of an input audio signal in three categories – speech, humming and silence (including other sounds, such as breathing). Other types of non-speech recognition systems – Non-speech Operated Emulation of Keyboard [9] and Non-speech input and speech recognition for real-time control of computer games [10] – have also been developed by the authors of this text.

Pruthi et al. [11] published a segmentation method that can distinguish between humming and other sounds, including speech. The method is based on computing features such as standard deviation of pitch, mean and standard deviation of a low-to-high energy ratio and the mean of the low frequency maximum. The limitations of this method are the need for constant pitch of the humming, since a maximum standard deviation of only 5 Hz is allowed. Another limitation is the fact that the input signal must continue for at least 400 ms before being classified as humming. Moreover, the method cannot perform speech segmentation. Almost no evaluation of the method has been described by the authors.

Neural network approaches have been used in several timbre recognition systems. For example, Hacihabiboglu et al. [12] used a multi-layer perceptron network for classifying short frames of musical instruments containing flute, clarinet and trumpet. Audio features were obtained from the discrete wavelet transform. The multi-layer perceptron was also used for classifying percussive sounds [13]. Several audio features were considered, e.g. zero-crossing, RMS, spectral centroid, or mel-frequency cepstrum coefficients. Another system developed by Fragoulis et al. [14] used an ARTMAP neural network to distinguish single notes played by five different instruments. Neural network approach has been also used in Vocal Joystick [15], in which the mouse cursor is controlled by vowels.

Of course, this is not an exhaustive list of all the applications of humming recognition systems that have been designed and implemented. Humming can be used not only as an assistive technology, but also for example for audio classification systems [16], [17], [18].

3. Segmentation Methods

In this section, we describe two methods for segmenting speech/humming/silence. While the first method uses MFCC and RMS features classified in a neural network (MFCC method, Sec. 3.1), the other one is based on an observation that the volume level of the sound changes more rapidly in a speech signal than in a humming (IAC method, Sec. 3.2).

3.1 Segmentation using MFCC and RMS

The segmentation method is based on classification of an audio signal by a neural network. The inputs of the network are features extracted from an audio signal. An audio signal recorded at 16 kHz is expected. First, the audio signal is divided into frames. Each frame contains 512 samples of the signal, and the step between two consecutive frames is 256 samples. The overlapping of the frames improves the time resolution of the method. Features are then extracted from each frame, as follows:

1. *Mel-frequency cepstral coefficients (MFCCs).* These coefficients are usually used in speech recognition systems [19]. First, the power spectrum of the signal is computed by the fast Fourier transform. Then the spectrum is mapped onto the mel scale by a triangular band pass filter bank with 24 triangular filters. The MFCCs are computed by taking the discrete cosine transform of the logarithms of each band pass spectrum. The mel scale maps frequency to pitch, so that the subjective step in pitch is equal to the same step in the mel scale.

2. *Low and high frequency energy.* The energy was computed as the root mean square of the amplitudes of the signal after applying low-pass and high-pass filters. The cutoff frequency was set to 350 Hz, which was found as optimal. This finding is consistent with research of Pruthi et al. [11].

After extracting the features listed above from one frame of the audio signal, a vector of 26 features is obtained (24 MFCCs and 2 energy parameters). A multi-layer perceptron (MLP) neural network is used for classifying the feature vectors. MLP is a feed-forward artificial neural network that uses supervised learning, and it is capable of approximating the outputs for a previously unseen input vector. The neurons in the MLP are organized into three layers – the first layer contains the input neurons. There are as many input neurons as there are features used for classification. The second layer is called the hidden layer, and we use 20 neurons in that layer. The last (output) layer has

three neurons in our case. Each neuron corresponds to one class (speech/silence/humming). Before using the network, the weights of the neurons are trained by a back-propagation learning strategy. In order to achieve the best performance of a neural network, the number of training vectors for each class should be approximately equal. Therefore, the number of training vectors should be limited to satisfy this condition. Another reason for reducing size of the training vectors is memory limitation in MATLAB's Neural Network Toolbox. A simple linkage clustering algorithm is used to create clusters of similar vectors. One representative vector is randomly chosen from each cluster, and these vectors are used to train the network.

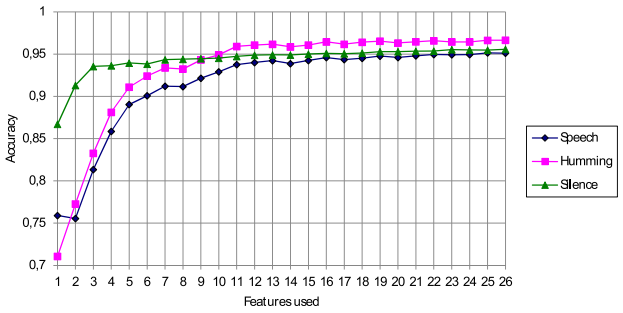

Fig. 1. Recognition accuracy of speech, humming and silence as a function of number of sorted features.

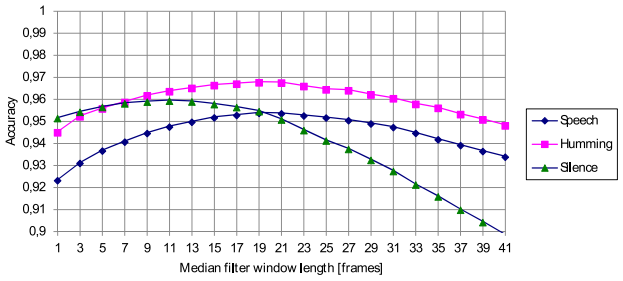

Fig. 2. Recognition accuracy of speech, humming and silence as a function of length of median filter window.

As mentioned above, a total of 26 features are extracted from the audio signal. However, it is highly unlikely that each feature conveys information that is significant for classification. In order to select the features, we used the minimum-Redundancy-Maximum-Relevance (mRMR) feature selection method [20]. This method ranks features according to their mutual dissimilarity and their similarity to the classification. The features in the vector were sorted according to the rank obtained from the mRMR method. For the ranking purposes we used data undermentioned in Section 4.1. The effect on the accuracy of the method of selecting a subset of features is depicted in Fig. 1. The figure shows that approximately the first twelve features convey significant information for classification. The use of more features does not significantly improve the classification accuracy. Reducing the set of features leads to a lower number of neurons in the input and hidden layers, and therefore to faster learning of the neural network. However, all 26 features were used for evaluation purposes.

As mentioned above, the output of the MLP classifier is a sequence of frames labeled as speech, humming, and silence. A single frame of one class should never appear alone, surrounded by frames of another class, as the input audio signal contains segments of several frames of the same class. However, the classification method can misclassify some frames. For example, parts of words with nasal phonemes (m, n, η) can easily be misinterpreted as humming. To avoid such problems, a 1D median filter is used after frame classification. A single class of frame is always replaced by a dominant class within a window of N frames. The use of the median filter improves the accuracy of the segmentation method, but it introduces a time lag of N/2 frames. The effect of window length on accuracy is depicted in Fig. 2. For the purposes of the evaluation, the length of the window was set to 17 frames. The time lag of the method was therefore 128 ms.

3.2 Segmentation using the Energy Profile

This segmentation method is based on the very simple observation that the volume level of the sound changes more rapidly in a speech signal than in a non-speech signal (humming). A very simple approach for quantifying this process is to count *important amplitude changes* (IAC) in the energy profile of a sound signal, as shown in Fig. 3.

In our implementation, a sequence of logarithms of energy level values (RMS) is calculated for each frame of the input signal. In our setup, we take frames of 1000 samples in length in a signal sampled at 16 kHz, yielding 62.5 frames per second.

The algorithm (Fig. 3) tracks the RMS energy level in the signal and adjusts the position $CENTER$ (dashed line) of a sliding interval of a fixed $WIDTH$ (solid blue line), so that the current RMS energy level (solid black line) is within this interval. The algorithm counts how many times the sliding of this interval changes its direction, i.e. how many times the RMS energy level starts exceeding the boundaries of the sliding interval one way or the other (red triangles).

The process of segmenting a sound signal is controlled by a simple state automaton (Fig. 4) that operates synchronously with the input frames. It starts in the `Idle` state. When a non-silent frame is received (i.e. a frame whose RMS energy exceeds a certain threshold Θ_{RMS}), it goes to the state `Humming`, and the utterance is labeled as humming. In this state, the number of IACs is counted. Only a history of N_{hist} frames is considered. The best results were obtained with N_{hist} of 20, which corresponds to a lag of 320 ms. This time is sufficient to determine whether an utterance contains some linguistic contents, indicated by the IACs. If the count of IACs exceeds a threshold Θ_I, the automaton goes to the state `Speech` and the whole utterance is re-labeled as speech. The automaton goes back to `Idle` after receiving more than Θ_S consecutive silent frames. (If this happens, the application needs to undo the effect of the supposed non-speech

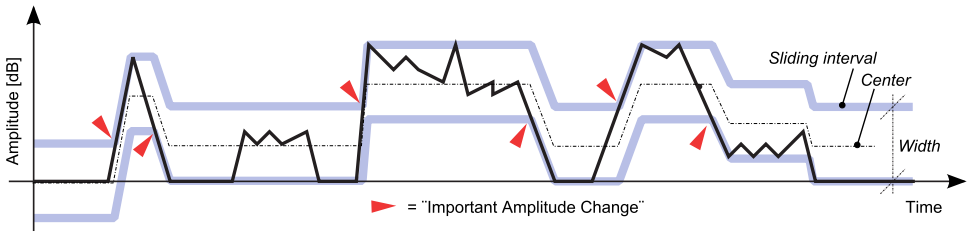

Fig. 3. Segmentation using Energy Profile: Counting the important energy changes in a signal.

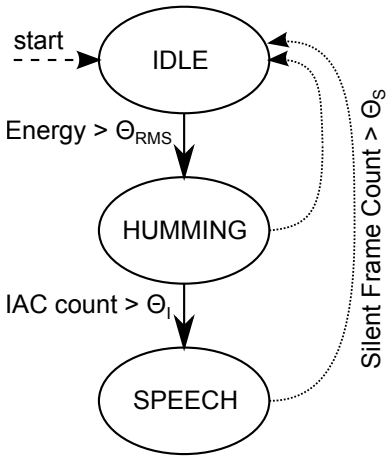

Fig. 4. Segmentation using Energy Profile: State automation.

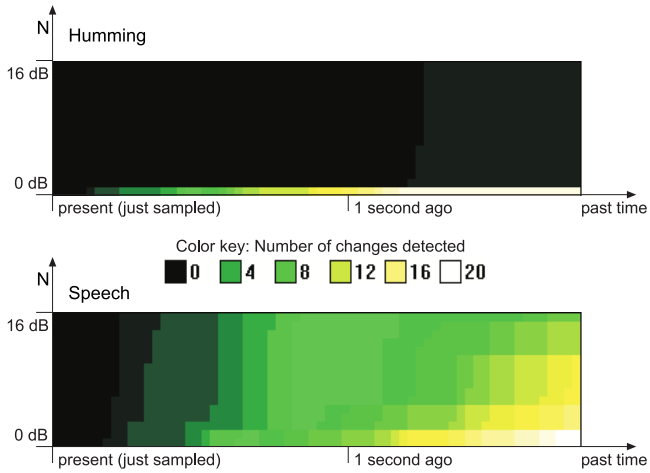

Fig. 5. Segmentation using Energy Profile: Example output for humming and speech. N is the number of important amplitude changes (IAC).

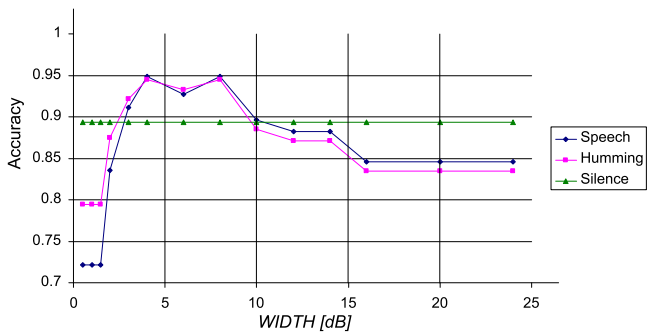

Fig. 6. Recognition accuracy of speech, humming, and silence as a function of $WIDTH$ [dB].

control so far but this is easy to implement using a simple roll-back mechanism.)

The parameters Θ_{RMS}, Θ_I, Θ_S, N_{hist}, and the sliding interval $WIDTH$ need to be set up according to the input signal. Some training is therefore required. In our evaluation a simple "Monte-Carlo optimization" was employed, where the parameters were altered randomly and the combination yielding the best performance over the training data was selected. The output for various values of Θ_{RMS} and N_{hist} is shown in Fig. 5 for humming and speech.

Fig. 6 shows the accuracy of recognition of the three classes, depending on the $WIDTH$ of the sliding energy interval. For small $WIDTH$ the method considers even tiny fluctuations of the energy important, and so there are a large number of false positives on speech. The accuracy for humming and speech is therefore low. The accuracy increases as the $WIDTH$ reaches about 6 dB which corresponds to typical oscillations of the energy level in speech. As the $WIDTH$ further increases, the ability to discriminate between speech and humming deteriorates, as larger portions of speech tend to be labeled as humming. The accuracy of the recognition of silence is constant, as only Θ_{RMS} affects this process.

The algorithm requires a certain amount of time corresponding to N_{hist} frames to determine between speech and a non-speech sound, which is a drawback of this method. However, with a careful design of voice commands and non-speech tonal patterns it is possible to minimize the impact of this constraint.

4. Evaluation

Both methods described above were used to find segments of speech and humming in a small corpus. The recognized segments were then compared to a gold standard (manually endpointed and labeled segments).

4.1 Experiment Data

The corpus was collected during a simulation of an interaction with a vector graphic editor controlled by spoken

	Speech			Humming			Silence		
Speaker	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1	0.99	0.99	0.94	0.99	0.99	0.97	0.98	0.97	0.99
2	0.99	0.98	0.98	0.99	0.99	0.98	0.98	0.97	0.98
3	0.98	0.97	0.96	0.99	0.99	0.99	0.98	0.96	0.97
4	0.99	0.99	0.97	0.99	1.00	0.97	0.98	0.96	1.00
5	0.99	0.99	0.98	0.99	1.00	0.98	0.99	0.98	0.99
6	0.99	0.98	0.97	0.99	0.99	0.97	0.98	0.98	0.99
7	0.99	0.99	0.97	0.99	1.00	0.96	0.98	0.96	0.99
8	0.98	0.93	0.95	0.98	0.96	0.98	0.97	0.97	0.96
9	0.99	0.97	0.97	0.99	0.99	0.99	0.98	0.98	0.98
10	0.99	0.96	0.97	0.99	1.00	0.98	0.98	0.98	0.99
11	0.98	0.97	0.97	0.97	0.98	0.92	0.95	0.89	0.97
12	0.98	0.94	0.98	0.98	0.97	0.99	0.97	0.98	0.94
13	0.98	0.99	0.94	0.98	0.97	0.98	0.97	0.95	0.98

Tab. 1. Performance of the MFCC method for speaker-dependent training of the neural network.

	Speech			Humming			Silence		
Speaker	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1	0.96	0.90	0.92	0.96	0.91	0.95	0.97	0.99	0.96
2	0.98	0.97	0.93	0.98	0.95	1.00	0.96	0.96	0.94
3	0.94	0.85	0.89	0.96	0.93	0.98	0.94	0.96	0.85
4	0.97	0.99	0.84	0.98	0.95	1.00	0.97	0.95	0.98
5	0.97	1.00	0.85	0.98	0.95	1.00	0.98	0.97	0.98
6	0.97	0.99	0.85	0.99	0.95	0.99	0.97	0.97	0.99
7	0.97	0.98	0.92	0.98	0.93	1.00	0.96	0.97	0.94
8	0.92	0.82	0.80	0.94	0.95	0.84	0.94	0.91	0.98
9	0.94	0.89	0.82	0.96	0.92	0.94	0.97	0.96	0.98
10	0.96	0.94	0.77	0.97	0.94	0.97	0.98	0.97	0.99
11	0.93	0.94	0.84	0.96	0.96	0.92	0.93	0.85	0.96
12	0.96	1.00	0.81	0.98	0.99	0.97	0.95	0.89	0.99
13	0.92	0.76	0.95	0.93	0.96	0.82	0.95	0.96	0.93

Tab. 2. Performance of the MFCC method for speaker-independent training of the neural network.

	Speech			Humming			Silence		
Speaker	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1	0.93	0.94	0.68	0.91	0.75	0.99	0.94	0.97	0.92
2	0.81	0.73	0.40	0.76	0.60	0.95	0.89	0.97	0.74
3	0.68	0.39	0.80	0.71	0.85	0.45	0.92	0.91	0.85
4	0.91	0.85	0.59	0.89	0.75	1.00	0.93	0.98	0.87
5	0.89	0.77	0.56	0.83	0.74	0.92	0.90	0.92	0.82
6	0.95	0.80	0.95	0.95	0.83	0.96	0.92	0.98	0.89
7	0.77	0.55	0.66	0.78	0.61	0.68	0.90	0.99	0.79
8	0.86	0.67	0.57	0.78	0.61	0.83	0.81	0.87	0.71
9	0.91	0.88	0.67	0.90	0.78	0.96	0.95	0.97	0.93
10	0.95	0.82	0.85	0.91	0.76	1.00	0.90	1.00	0.83
11	0.89	0.80	0.88	0.89	0.78	0.91	0.89	0.95	0.71
12	0.93	0.87	0.82	0.92	0.83	0.96	0.93	0.96	0.86
13	0.81	0.60	0.70	0.82	0.74	0.71	0.91	0.92	0.86

Tab. 3. Speaker-independent performance of the IAC method – parameters of the method are computed per speaker.

		Speech			Humming			Silence		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
MFCC	Average	0.99	0.97	0.96	0.99	0.99	0.97	0.98	0.96	0.98
non-overlapping	SD	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.02	0.02
MFCC	Average	0.95	0.92	0.86	0.97	0.94	0.95	0.96	0.95	0.96
overlapping	SD	0.02	0.08	0.06	0.02	0.02	0.06	0.02	0.04	0.04
IAC	Average	0.87	0.74	0.70	0.85	0.74	0.87	0.91	0.95	0.83
non-overlapping	SD	0.08	0.15	0.15	0.07	0.08	0.16	0.03	0.04	0.07

Tab. 4. Overall performance of the methods.

commands (such as "draw line", "color green", "from here", "to here") and humming commands, which were similar to commands used in emulating the mouse cursor [1]. The whole utterances therefore matched the examples provided by Igarashi and Hughes [2].

A total of 25 minutes and 34 seconds of audio data was collected from 13 speakers (4 females and 9 males) at 16 kHz. The recordings were acquired in various conditions. Each speaker used a different microphone (headset, table or laptop built-in microphone), so the quality, background noise level and volume level was different in each recording. The corpus contains 434 speech commands (each up to 4 words), and 579 humming commands. Speech utterances and humming commands were manually searched for in the audio data in order to generate a gold standard annotation. Each recording was randomly split in a ratio of 80:20, and 80 % of each recording was considered as the training set. The rest was used for evaluating the two methods. The same split was used for each method.

4.2 Results and Discussion

Accuracy, precision and recall measures are used for evaluating the two methods. The definitions of these values are shown in the formulas 1, 2 and 3. T_p stands for the number of frames with a true positive classification, F_p for false positive, T_n for true negative, and F_n for false negative.

$$\text{accuracy} = \frac{T_p + T_n}{T_p + F_p + F_n + T_n}, \tag{1}$$

$$\text{precision} = \frac{T_p}{T_p + F_p}, \tag{2}$$

$$\text{recall} = \frac{T_p}{T_p + F_n}. \tag{3}$$

The aforementioned measures are usually used for the identification of two classes only. However, the same measures can be easily used for more classes as well. The values have to be then computed separately, i.e. each class is compared with the rest. When computing the accuracy for speech, one class contains only the speech frames, while the other class contains humming and silence frames. Precision and recall values are computed for the same two classes. The measures are computed similarly for humming and silence. This yields total of nine values expressing the performance of a method.

Tabs. 1, 2 and 3 show results per speaker of both methods. Two variants of the MFCC are evaluated. In the first variant (see Tab. 1), the neural network is trained independently for each speaker. In the second variant (see Tab. 2), the neural network is trained for all speakers together. Speakers overlapped in training/testing sets, as only limited number of different speakers were present in the corpus. The performance of the second variant slightly degrades. The parameters of IAC method (see Tab. 3) has to be adjusted per speaker. The average performances of methods and variants are summarized in Tab. 4.

The methods presented in the paper can be compared from several points of view:

- *Robustness.* The MFCC method outperforms the IAC method in all values, as shown in Tab. 4. The MFCC method is therefore more robust than the IAC method.

- *Method calibration.* The IAC method must be adapted separately for each speaker. However, the configuration process is simple and fast (in seconds). On the other hand, the MFCC method can be configured for several speakers at once (13 speakers in our experiment). The configuration process for this method takes a longer time (in minutes), as the neural network has to be properly trained.

- *Real-time application.* Both MFCC and IAC methods can be used in real-time application. The delay introduced by both methods is constant: 128 ms for the MFCC method, and 320 ms for the IAC method.

The results indicate that the development of both methods is a promising step towards an assistive application that will provide interaction based on speech combined with humming commands. The short delay of the MFCC method is important for providing continuous control – while the speech segments are processed as a whole, the humming commands must be processed frame-by-frame to provide immediate feedback for the user. The accuracy of the IAC method may be improved by continuous adaptation of the the threshold values Θ_{RMS}, Θ_I. This is due to the fact that the volume of the user's speech varies over time. This would be a suitable topic of a future work.

5. Conclusion

Two methods for segmentation of speech and humming in an audio signal are described in this paper. The first one is based on computing MFCC and RMS features. Those features are processed by a neural network classifier. The other one – IAC method – is based on a simple observation that the volume level of the sound changes more rapidly in a speech signal than in a humming. Both methods were tested on a small corpus gathered from 13 speakers. The evaluation of the methods, detailed results and discussion are placed in Section 4 – Evaluation. As it was shown the MFCC method outperforms the IAC method in terms of accuracy, precision and recall.

The first step towards developing an interactive assistive application operated by a combination of speech and humming has been presented in this paper. However, more work needs to be done. Formal descriptions of speech and humming exist separately, and they need to be combined to provide an easy-to-use tool for developers. There are also no design guidelines for an interaction that combines speech and humming. The guidelines will have to accrue from extensive testing with users.

Acknowledgements

This work has been supported by research programs MSM 6840770014 and MSM 6840770012 of the Czech Technical University in Prague (sponsored by the Ministry of Education, Youth and Sports of the Czech Republic).

References

[1] SPORKA, A. J., KURNIAWAN, S. H., SLAVÍK, P. Whistling user interface (U3I). In *8th ERCIM International Workshop "User Interfaces For All"*. Vienna (Austria), 2004, p. 472 - 478.

[2] IGARASHI, T., HUGHES, J. F. Voice as sound: using non-verbal voice input for interactive control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology.* New York (USA), 2001, p. 155 - 156.

[3] MARTIN, A., CHARLET, D., MAUUARY, L. Robust speech/non-speech detection using LDA applied to MFCC. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Salt Lake City (USA), 2001, p. 237 - 240.

[4] SHAFRAN, I., ROSE, R. Robust speech detection and segmentation for real-time ASR applications. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. Hong Kong, 2003, 432 - 435.

[5] ŽIBERT, J., PAVESIĆ, N., MIHELIČ, F. Speech/non-speech segmentation based on phoneme recognition features. *EURASIP Journal on Applied Signal Processing*, 2006, vol. 2006, p. 47 - 47.

[6] SCHEIRER, E., SLANEY, M. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. Munich (Germany), 1997, p. 1331 - 1334.

[7] KIM, B.-W., CHOI, D.-L., LEE, Y.-J. Speech/music discrimination using Mel-Cepstrum modulation energy. *Lecture Notes in Computer Science*, 2007, vol. 4629, p. 406 - 414.

[8] SPORKA, A. J., ŽIKOVSKÝ, P., SLAVÍK, P. Explicative document reading controlled by non-speech audio gestures. *Lecture Notes in Computer Science*, 2006, vol. 4188, p. 695 - 702.

[9] SPORKA, A. J., KURNIAWAN, S. H., SLAVÍK, P. Non-speech operated emulation of keyboard. In *Cambridge Workshop on Universal Access and Assistive Technology, CWUAAT 2006: Designing Accessible Technology*. Cambridge (UK), 2006, 145 - 154.

[10] SPORKA, A. J., KURNIAWAN, S. H., MAHMUD, M., SLAVÍK, P. Non-speech input and speech recognition for real-time control of computer games. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. New York (USA), 2006, p. 213 - 220.

[11] PRUTHI, T., CHHATPAR, S., NGIA, L., BROWN, J., HARRIS, J. *Method for Non-Speech Vocalization to Control UGVs using Humming.* Technical manual. [Online] Available at: http://www.think-a-move.com/pdfs/AUVSIJune2008.pdf.

[12] HACIHABIBOGLU, H., CANAGARAJAH, C. Musical instrument recognition with wavelet envelopes. In *Proceedings of Forum Acusticum*. Sevilla (Spain), 2002.

[13] TINDALE, A., KAPUR, A., FUJINAGA, I. Towards timbre recognition of percussive sounds. In *Proceedings of International Computer Music Conference*. Miami (USA), 2004.

[14] FRAGOULIS, D., AVARITSIOTIS, J., PAPAODYSSEUS, C. Timbre recognition of single notes using an ARTMAP neural network. In *Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems.* Washington DC (USA), 1999, p. 1009 - 1012.

[15] BILMES, J. A., LI, X. *et al.* The vocal joystick: a voice-based human-computer interface for individuals with motor impairments. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Morristown (NJ, USA), 2005, p. 995 - 1002.

[16] BRØNDSTED, T., AUGUSTENSEN, S., FISKER, B., HANSEN, C., KLITGAARD, J., NIELSEN, L., RASMUSSEN, T. A system for recognition of hummed tunes. In *Proceedings of the COST G-6 Conference on Digital Audio Effects*. Limerick (Ireland), 2001.

[17] ZHU, Y., SHASHA, D., ZHAO, X. Query by humming: In action with its technology revealed. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. New York (USA), 2003, p. 675 - 675.

[18] LU, G., HANKINSON, T. A technique towards automatic audio classification and retrieval. In *Proceedings of the 4th International Conference on Signal Processing.* Beijing (China), 1998, p. 1142 - 1145.

[19] DAVIS, S., MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980, vol. 28, no. 4, p. 357 - 366.

[20] PENG, H., LONG, F., DING, C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27, no. 9, p. 1226 - 1238.

About Authors ...

Adam J. SPORKA is an assistant professor at the Department of Computer Graphics and Interaction at the Czech Technical University in Prague. In his research and teaching he focuses on assistive technology, non-verbal vocal input, and the user interfaces for music and sports. Currently he works on projects TextAble (using myoelectric signals as a virtual keyboard) and net-o-peer.com (a video-coaching system for snowboarders and wakeboarders).

Ondřej POLÁČEK is a PhD student at the Department of Computer Graphics and Interaction of the Czech Technical University in Prague, Faculty of Electrical Engineering. Ondřej received his MSc (2008) degree at the Department of Computer Science at the same school. In his research, he has been focusing on the pitch-based input for various assistive tools. He wrote or contributed to several papers and published in proceedings of various international conferences. He worked on several project including international EU-funded project Vital Mind.

Jan HAVLÍK received his Master degree in Electronics at the Faculty of Electrical Engineering of the Czech Technical University in Prague, Czech Republic in 2001. In 2008 he received his Ph.D. degree in Electrical Engineering Theory at the Czech Technical University in Prague. He is currently working as assistant professor at the Department of Circuit Theory, FEE CTU in Prague. His main research interests are biomedical engineering and ambient assisted living, especially in the field of medical equipment, biomedical hardware development, biomedical signal processing, telemedicine and telemonitoring.