AN EYES-FREE USER INTERFACE CONTROLLED BY FINGER SNAPS

Sampo Vesa and Tapio Lokki

Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
P.O. Box 5400, FIN-02015, HUT, Finland
svesa@tml.hut.fi

ABSTRACT

A novel way of controlling a simple user interface based on detecting and localizing finger snaps of the user is presented. The analysis method uses binaural signals recorded from the ears of the user. Transient sounds are first detected from a continuous audio stream, followed by cross-correlation based localization and simple band-energy ratio based classification. The azimuth plane around the user is divided into three sectors, each of which corresponds to one of the three “buttons” in the interface. As an example, the interface is applied for controlling the playlist of an MP3 player. The algorithm performance was evaluated using a real-world recording. While the algorithm looks promising, more research is needed before it is ready for commercial applications.

1. INTRODUCTION

Gesture-controlled auditory interfaces have been receiving some attention lately [1] [2]. However, their implementation requires special hardware, such as trackers, for detection of the gestures of the user. Low-cost solutions are desirable in many cases. In mobile augmented reality audio (MARA) there are two microphone signals, recorded from both ears of the user, available for analysis of the surrounding sound environment [3] [4]. While doing research on automatic estimation of the reverberation time [5], we came up with an idea of using finger snaps for controlling a user interface.

Transient sounds have favorable properties for use in a sound-controlled user interface. They can be easily localized in both time and space. The concentration of energy into a short time window over a relatively wide band allows azimuth localization based on the location of the cross-correlation peak. The accurate localization of the onset is easy for transients, allowing the use of only the onset for localization, thus alleviating the negative effects of the room reflections on the calculated cross-correlation. The short duration of transients also allows a fast response in the system.

2. THE METHOD

The algorithm consists of a preprocessing and detection stage followed by classification and localization stage, as shown in Fig. 1. The first stage tries to extract the preliminarily interesting sound events from the continuous stream. The events are then classified by calculating simple frequency-domain features (band-energy ratios) and evaluating their proximity to a pre-calculated feature vector which is derived as the mean of several recorded finger snaps. Sound segments that are spectrally similar enough to the mean vector are accepted. Finally, the sound events are localized based on cross-correlations calculated from the accepted segments. The cross-correlation maximum locations are then mapped to input commands to the application.

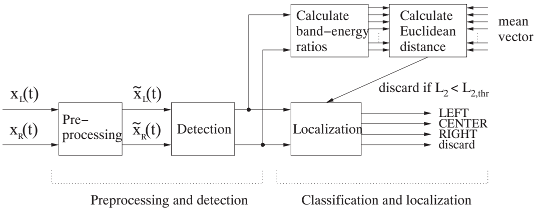
$x_L(t)$
$x_R(t)$
Pre−processing
$\tilde{x}_L(t)$
$\tilde{x}_R(t)$
Detection
Calculate band−energy ratios
Calculate Euclidean distance
mean vector
discard if $L_2 < L_{2,thr}$
Localization
LEFT
CENTER
RIGHT
discard
Preprocessing and detection
Classification and localization

Figure 1: *The overall algorithm.*

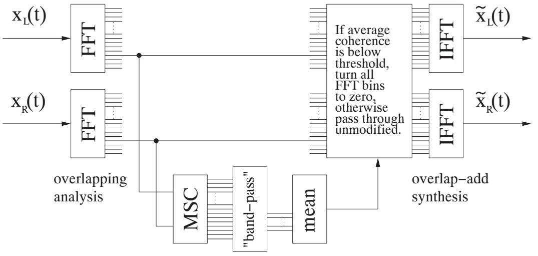

Figure 2: *Preprocessing stage of the algorithm.*

2.1. Preprocessing

The incoming sound signal is first processed by an FFT-based analysis/synthesis block that acts as some kind of a coherent signal detector illustrated in Fig. 2. The segments are windowed by a square root of a Hanning window. Short-time magnitude-squared coherence (MSC) is calculated for the kth signal block (frame) using the following set of equations [6]:

$$\hat{\gamma}^2_{lr}(f,k) = \frac{|\hat{G}_{lr}(f,k)|^2}{\hat{G}_{ll}(f,k)\hat{G}_{rr}(f,k)} \qquad (1)$$

$$\hat{G}_{ll}(f) = \langle |X_l(f,k)|^2 \rangle \qquad (2)$$

$$\hat{G}_{rr}(f) = \langle |X_r(f,k)|^2 \rangle \qquad (3)$$

$$\hat{G}_{lr}(f) = \langle X_l^*(f,k) X_r(f,k) \rangle \qquad (4)$$

where G_{lr} is the one-sided[1] cross-spectrum estimate between x_l and x_r. G_{ll} and G_{rr} are the one-sided power spectrum estimates of x_l and x_r, respectively. $X_l(f,k)$ and $X_r(f,k)$ are the Fourier transforms of the kth signal segments of the left and right signals, respectively. The discrete frequency index is denoted by f. The cross and power spectra are estimated using a leaky integrator defined for an arbitrary time-series $Q(k)$ as:

$$\langle Q(k)\rangle = \beta \cdot \langle Q(k-1)\rangle + (1-\beta)\cdot Q(k) \tag{5}$$

where k is the time index and $\beta \in [0,1]$ is a forgetting factor that adjusts the amount of smoothing[2].

The coherence function $\gamma^2_{lr}(f)$ is averaged over a certain frequency range of interest (e.g. 500-3000 Hz). This average value is thresholded to track the signal parts that have high enough coherence. A high coherence over a wide frequency band indicates a transient sound. Thus, the coherence thresholding scheme acts as a transient detector. When the coherence is lower than the threshold, all FFT bins are set to zero prior to IFFT (see Fig. 2). Otherwise the sound is passed through unmodified. It is actually not necessary to have an inverse FFT at all, since the signal could be simply switched on and off in the time domain. This way was chosen for convenience and simplicity of the real-time implementation.

2.2. Detection

Due to the fact that the coherence-based preprocessing block may occasionally let some unwanted parts of the signal through, a simple energy level check takes place after the preprocessing stage. These parts usually have very low energy compared to the transients, so they can easily be ruled out by simple energy-based thresholding. Effectively the signal is set to zero after the detection part in Fig. 1 for low-energy signal frames. No further processing is made for zero-energy frames. Only the frames with high enough energy are useful for analysis, since they are likely to contain the transients.

A sound event is detected when the short-time energy of the output of the preprocessing stage exceeds a certain level. The energy is evaluated from the left and right channel signals and the largest of the two is chosen. The threshold is chosen by hand.

2.3. Classification by band-energy ratios

The finger snaps have to be discriminated from other transients. In this algorithm a simplistic classification method was chosen, relying on band-energy ratios, which are commonly used features in general audio recognition (e.g. [7]). Band-energy ratio (BER) is calculated as the ratio of the energy on a certain band compared to the total energy. Practically the squared magnitudes of DFT bins belonging to a certain band are summed and divided by the sum of all squared bin magnitudes. The frequency bands in this application were chosen to be combined Bark bands so that three adjacent bands are merged, resulting in a total of eight bands (there are a total of 24 Bark bands).

The calculated BER vector is compared to an average vector by the Euclidean distance metric. The average vector is calculated beforehand from several finger snaps. If the Euclidean distance L_2 is below a certain threshold $L_{2,thr}$ (e.g. 0.5), the frame is accepted and a cross-correlation is calculated (see Fig. 1). This kind of classification is very crude, but it excludes the most unsuitable frames from localization.

Fig. 3 presents the average spectrum of 75 finger snaps recorded in an office environment. The Fourier transforms are calculated from 46.4 ms frames[3] by an actual real-time implementation of the algorithm. Most of the energy in finger snaps seems to be concentrated around 1500-3500 Hz. This suggests that the band-energy ratios should discriminate between finger snaps and sounds having lots of low-frequency content. It is also hypothesized that the system is able to detect cases where the sound segments containing finger snaps are corrupted by distracting noise.

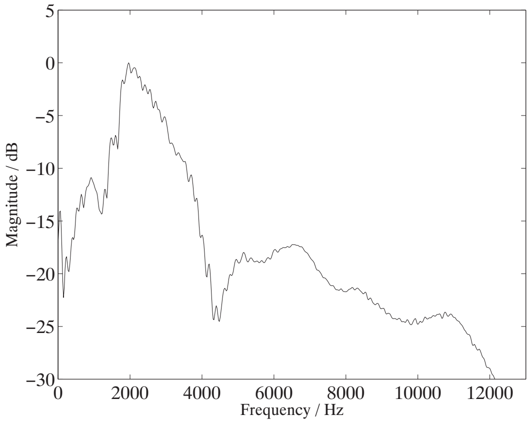

Figure 3: *Average spectrum of 75 finger snaps.*

2.4. Localization

Cross-correlation between the left and right ear signals is calculated for the first signal frame of each detected sound event. Averaging over a few frames would sound reasonable, but in this case a fast response is desired. It is also assumed that the very first frame of a sound event has the most relevant localization information, since high coherence indicates that the interaural time difference (ITD) is a reliable cue [8]. The calculations are performed using a discretized version of the following formula (based on [9]):

$$R_{lr}(\tau) = \mathcal{F}^{-1}\left\{\frac{M(f)S(f)}{|S(f)|^{\gamma}}\right\} \tag{6}$$

where $S(f) = E\{\tilde{X}_l(f)\tilde{X}_r(f)^*\}$ is an estimate for the cross-spectrum between the preprocessed left and right ear signals (see Fig. 1), $M(f)$ is a frequency-domain mask, and γ determines the amount of magnitude normalization. In this algorithm, the cross-correlation is calculated over full band in frequency-domain ($M(f) \equiv 1$), even though it might be useful to restrict the correlation calculations to a certain band. The magnitude normalization parameter γ was fixed to zero, i.e., no normalization is done, resulting in the more accurate localization results. In this algorithm, only one signal frame of the signals $\tilde{x}_l(t)$ and $\tilde{x}_r(t)$ is used for estimating the cross-spectra in Eq. (6) to yield an estimate for the

[1] This implies that f is restricted to positive frequencies in Eqs. (1)-(4).

[2] Note that setting $\beta = 0$ would result in coherence being identically one at all frequencies.

[3] 2048 samples at f_s = 44.1 kHz

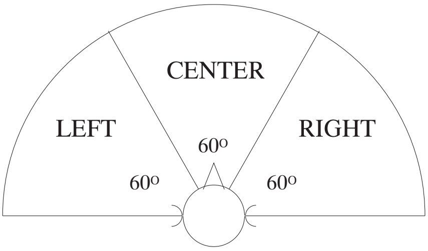

Figure 4: *Dividing the azimuth plane into three sectors.*

instantaneous cross-correlation during the finger snap onset. Effectively, Eq. (6) reduces to an inverse Fourier transform of the cross-spectrum calculated from a single frame:

$$R_{lr}(\tau) = \mathcal{F}^{-1}\left\{S(f)\right\} = \mathcal{F}^{-1}\left\{X_l(f)X_r(f)^*\right\} \qquad (7)$$

Fig. 4 shows how the front half of the azimuth plane around the user is divided into three sectors, each corresponding to one command in the interface. The maximum location of the cross-correlation is mapped to an azimuth value using the simple formulas:

$$\phi = \cos^{-1}\left(\frac{d_{maxlag}}{d_{head}}\right) \cdot \frac{360^{o}}{2\pi} \qquad (8)$$

$$\phi = \begin{cases} -(90^{o} - \phi)\ , d_{maxlag} < 0 \\ 90^{o} - \phi\ , d_{maxlag} \geq 0 \end{cases} \qquad (9)$$

where d_{maxlag} is the maximum location of the cross-correlation, converted from samples to meters, and d_{head} is the head diameter in meters. This simplified procedure is accurate enough for this application. A more elaborate version would use e.g. HRTF lookup [10]. The level differences could also be considered as an additional cue in localization.

After calculating the azimuth angle from one audio segment, the corresponding command is executed. Eq. (9) adjusts the the azimuth angle obtained from Eq. (8) so that negative azimuths are to right of the center (clockwise) and positive azimuths are towards the left (counterclockwise). If $d_{maxlag} > d_{head}$, Eq. (8) can not be used and ϕ is set to either -90° or +90°, depending on which one is closer. If the lag is considerably larger than the head diameter, it is also possible that the cross-correlation function is corrupted by echoes or by distracting noise.

Since most of the energy in finger snaps is concentrated above 1.5 kHz (see Fig. 3), the inclusion of interaural level difference (ILD) cues would be well motivated. For example, the output command to the interface could be given only if there are no contradictions between the ILD and ITD cues. Preliminary tests indicated that the robustness of the system was increased at the expense of decreased sensitivity. The ILD cues were calculated as the ratio of the left ear signal to the right ear signal (in dB), so that a positive ILD corresponds to the left ear signal being stronger. The sound segments were discarded if any of the following conditions was met:

1. ILD $<$ 0 dB and ITD indicates the left sector
2. ILD $>$ 0 dB and ITD indicates the right sector
3. abs(ILD) $<$ 3 dB and ITD indicates a sector other than the center

3. EVALUATION

The algorithm was tested with a real-world recording made in an office environment by a user wearing binaural microphones. The recording was 2 minutes long and contained 21 commands, i.e., finger snaps, mimicking a real usage situation. Each finger snap was performed by the right hand of the user. The locations of the snaps were either to the left, front or right of the user, i.e., the azimuth angle was approximately +90°, 0° or -90°.

Fig. 5 presents the results of one algorithm run in a single picture. The frame length was 46.4 ms[4], the forgetting factor in Eq. (5) was $\beta = 0.64$, the coherence threshold was 0.67 and the band for coherence averaging was 500-3500 Hz. The head diameter in Eq. (8) was set to 0.2 m. A real-time implementation of the algorithm was made using C++ and the Mustajuuri audio processing software [11] running on a 1.6 GHz Linux machine[5].

The algorithm detected 20 out of the 21 finger snaps correctly, missing one low-amplitude snap (indicated by a circle in Fig. 5). One of the snaps was incorrectly localized to the right, while the correct localization would have been at the center. The algorithm also detected a cough made by the user, which was correctly localized to the center though.

Based on this quick experiment, a localization error ratio for the algorithm was calculated as $\frac{2}{20} = 0.1$ (excluding the correctly localized cough). Thus the algorithm correctly localized 90 % of the commands. The cough could be ruled out by setting a threshold for the Euclidean distance to somewhere around 0.7, which is a reasonable choice even though one of the finger snaps would be excluded as well. More experiments on the validity of the current classification approach should be made. It is clear that a more advanced method should be used in order to reliably discriminate finger snaps from other transients. There is a lot of variation in the Euclidean distance metric among the finger snaps, as can be seen in Fig. 5.

When the ILD cues were included, the algorithm discarded the first incorrectly localized finger snap. The use of ILD cues should be investigated further and more complicated rules should possibly be developed.

While writing this paper, the primary author also occasionally used the algorithm for controlling the playlist of a software MP3 player called XMMS. The three commands were mapped to “skip backwards in playlist” (`xmms -r`), “play/pause” (`xmms -t`) and “skip forwards in playlist” (`xmms -f`). If the Euclidean distance was set low enough (around 0.4-0.5) and the energy threshold was set high enough to reject normal keyboard and mouse clicks, the user interface was found to be quite a handy way for controlling the playback.

4. CONCLUSIONS

A method for using finger snaps to control a user interface was presented, utilizing relatively simple, yet effective, signal processing concepts for localizing transients from a binaural signal. While the approach looks promising, the performance of the system should

[4] 2048 samples at $f_s = 44.1$ kHz.

[5] The CPU load during the algorithm run was 35-37% on a 1.6 GHz Linux PC.

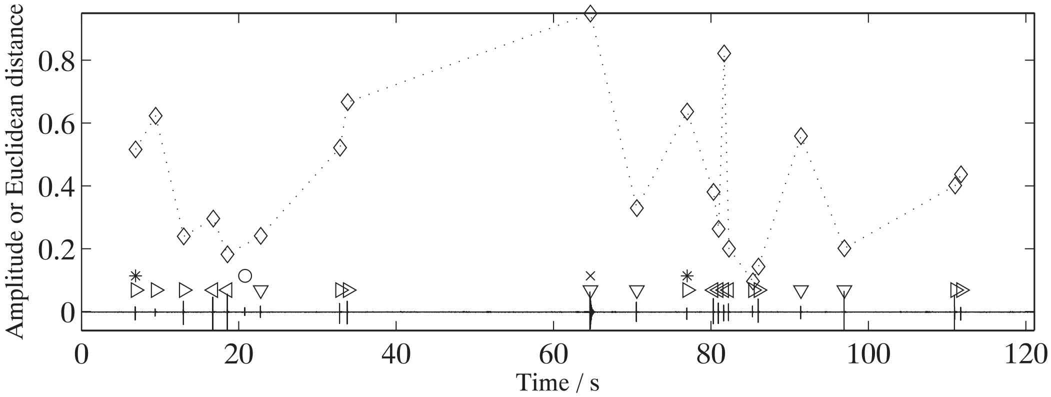

Figure 5: *Results of a test run on a recording made in an office. The triangles ◁, ▽ and ▷ indicate the moments in time the algorithm has detected a finger snap to the left, center (front) or right of the user, respectively. The asterisk (∗) indicates a localization to the wrong sector (side instead of center), the circle (○) indicates a missed finger snap and the 'x' marks a coughing sound made by the user. The diamonds (◇, connected by dotted lines for easier readability) indicate the Euclidean distance to the mean BER vector, calculated from each detected event, i.e., finger snap. Note that the Y axis represents both the Euclidean distance and the amplitude of the waveform (mean between channels), the latter being presented in the lower part of the figure.*

be increased. For commercial applications, the system should have a localization accuracy close to 100 %. Informal tests also suggest that the current system is not very robust against loud background noise, making the system useful only in relatively quiet indoor environments. Future research should concentrate on incorporating more robust localization and transient classification methods into the system. The use of feedback sounds should also be investigated, since the robustness of the system might be increased if the user gets a confirmation on each command, allowing the repetition of a missed command and the correction of a mistaken one.

5. REFERENCES

[1] S. A. Brewster, J. Lumsden, M. Bell, M. Hall, and S. Tasker, "Multimodal 'Eyes-Free' Interaction Techniques for Wearable Devices," in *Proceedings of ACM CHI 2003*, Fort Lauderdale, FL, USA, 2003, pp. 463–480.

[2] G. Marentakis and S. A. Brewster, "A Study on Gestural Interaction with a 3D Audio Display," in *Proceedings of MobileHCI2004*, Glasgow, Scotland, 2004, vol. 3160, pp. 180–191.

[3] A. Härmä, J. Jakka, M. Tikander, M. Karjalainen, T. Lokki, J. Hiipakka, and G. Lorho, "Augmented Reality Audio for Mobile and Wearable Appliances," *Journal of The Audio Engineering Society*, vol. 52, no. 6, pp. 618–639, June 2004.

[4] A. Härmä, J. Jakka, M. Tikander, M. Karjalainen, T. Lokki, H. Nironen, and S. Vesa, "Techniques and Applications of Wearable Augmented Reality Audio," in *Proceedings of the AES 114th International Convention*, Amsterdam, the Netherlands, March 2003.

[5] S. Vesa and A. Härmä, "Automatic Estimation of Reverberation Time from Binaural Signals," in *Proceedings ofthe IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2005)*, Philadelphia, PA, USA, March 2005.

[6] Thomas Wittkopp, *Two-Channel Noise Reduction Algorithms Motivated by Models of Binaural Interaction*, Ph.D. thesis, Carl von Ossietzky University Oldenburg, March 2001.

[7] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational Auditory Scene Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2002)*, Orlando, FL, USA, May 2002.

[8] C. Faller and J. Merimaa, "Source Localization in Complex Listening Situations: Selection of Binaural Cues Based on Interaural Coherence," *Journal of Acoustical Society of America*, vol. 116, no. 5, pp. 3075–3089, November 2004.

[9] M. Tikander, A. Härmä, and M. Karjalainen, "Binaural Positioning System for Wearable Augmented Reality Audio," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, New Paltz, New York, USA, October 2003.

[10] H. Viste and G. Evangelista, "On the Use of Spatial Cues to Improve Binaural Source Separation," in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx'03)*, London, England, September 2003.

[11] T. Ilmonen, "Mustajuuri - An Application and Toolkit for Interactive Audio Processing," in *Proceedings of the The Seventh International Conference on Auditory Display (ICAD 2001)*, Espoo, Finland, July/August 2001.