Intentional Voice Command Detection for Completely Hands-Free Speech Interface in Home Environments

Yasunari Obuchi, Masahito Togami, Takashi Sumiyoshi

Central Research Laboratory, Hitachi Ltd., Kokubunji, Tokyo, Japan
yasunari.obuchi.jx@hitachi.com

Abstract

We introduce a new class of speech processing, called Intentional Voice Command Detection (IVCD). It is necessary to reject not only noises but also unintended voices to achieve completely hands-free speech interface. Conventional VAD framework is not sufficient for such purpose, and we discuss how we should define IVCD and how we can realize it. We investigate implementation of IVCD from the viewpoint of feature extraction and classification, and show that the combination of various features and SVM can achieve IVCD accuracy of 93.2% for a large-scale audio database in real home environments.

Index Terms: IVCD, VAD, speech/non-speech discrimination, GMM, SVM, speech recognition

1. Introduction

One of the reasons why speech interface attracts widespread interests is that it is (expected to be) a hands-free interface. However, most of the systems with speech interface have a trigger button to activate the interface. In particular, if the system is used in noisy environments such as in home, a trigger button is essential to avoid frequent false alarms. It is more important not to work when not requested (> 99% of time) rather than to work when requested (< 1% of time). Obviously, nobody wants a TV set which switches channels every time (s)he coughs, no matter how correctly it recognizes his/her voice command.

We have been developing completely hands-free speech interface in home environments, which by definition requires no trigger button. To realize such interface with sufficiently small number of errors, it is most important to determine if the system should react to an audio input. This task seems to be very close to voice activity detection (VAD), but a small but important difference is that the system should keep quiet if the input is unintended speech such as laughs and chats. Therefore, we introduce a new class of speech processing, called intentional voice command detection (IVCD). An ideal IVCD system would output positive when the user wants the system to react, and negative when the user wants the system to keep quiet.

VAD is a task to detect active voice portions from continuous audio signals. Originally, it was a part of speech communication systems [1] and tried to reduce the bandwidth utilization and computational cost by removing silence part from the input signal. Later VAD was also applied to speech recognition systems, in which the precise endpoints of the utterance were estimated. The classic approach of VAD is based on short-term energy and zero crossing rate [2], but some recent studies focused on other features such as higher order statistics [3], order statistics filter [4], periodic structure [5], cepstral features with Gaussian mixture model [6], etc. These features were proved to be effective for specific type of noises. There is no doubt that VAD can be a basis of the development of IVCD.

IVCD can also be compared with emotion recognition [7] from speech, because estimating the user’s intention is necessary to distinguish intentional commands and other utterances. In some cases, IVCD must distinguish two utterances of the same word, one of which is a command to the speech recognition system and the other is just a monologue. Utterance verification [8] is another related topic, in which In-Vocabulary (IV) utterances are accepted and Out-Of-Vocabulary (OOV) utterances are rejected. Although IVCD must accept both IV and OOV utterances, we can learn many things from utterance verification because majority of the intentional voice commands are expected to be IV utterances.

Our approach to achieve accurate IVCD is based on combination of these studies. IVCD is defined as a two-class classification task of intentional voice command (IVC) and other, usually applied to a relatively long (1 to 2 seconds) audio segment. We investigate this task from two viewpoints: feature extraction and classification. Feature extraction includes power (short-term energy) calculation as it is still the best known feature of VAD, and also cepstral features and GMM scores. We adopt various prosodic features from emotion recognition and confidence measure from utterance verification. Those features are used to judge if the input audio segment should be accepted or rejected using various classification tools. A large-scale audio database in real home environments [9] is used to prove the effectiveness of such an approach. Although it is impossible to eliminate all false alarms, the performance of IVCD is greatly improved from a typical VAD system.

The remainder of this paper is organized as follows. In the next section, we analyze a large-scale audio database which we created to develop hands-free speech interface in home environments. We also define IVCD according to the analysis results of the database. In Section 3, various features for IVCD are investigated. Results of the evaluation experiments are presented in Section 4, and the last section gives conclusions and future works.

2. Database Analysis and Task Definition

2.1. Database overview

Prior to this work, we created a large-scale audio database in real home environments, referred to as HITHOME07. It is a collection of audio data recorded in real apartment rooms from morning to evening. There were two or three subjects in the room in each daily session and they behaved as in their ordinary lives. Their activities included cooking, vacuum-cleaning, machine-washing, chatting, watching TV, etc. They were also asked to use speech interface to control the TV set. A remote controller with a microphone was prepared, and the sub-
ject pushes a button on it and then speak to it. The subjects had a voice command list including 17 control phrases. Although the speech interface was activated only when the button was pushed, the microphone itself had been active through the recording session, and therefore continuous audio data were recorded.

**HITIHOME07** is made of 278 hour audio data from 36 sessions. We prepared two similar rooms, and the database can be divided into two subsets correspondingly. There are multiple channels of data, but only one channel was used in this work. More details about the database is found in [9].

### 2.2. Segmentation and labeling

The original database consists of 36 daylong audio streams, but they were segmented and labeled to define the task of IVCD more clearly. We first applied a coarse power-based VAD and obtained 66,140 audio segments. The threshold was set sufficiently low, so we assume that those segments include all IVCs, and there are also a huge number of other segments. For reference, applying G.729B [1], a standard VAD algorithm, to the original data resulted in 179,766 voice segments, meaning that approximately three times as many other segments as by our segmentation are included.

Next, we labeled all of the 66,140 segments as IVC or other. To reduce the labeling burden, we assumed that a segment is not an IVC if it is far from any trigger button push. About 58,000 segments were labeled as other automatically, and two human labelers checked remaining 8,000 segments in parallel. If two labels for the same segment are inconsistent, a third labeler made the final decision. Finally, we labeled 4,434 segments as IVC and 61,706 as other. Daily counts of IVC and other are shown in Fig. 1, and detailed classification of the segments are shown in Table. 1. We obtained 6,455 segments in the most active day, and only 202 segments in the quietest day. All the IVC segments were manually labeled, so we have precise labels such as power on/off, volume control, and channel control. In contrast, since most of the other segments were automatically labeled and do not have precise labels, we selected 1% of them randomly and labeled them manually.\(^1\)

It is surprising that about 90% of other segments are either laughs or chats (or their mixture). Although this percentage may be augmented by the fact that there are always two or three segments near the trigger button push, but we discarded them because the segments far from the trigger may have different trend.

\(^1\)We also have precise labels of approx. 3,500 other segments near the trigger button push, but we discarded them because the segments far from the trigger may have different trend.

### 2.3. Definition of IVCD

Following the segmentation and labeling process of the database, we define the task of IVCD as the binary classification of the segments into IVC and other. Since the number of IVCs and others are not balanced, it is not a good criterion to simply count correctly classified segments (if all segments are classified as other, 93.3% of them are correct). Therefore, we use two separate indicators, IVC Hit Rate (IHR) and Non-IVC Hit Rate (NIHR). IHR is the ratio of correctly classified IVC segments to all IVC segments. NIHR is the ratio of correctly classified other segments to all other segments. We also use Non-IVC False Alarm Rate (NIFAR) which is equal to 1.0 – NIHR. Then the overall system performance is evaluated by IVCD accuracy which is equal to (IHR+NIHR) / 2.

Figure 2 shows a typical flowchart of the speech recognition system which adopts IVCD. IVCD itself is a batch process to the relatively long audio segment. However, frame-synchronous decoding must be running in parallel to avoid a long latency in case IVCD gives an order to decode the signal. Therefore, a VAD module extracts frames which are likely to be speech and send them to the decoder. When the VAD module detects the endpoint, it tells the decoder to finalize the recognition process, and simultaneously tells the IVCD module to operate using the entire segment. If the IVCD module judges that the input should be accepted, the system operates using the speech recognition result. If the IVCD’s output is negative, nothing happens.

#### Table 1: Classification of segments. All IVC segments have precise labels. Only 1% of other segments were randomly selected and manually labeled.

<table>
<thead>
<tr>
<th></th>
<th>IVC</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3,869</td>
<td>61,706</td>
</tr>
<tr>
<td>power on/off</td>
<td>562</td>
<td>565</td>
</tr>
<tr>
<td>volume</td>
<td>1,121</td>
<td>7/620 (1.1%)</td>
</tr>
<tr>
<td>channels</td>
<td>1,286</td>
<td>52/620 (8.4%)</td>
</tr>
</tbody>
</table>

#### Table 2: Daily counts of audio segments.

<table>
<thead>
<tr>
<th>session ID (sorted by number of daily segments)</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of segments</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>565/620 (90.5%)</td>
<td>10</td>
<td>25</td>
<td>7/620 (1.1%)</td>
<td>562</td>
</tr>
</tbody>
</table>

![Figure 1: Daily counts of audio segments.](image1.png)

![Figure 2: Typical flow of speech recognition with IVCD.](image2.png)
3. Features for IVCD

3.1. Power
Power (short-term energy) is the best known feature of VAD. In the coarse VAD process before IVCD, we use the full-band power of half-overlapping frames with the frame shift of 10ms.

\[
E(i) = \frac{1}{T} \sum_{t=0}^{T} (x(t) - \langle x \rangle)^2
\]

where \( i \) is the frame index, \( x(t) \) is the input sample value of the \( i \)-th sampling point, \( \langle x \rangle \) is the average of \( x(t) \), \( \tau \) is the frame shift, and \( T \) is the frame length. The segment-level average power is a simple average of the frame power over the entire segment.

3.2. Cepstral features and GMM likelihood score
Cepstral features are known to be helpful for speech recognition. Mel-Frequency Cepstral Coefficients (MFCCs) of 10 to 15 dimension are often used. In this work, 0-th to 12-th MFCCs are used with segment-level cepstral mean normalization (CMN).

To obtain segment-level features, the variance of each dimension of MFCC is calculated (note that the average is always 0 after CMN). Therefore, the segment-level feature vector consists of 15 elements.

In the previous work of VAD, MFCCs are often used as the input of GMM likelihood score calculation.

\[
L(x_i) = \sum_{j=1}^{M} w_i N(x_i; \mu_j, \Sigma_j)
\]

where \( x_i \) is the MFCC feature vector of the \( i \)-th frame, \( w_i \) is the weight parameter, \( M \) is the number of mixtures, \( \mu_j \) and \( \Sigma_j \) are the mean and variance of the \( j \)-th Gaussian mixture, and

\[
N(x; \mu, \Sigma) = \frac{1}{\sqrt{2\pi} |\Sigma|} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)
\]

The segment-level GMM score is a simple average of \( L(x_i) \). To obtain the GMM score, we trained GMMs using HITHOME07. We separate HITHOME07 into two subsets according to the recording location, and trained an IVC GMM and a Non-IVC GMM for each subset. Each GMM is made of 1,024 Gaussian mixtures. The GMM score of subset A is calculated using GMM ob subset B, and vice versa. Finally, the difference between two GMM scores for IVC and Non-IVC was calculated.

3.3. Prosodic features
Prosodic features are based on the power \( E(i) \) and the fundamental frequency \( F_0(i) \) of each frame. We estimated \( F_0(i) \) using the YIN algorithm [10]. \( F_0 \) estimation also gives voiced/unvoiced labels to each frame. We also defined jitter \( J(i) \) as the difference of the pitch period lengths between adjacent frames. Using these fundamental variables, we then calculated 26 segment-level features listed in Table 2.

3.4. Confidence measure of ASR
As illustrated in Fig. 2, IVCD can take advantage of the confidence measure of automatic speech recognition (ASR). Although we used our proprietary ASR engine for online recognition, we used Julius [11] for IVCD evaluation because we expect that it provides more reliable confidence measures. The

### Table 2: List of prosodic features.

<table>
<thead>
<tr>
<th>Power (8)</th>
<th>Jitter (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>max ( E(i) )</td>
<td>max ( F_0(i) )</td>
</tr>
<tr>
<td>min ( E(i) )</td>
<td>min ( F_0(i) )</td>
</tr>
<tr>
<td>argmax ( E(i)/N )</td>
<td>argmax ( F_0(i)/N</td>
</tr>
<tr>
<td>argmin ( E(i)/N )</td>
<td>argmin ( F_0(i)/N</td>
</tr>
<tr>
<td>ave of ( E(i) )</td>
<td>ave of ( F_0(i) )</td>
</tr>
<tr>
<td>s. d. of ( E(i) )</td>
<td>s. d. of ( F_0(i) )</td>
</tr>
<tr>
<td>regression coef. of ( E(i) )</td>
<td>regression coef. of ( F_0(i) )</td>
</tr>
<tr>
<td>regression MSE of ( E(i) )</td>
<td>regression MSE of ( F_0(i) )</td>
</tr>
</tbody>
</table>

same vocabulary of 17 TV control commands was used by Julius as in the online recognition.

3.5. Combined features
So far we have power (1d), cepstrum variance (13d), GMM score (1d), prosodic features (26d), and ASR confidence measure (1d). Naturally we can combine all of them to create a single large feature vector. In fact, taking into account that the average of \( E(i) \) appears in the prosodic features, we eliminated the power (1d) and prepared a 41 dimension combined feature vector for each segment.

4. Experimental Results
We carried out a set of evaluation experiments using HITHOME07. First, we compared various features described above using simple thresholding and linear discriminant analysis (LDA). For single-dimension features such as power, GMM score, and ASR confidence measure, an ROC (Receiver Operating Characteristic) curve can be obtained by applying various thresholds. For multi-dimension features such as cepstral variance, prosodic features, and combined features, we first applied LDA to obtain a single score for each segment, and then applied various thresholds to obtain an ROC curve. In LDA, we divided HITHOME07 into 36 subsets according to the sessions, and applied the leave-one-out method to avoid closed-data training.

The obtained ROC curves are shown in Fig. 3. Although only the GMM score has a slightly different tendency, all the ROC curves have similar shapes. IVCD accuracies of each feature are put in the parentheses, which are obtained with the optimal threshold setting. We also tried cross validation of the threshold setting using leave-one-out of 36 sessions, but IVCD accuracies had only slight drops from 0.1 to 0.4 pts and the order of the features did not change.

Next, we investigated the influence of the classifier’s performance on IVCD accuracy. Since we confirmed that it is worth combining various features, we made some additional experiments using the combined feature and various classifiers. We already tried LDA, and it can be our baseline. Then we tested Decision Tree (DT) and Support Vector Machine (SVM). DT is a popular classification tool based on the machine learning theory. SVM is a binary classification technique characterized...
We proposed a new class of speech processing, called Inten-
tional Voice Command Detection (IVCD). It is an extension of
VAD or speech/non-speech discrimination, taking into account
that the speech interface should not respond to noises and un-
intended voices. We expect that completely hands-free speech
interface would be possible if we achieve accurate IVCD. We
investigated various features and classifiers for IVCD, and ob-
tained sufficient accuracy using a combination of various fea-
sures and SVM as a classifier. Our next step would be expan-
sion of this framework to other situations. We’ve already started
new data collection in which the subject is asked to talk to the
microphone on the ceiling, and it would evoke another future
vision of home appliances.

6. Acknowledgements

The authors are thankful to Prof. Sadao Furu of Tokyo Insti-
tute of Technology and Prof. Tetsunori Kobayashi of Waseda
University for their valuable comments. This work was sup-
ported by the Ministry of Economy, Trade and Industry, Japan.

7. References

[1] ITU-T Recommendation G.729 : Coding of speech at 8kb/s us-
ingen conjugate structure algebraic-code-excited linear prediction
the endpoints of isolated utterances,” The Bell Systems Technical
pression for robust speech recognition,” IEEE Trans. Speech and
detection based on periodic component to aperiodic component ratio,” Proc. ISCA Tutorial and Research Workshop on Statistical
and Perceptual Audition, Pittsburgh, USA, 2006.
ning GMM based Rejection of unintended inputs,” Proc. INTER-
1997.
[9] Y. Obuchi and A. Amano, “Always listening to you: Creating exhaustive audio database in home environments,” Proc. INTER-
[10] A. de Cheveign and H. Kawahara, “YIN, a Fundamental Fre-
quency Estimation for Speech and Music,” J. Acoust. Soc. Amer.,
[11] Open-Source Large Vocabulary CSR Engine Julius,
http://julius.sourceforge.jp/.
vector machine using effective feature vectors,” Proc. INTER-
[13] Weka 3 - Data mining with open source machine learning software
in java, http://www.cs.waikato.ac.nz/ml/weka/