Finding Two-level Interpersonal Context: Proximity and Conversation Detection from Personal Audio Feature Data

Masayuki Okamoto, Naoki Iketani, Keisuke Nishimura,
Masaaki Kikuchi, Kenta Cho, Masanori Hattori, Sougo Tsuboi

Corporate Research & Development Center, Toshiba Corporation, Japan
masayuki4.okamoto@toshiba.co.jp

Abstract

We propose a method to detect adhoc meeting based on cross-correlation between audio feature data, which are collected from personal mobile terminals. This method can detect whether there is conversation between each pair of users without raw audio data. Through a two-day evaluation with eight users, we found our method could detect meeting contexts with 0.9 F-measures on average. We also introduce example applications such as a document search application in which detected meeting context is used as a file annotation.

Index Terms: proximity detection, conversation detection, cross-correlation, context-aware computing

1. Introduction

The purpose of this paper is the proposal of interpersonal context detection method, which detects *when* and *with whom* a user shared an environment or talked, for community analysis or applications that use these contexts for an annotation.

There have been many works about recording and analyzing various communication contexts. In particular, grasping communications in a company or a large organization is an important topic for community analysis or supporting group work.

There are two approaches for recording these contexts: the automation of writing minutes of meetings or analyzing of utterances, and recording the fact of communication occurrence. We focus on the latter approach.

For the former approach, there are annotation-based methods [1] and methods using speech or video recognition [2].

For the latter approach, the purpose is recording and visualizing the history of communication in everyday life. To detect a conversation, not only recording voice of participants but also direct detection with IC- or IR-based devices is used.

However, there are two problems concerning the recording of communication history in everyday life. First, it is necessary to record not only conversation in meeting rooms but also informal conversation that occasionally occurs in hallways, elevators, and so on. Second, some methods with personal devices, e.g., RFID or IR [3] have another problem. Though an RFID antenna detects proximity of two or more users, it cannot detect conversation. It is difficult to detect multi-person conversation with IR-based methods, whereas it is easy to detect face-to-face situations.

To solve these problems, we propose a method to detect proximity and conversation contexts by collecting audio feature data from personal devices and calculating cross-correlation of these data. Our method has three features: a privacy-sensitive method such that no conversation can be reconstructed by other users, which is an approach similar to that in [4], two-level interpersonal context detection including not only the conversation detection but also the proximity detection which is important to analyze group activity, and that it is not necessary that the timestamps of audio features are synchronized strictly.

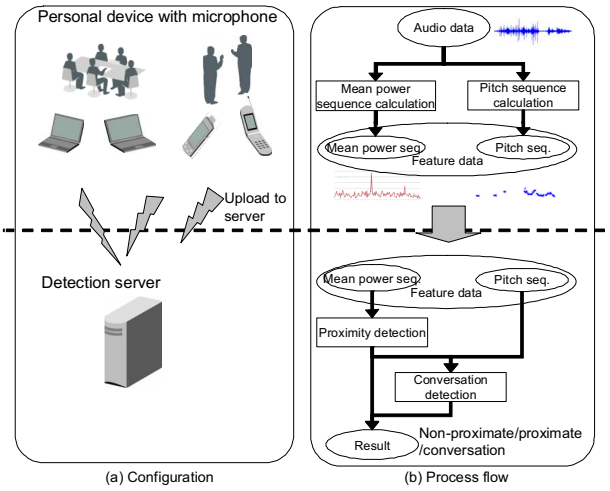


Figure 1: *Interpersonal context detection system.*

Detected context can be used for group activity analysis in an organization and as an extension of groupware or desktop applications. For example, an extension of desktop search with person-as-query functionality is a solution for users' conventional desktop search needs such as 'To whom did I show this document?' or 'Where is the slide that I showed Tom last week?'

Section 2 describes our proximity and conversation detection method. Section 3 reports an evaluation result with actual audio feature data from eight people. Section 4 introduces applications including an advanced desktop search that accepts user names for a search query.

2. Proximity and Conversation Detection

This section shows the overview of our interpersonal context detection system and detection method.

The meeting detection system consists of personal devices with microphones and a meeting detection server as shown in Fig. 1(a). Through this system, the relations between users are detected as shown in Fig. 1(b).

In this paper, we introduce the following interpersonal contexts for the detection target. One is *proximity*, which means

Copyright © 2008 ISCA

two or more users are in the same place sharing environmental audio. The distance is the social distance defined by Hall [5]. The other is the *conversation*, which means two or more users are in the proximity context *and* talking to each other. A conversation does not include short utterances such as greetings. We do not mention phone conversations in this paper.

2.1. Process Flow

Fig. 1(b) shows the flow of interpersonal context detection.

First, each user's client software on a personal device, e.g., a laptop computer, a mobile phone, or a voice recorder with a microphone, records audio feature data including *mean power sequence* and *pitch sequence* with timestamp. Then, the feature data is uploaded to the detection server. The detection server determines whether the user shared the place with another user for each pair of users using mean power sequences. If a pair of users shares the place, the server determines whether the pair shared the conversation additionally using pitch sequences. Finally, the determination result is stored.

Our detection server treats only feature data, not rare audio data. This method has the following advantages:

Two-level context detection: Our method can detect two-level interpersonal contexts, proximity and conversation, whereas the related method [4] detects only whether users share the same conversation or not. The main difference between our approach and the related methods [4, 6] is when the voiced/unvoiced speech representation is used in the detection process. These related methods first compute the mutual information that represent voiced/unvoiced speech and place two people in a conversation if their correlation coefficients are above a threshold. Whereas, our method first compares mean power sequences before using the estimated voiced/unvoiced speech representation, e.g., pitch information .

Lower network and CPU load: Compared with methods with raw audio data, data size is much smaller in the case of our method because only feature data used. In the current implementation, our system needs 2 kbps for each client. The computation cost for proximity and conversation detection is also lower. Although there is a limitation that this method detects multi-person (three or more persons') proximity or conversation contexts based on the combination of each pair's calculation result, a PC server with dual Intel Pentium 4 3.8 GHz processors can handle the combination of 50 people from our estimation.

Privacy sensitive method: The same as in the case of [4], original audio data cannot be restored from the feature data. In [7], normalized cross-correlation between raw audio signals are computed for the conversation detection. However, using raw audio data has the problem of invasion of privacy. We also think that this method determines that two people are in the same audio environment regardless of whether or not they are talking with each other. We consider this method as detecting the proximity of a pair of users, and conclude that the proximity context can be detected with rough speech feature data, e.g., mean power sequences.

2.2. Detection Method

This section describes the process of the feature-data calculation on each client and the proximity and conversation detection method on the detection server.

On a client, the feature data is calculated as follows. First, the mean power sequence is calculated from raw audio with predefined shift and frame parameter. The mean power $P(t_i)$ for a time span from t_i is calculated as $P(t_i) = \sum_{t=t_i}^{t_i+f-1} l^2(t)/f$ where $l(t)$ means the audio gain of time t, and f means the frame size. Then, pitch sequence is generated. We use a popular method by Talkin [8] using normalized cross-correlation function (NCCF) [9]. After the estimation, the candidate utterance time spans are acquired.

Created feature data including mean power sequence and pitch sequence is uploaded to the detection server. On the detection server, the two-level context detection is performed. In the proximity detection process, the confidence score is calculated based on the NCCF. We chose the NCCF-based algorithm to detect the same sound from the different users' microphones rather than to detect each user's speech from his/her own microphone. The confidence score is calculated as follows. First, NCCF $R_{fg}(t_0, m)$ between two mean power sequences w_1 and w_2 corresponding to two users u_1 and u_2 is calculated. m means the time difference between the two sequences. Then, m that maximizes the cross-correlation is determined through changing m by the step t_{step} in the range $-M \leq m \leq M$. $m > 0$ means w_2 delays and $m < 0$ means w_1 delays. We can regard m that maximizes the cross-correlation as the time difference between $w1$ and $w2$ on time t_0. After repeating the above process, a distribution histogram of time difference $d(t)$ $(-M \leq t \leq M)$ is acquired. If the ratio of $r_{conf} = \max(d(t))/\sum d(t)$ is larger than a predefined threshold r_{th} and the average cross-correlation R_{av} is R_{th} or more, the system determines that users u_1 and u_2 are close. We use r_{conf} as the *confidence score* below.

The time difference between each user's client and network delay are absorbed in the confidence score calculation step if M is large.

When it is determined that two users are close, the system also processes conversation detection with both pitch data and mean power data. The pitch data means the *sense of voice* and the mean power means the *sense of the client user*. The sense of voice is given by the length of time span in which the pitch is between a frequency range (in fact, between 50Hz and 550Hz). We call this span *speech-candidate span* in this paper. If the speech-candidate span is larger than a threshold and mean power is larger than another threshold, it is determined that two users are in the conversation context. In the conversation detection step, there are three approaches: using only pitch data, using only mean power data, and using both types of data. We compare these approaches in Section 3.

The result of the above detection process is 'not close,' 'close,' or 'in a conversation.' Below, we call the proximity detection result *proximity detection data* and conversation detection result *conversation detection data*.

3. Evaluation

In this section, we evaluated the method described in Section 2 through the two-day actual use of eight users in an office.

3.1. Method

Each user uses his client on his personal computer and inputs the correct data for each pair for each minute. Therefore, a user checks seven or less data for each minute.

We evaluated our method by comparing proximity detection data and conversation detection data with correct data labeled by eight users, through changing the threshold of mean power and speech-candidate span. The evaluation criteria used were recall, precision, and F-measure.

Table 1: *Recall, precision, and F-measure of proximity context (threshold of confidence score is 20).*

User	# of data	recall	precision	F-measure
A	1710	0.87	0.96	0.92
B	494	0.84	0.96	0.90
C	1839	0.85	0.96	0.90
D	1316	0.83	0.96	0.89
E	1715	0.82	0.91	0.86
F	1589	0.86	0.93	0.89
G	1514	0.85	0.98	0.91
H	2047	0.87	0.96	0.91

The configuration of the experimentation is shown below. Users' work was done mainly in an office, including personal work at each user's desk, small talks among a few users, and larger meetings among eight users. A part of our data is collected outside the office. Each user's client is a Windows-based laptop personal computer with a microphone. The original audio quality is 16 bit 8 kHz monaural. We used 0.125 seconds and 0.05 seconds for the frame size and the shift size of mean power sequence, respectively.

The parameter used for the proximity detection is $N = 40$, $M = 60$ seconds, and $t_{step} = 0.1$ seconds. The parameter used for the pitch estimation is the range between 50Hz and 550Hz, and 0.0075 seconds for the frame. Since the unit time of each detection in this evaluation is 60 seconds, the denominator for confidence score calculation is 600, which came from $t_{step} = 0.1$ seconds. All parameters were decided based on the preliminary investigation.

In the above settings, we collected $6,397$ proximity detection data and $4,605$ conversation detection data from actual use for two days. The number of conversation detection data is less than that of proximity detection data because some users were not sufficiently confident to label some parts of the data.

3.2. Result

This section shows the results of proximity detection and conversation detection.

First, we show the result of proximity detection. From Table 1, our method achieved a high recall and a high F-measure. This result means that the rough audio feature data useful to detect proximity context.

Next, we show the result of conversation detection. Table 2 shows the comparison of recall, precision, and F-measure among eight users. In this table, we chose an appropriate parameter for each user. From Table 2, this method detected conversation context with 0.77 or more F-measure; notably 0.9 for eight-person average.

We also investigated which factor, threshold by power or threshold by speech-interval, contributes to the F-measure. Fig. 2 shows a comparison among users E, G, and H, which is a typical pattern. The horizontal axis is common logarithm of the threshold of average power and the vertical axis is F-measure. Blue lines, red lines, and green lines are the result using only the threshold of average power, only the threshold of speech interval, and both thresholds, respectively. Fig. 2 shows results corresponding to three threshold levels of speech interval for each user.

From Fig. 2, two conclusions can be drawn. First, we can use a common threshold for speech interval if the target environment is common to some extent because the effect of changing the threshold of speech interval to the F-measure is more gradual than that of changing the threshold of average power. In this experiment, we can get the best score when the threshold parameter is $100/600$. Second, a result can be obtained more efficiently using both power and pitch than in the case of only pitch, although the contribution of the pitch information is much larger than that of the average power information. In particular, user G and H reap the benefit. However, an excessive threshold may decrease the recall.

Table 2: *Comparison between meeting-detection results (using parameters with the maximum F-measure. The threshold of pitch occurrence rate: 100/600).*

User	# of data	Recall	Precision	F-measure
A	1014	0.93	0.87	0.9
B	343	0.9	1.0	0.95
C	905	0.92	0.92	0.92
D	913	0.89	0.97	0.93
E	1261	0.89	0.96	0.92
F	1475	0.93	0.94	0.93
G	1174	0.94	1	0.97
H	1775	0.72	0.83	0.77

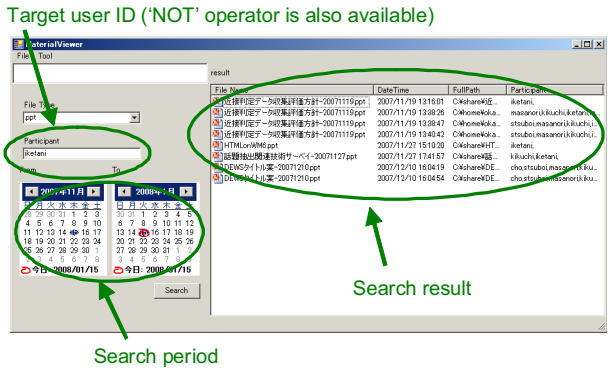


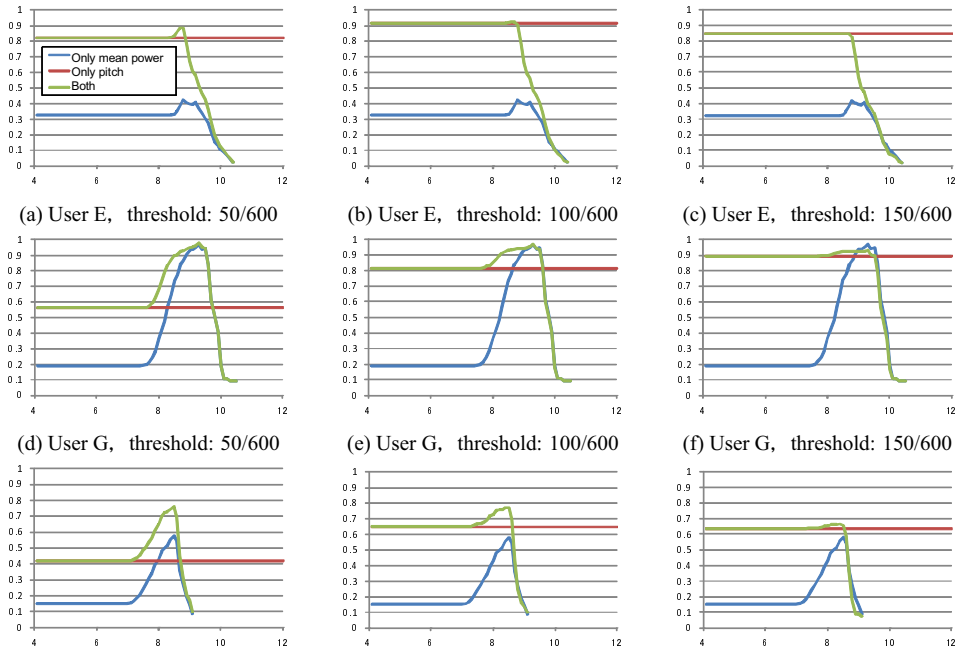
Figure 3: *Snapshot of a desktop search application.*

4. Example Applications

This section shows two example applications in which interpersonal contexts are used as a kind of annotations.

First example is a desktop search application with the file-operation history. We implemented a file-operation recording tool which stores a client user's operation including file operation, e.g., file creation, copy, and delete, and Microsoft Office application operation, e.g., file open, slide presentation, and so on. We combined these operation history with conversation participants information to add person-as-query functionality to a desktop search application. Fig. 3 shows a snapshot of our desktop search application. When a user wants to find files which was used in a conversation with a specific person, he inputs the person's ID, a file type, and a period. Then, the application asks the detection server when conversation occurred between the pair of users, combines the time and file operation history, and then outputs the result.

Second example is a visualization of group activities. Fig. 4 shows two examples of ten-minute interpersonal context history. Each node means a person, and each edge means the oc-



(a) User E, threshold: 50/600 (b) User E, threshold: 100/600 (c) User E, threshold: 150/600

(d) User G, threshold: 50/600 (e) User G, threshold: 100/600 (f) User G, threshold: 150/600

(g) User H, threshold: 50/600 (h) User H, threshold: 100/600 (i) User H, threshold: 150/600

Figure 2: *Comparison of F-measure (users E, G, and H). The horizontal axis is common logarithm of the threshold of average power and the vertical axis is F-measure.*

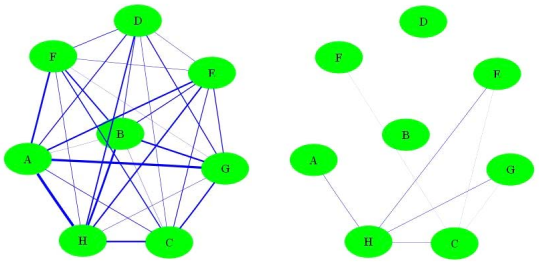


Figure 4: *Example of visualizing group activities.*

currence of proximity or conversation. The thickness of each line means the length of period of interpersonal context occurrence. When eight persons are in a meeting, the graph is strongly-connected as the left image of Fig. 4. The difference in line thickness depends on the distance between users and the voice level of each user. The right image of Fig. 4 shows the occurrence of adhoc conversation between some of users.

5. Conclusions

In this paper, we proposed an interpersonal context detection method based on users' audio feature data.

According to an evaluation with eight users, we found that the proposed method can detect proximity and conversation with 0.9 or more F-measure in total. We also showed an application with such contexts as metadata of electronic documents.

Subjects for future work include larger-size evaluation with many more users, group dynamics analysis, expansion of this method to detect multi-person conversation, and application for another domain such as music retrieval [10], TV-program matching [11].

6. References

[1] K. Nagao, K. Kaji, D. Yamamoto, and H. Tomobe, "Discussion Mining: Annotation-Based Knowledge Discovery from Real World Activities," in *Proceedings of Pacific-Rim Conference on Multimedia (PCM-04)*, pp. 522–531, 2004.

[2] S. Renals, T. Hain, and H. Bourlard, "Recognition and Understanding of Meetings The AMI and AMIDA Projects," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU-02)*, pp. 238–247, 2007.

[3] R. Borovoy, F. Martin, S. Vemuri, M. Resnick, B. Silverman, and C. Hancock, "Meme Tags and Community Mirrors: Moving from Conferences to Collaboration," in *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW-98)*, pp. 159–168, 1998.

[4] D. Wyatt, T. Choudhury, and J. Bilmes, "Conversation Detection and Speaker Segmentation in Privacy-Sensitive Situated Speech Data," in *Proceedings of Interspeech-2007*, pp. 586–589, 2007.

[5] E. T. Hall, *Hidden Dimension*, Doubleday, 1966.

[6] S. Basu, *Conversational Scene Analysis*, Ph.D Thesis, MIT, 2002.

[7] S. R. Corman and C. R. Scott, "A synchronous digital signal processing method for detecting face-to-face organizational communication behavior," *Social Networks*, vol. 16, pp. 163–179, 1994.

[8] D. Talkin, "A Robust Algorithm for Pitch Tracking," *Speech Coding and Synthesis*, pp. 495–518, 1995.

[9] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons Inc., 1973.

[10] J. Haitsma and T. Kalker, "A Highly Robust Audio Fingerprinting System," in *Proceedings of International Conference on Music Information Retrieval (ICMIR-02)*, pp. 107–115, 2002.

[11] M. Fink, M. Covell, and S. Baluja, "Social- and Interactive-Television Applications Based on Real-Time Ambient-Audio Identification," in *Proceedings of European Conference on Interactive TV (EuroITV-06)*, pp. 138–146, 2006.