PAPER Special Section on Emerging Technologies of Ubiquitous Computing Systems

Neary: Conversational Field Detection Based on Situated Sound Similarity

Toshiya NAKAKURA[†*], ***Nonmember***, **Yasuyuki SUMI**[†**a)], ***and*** **Toyoaki NISHIDA**[†], ***Members***

SUMMARY This paper proposes a system called Neary that detects conversational fields based on similarity of auditory situation among users. The similarity of auditory situation between each pair of the users is measured by the similarity of frequency property of sound captured by head-worn microphones of the individual users. Neary is implemented with a simple algorithm and runs on portable PCs. Experimental result shows Neary can successfully distinguish groups of conversations and track dynamic changes of them. This paper also presents two examples of Neary deployment to detect user contexts during experience sharing in touring at the zoo and attending an academic conference.

key words: *conversational field detection, situated sound similarity, frequency property of sound, experience sharing, ubiquitous computing*

1. Introduction

This paper proposes a system called Neary that detects and keeps up with changes of conversational fields based on situated sound similarity among users. We define conversational field, in this paper, a topological area where multiple persons join the same conversation. Recognizing group activities like group conversation is one of the most important issues to enable context-aware applications for enriching our social activities.

Anthropologist Edward Hall introduced a concept called proxemics, measurable distances between people as they interact [4]. Many researches in the domain of ubiquitous computing try to detect social contexts by estimating the users' position and mutual orientation based on proximity detection by infrared tags [2], [3], [9], [13], location detection by signal intensity of WiFi access points [5], [10], [14], visual tracking of groups of people [6], [7], etc.

Physical clusters of people could be candidates of conversational field. However, it would be difficult to determine conversational field according to size of the clusters because the physical size of conversational fields would easily vary depending on size and shape of space, crowdedness of people, and situation of social activities. Our system, Neary [8], detects conversational fields based on similarity of auditory situation among users. The similarity of auditory situation between each pair of the users is measured by the similarity of frequency property of sound captured by head-worn microphones of the individual users.

Intuitively explaining, users whose microphones receive similar sound (voice of a certain person, ambient sound, etc.) are regarded as the members of a conversation. In this method, people situated in the same sound environment are naturally grouped in the same conversational field, not depending on its physical size. The conversational fields detected by this method match to the granularity of our social activities such as meeting, lectures, group touring, etc. The method is also adjustable to various size of conversational fields from ad-hoc chatting to a lecture in a big hall.

There have been some works aiming to estimate ad-hoc groups based on ambient sound similarity. Aoki et al. [1] proposed a method to detect conversation groups from colocated multiple simultaneous conversations. Their method needs prior training with users' speech data. Wirz et al. shares the aim and approach with us and reported detailed performance evaluation of their method of proximity estimation in [12]. We are more interested in application development with simple and light implementation and this paper aims to provide practical foundings from our trials in various fields.

Neary is implemented with a simple algorithm and runs on portable PCs. Experimental result shows Neary can successfully distinguish groups of conversations and track dynamic changes of them. This paper also presents two examples of Neary deployment to detect user contexts during experience sharing in touring at the zoo and attending an academic conference.

2. Detecting Conversationonal Fields

Neary estimates the existence of conversational fields using the similarity of sound environments. Its algorithm is based on a simple and intuitive assumption: the same sound is heard in the same conversational field. For example, consider four members are colocated and speaking as shown in Fig. 1. A's utterance has a stronger influence on B than on C or D, because the nearer the sound source is, the louder it is. If they have microphones, the microphones of A and B receive A's voice louder than D's. On the other hand, the microphones of C and D receive D's voice louder than A's.

Neary compares the captured sounds and then determines that A's is more similar to B's than C's and D's. By comparing sounds, Neary recognizes and distinguishes conversational fields.

This algorithm divides the users into different groups

Manuscript received November 10, 2010.
Manuscript revised March 9, 2011.

[†]The authors are with Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[*]Presently, with NTT Communications.
[**]Presently, with School of Systems Information Science, Future University-Hakodate, Hakodate-shi, 041–8655 Japan.
a) E-mail: sumi@acm.org

DOI: 10.1587/transinf.E94.D.1164

by sound environments. If there is one loud sound in the room, whether the room is large or small, the users are recognized in one group of conversation. Even in such situation, if a certain person whispers to a next person, Neary estimates they aplit off to form another conversational field (Fig. 2). When they are split to groups of conversation, Neary detects groups of conversational fields (Fig. 3). Due to this property, Neary adapts itself to various situations, including lectures, parties, etc.

When there are many conversational fields in a small room, some conversational fields may merge, or someone may leave the conversation. Our aim is to estimate such a dynamic emergence of conversational fields with a simple and light implementation on mobile computing technologies.

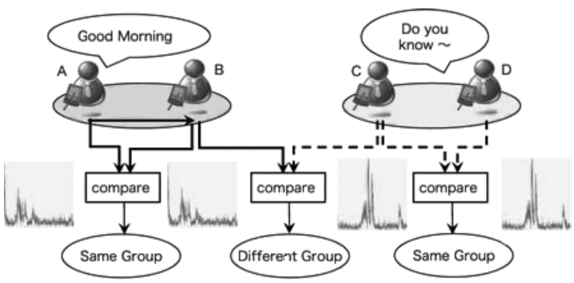

Fig. 1 Basic idea of detecting conversational fields.

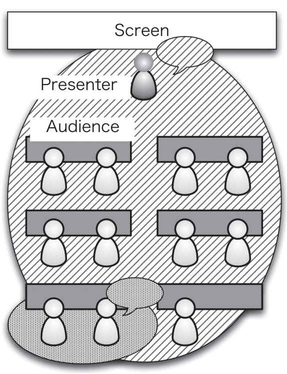

Fig. 2 Lecture setting and some attendees are whispering.

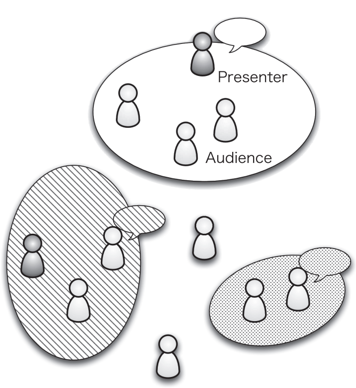

Fig. 3 Multiple conversational fields in a room.

3. Implementation of Neary

3.1 Basic Architecture

This paper proposes a method to detect conversational fields based on similarity of auditory situation among users. The similarity of auditory situation between each pair of the users is measured by the similarity of frequency property of sound captured by head-worn microphones of the individual users. The frequency property of each captured sound is calculated by Fast Fourier Transform (FFT) at regular intervals (six seconds in this paper), and the similarity of them between each pair of the users is calculated with the cosine similarity of the frequency properties.

Neary is designed to run on mobile PCs. Its current implementation employs an ad-hoc network by wireless peer-to-peer connections among Neary machines that communicate with each other by this network and mutually send data (output of FFT) for calculating the similarity of auditory situation between each pair of the machines. The ad-hoc network enables us to use Neary anywhere without servers and wireless access points.

Choosing adequate microphones in Neary's algorithm is important. We use a bluetooth headset microphone, which is omnidirectional, so it can record sound independent of user orientation. It also has another significant property: wirelessness. It does not disturb human communications. We chose an ominidirectional microphones since our aim is to estimate the sound similarity between ambient sound which individuals are situated in. The ominidirectional microphones are better than directional ones to detect the people situated in the same sound environment having sound given with loudspeaker and big noise as well as the people talking in a face-to-face manner. Also, the ominidirectional microphones are good for detecting not only conversation partner's voice but also the wearer's voice. Figure 4 shows Neary's users and devices.

3.2 Detection Algorithm

Figure 5 shows Neary's system chart and data flow.

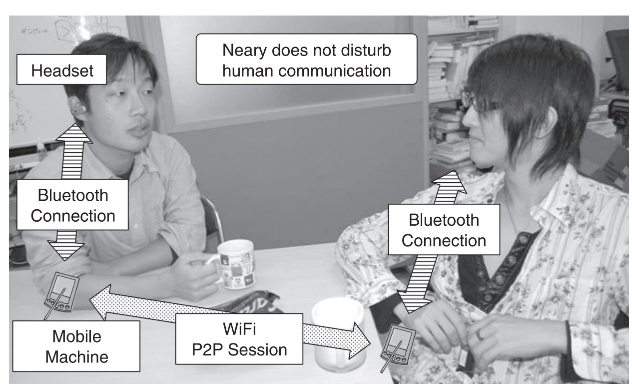

Fig. 4 User and device of Neary.

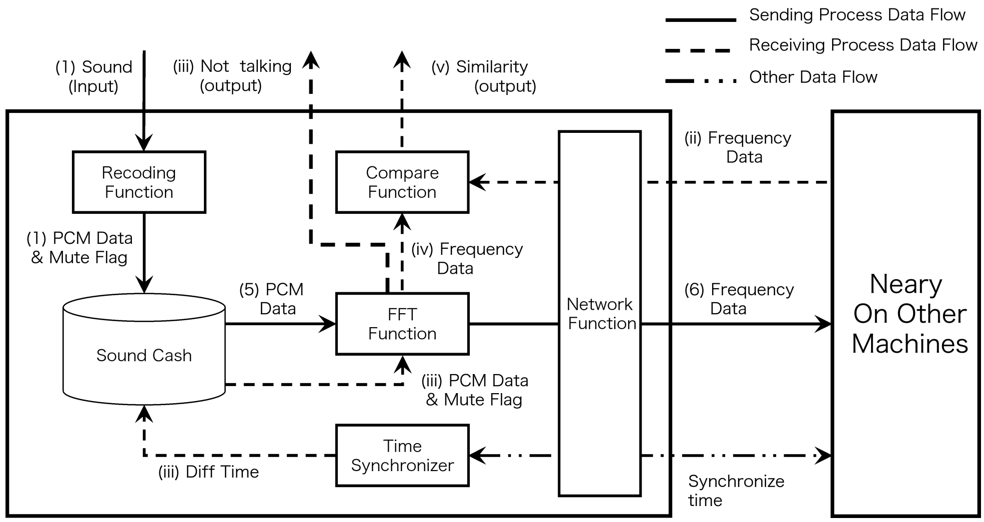

Fig. 5 Neary's system architecture and data flow.

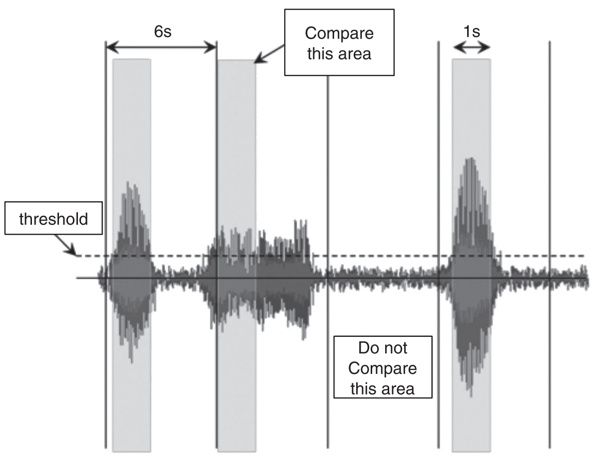

Fig. 6 Sound input and area compared by Neary.

Neary compares the frequency property of input sound. When the input sound is silent, clearly nobody is talking, so Neary assumes there is no conversational field without calculating. If the input volume is lower than the average of input volume (Fig. 6), Neary regards the input sound as silence. The detailed algorithm is as follows:

Sending process

1. Prepare six-second sound buffer.
2. Cue sound into the buffer.
3. Scan the buffer to find a sounded region every six seconds.
   If it only finds a mute region, go to 7.
4. Extract one second of sound data from the buffer.
   In this time, the system chooses the region with the longest sounded region.
5. Process the sound data with FFT, and put a timestamp and user name on the output.
6. Send the output to all other Neary devices.
7. Refresh the sound buffer.
8. Go to 2.

Receiving process

i. Wait to receive the Neary data.
ii. Get the timestamp and user name from the received data.
iii. Get the sound data recorded at the time shown on the timestamp from the buffer.
   If these sound data are silent, output a false value and go to i.
iv. Process the sound data with FFT, and calculate the cosine similarity of these two bits of data.
   If the value is over a threshold, Neary outputs a correct value and otherwise outputs a false value.
v. Go to i.

Neary extracts one second of non-silent area from six seconds sound buffer and transforms it into frequency information by FFT. The quality of recording sound is 16 kHz, 16 bit PCM data. The step number of FFT is 32768 (= 2^{15}). Neary transmits the data to the other machines via P2P network. The data size transmitted between the machines is extremely smaller than sound data, which does not cause the main reason of congestion among the machines gathering in a restricted space. Practically, we do not have any congestion caused by Neary when we deployed Neary running on eight machines in the same room and the museum.

The threshold in Fig. 6 was decided as an average value of environmental noise during ten seconds without speaking when initiating Neary. We have various type of environmental noise such as the fan and driving sound of electronics devices in the real fields. Humans do not notice the environmental noise as foreground sound, and then find meanings from different sound over it. Therefore, we regard the threshold as offset.

The buffer size (six seconds) was decided by practical reasons. If we make the buffer extremely short, the Neary's esitimation will be too sensitive to short breath and the cost of calculation and transmission of FFT data will unreasonablly increase. On the other hand, we assume there is a limit of silence duration for maitaining natural conversations. We intended the threshold would be settled between the above two values, and specified six seconds as the threshold by observing several daily conversations.

Every six seconds users get Neary's output, which consists of user names. For example, when Neary estimated users "Thomas" and "William" as conversation partners of user "George," "George"'s Neary output "Thomas, William."

Neary uses not only human voices but also other auditory data such as music. When persons are listening to the same music and seem to begin to talk, Neary regards them as being in the same conversational field. Neary uses 50 ~ 1600 Hz for its comparisons. The number of vector dimensions with which cosine similarity is calculated is 1,551. Human voices and 90% of piano keyboards are found in this band. Since the piano covers all classical music frequencies, Neary covers most music.

Synchronization of time

Since we must compare the frequency of the sounds that appear in the same time, we implemented the simple Network Time Protocol (NTP) to synchronize the system time of the Neary machines in the following order.

a. Get the system time, and send it to the other machines with a User Datagram Protocol (UDP).
b. Subtract the received system time from the system time. This difference is t_1.
c. Return t_1 and the system time.
d. Subtract the system time from the received system time. This difference is t_2.
e. Calculate $\frac{t_1+t_2}{2}$ as the difference of the system clocks.
f. Return and save the difference of the system clocks.

Neary gets the system time in 100-ns order and the margin error of the time difference of the machines in 10-ms order. Due to the property of the FFT Window Function, this error has little influence on the comparison results.

4. Experimental Evaluation

If there are many people in a room, various forms of conversational fields may be found. For example, the number of conversational fields may change, or some people may move to another conversational field. Since Neary's main purpose is to keep up with these changes, we examined whether Neary can detect them by designing a task for a conversational field that sometimes changed form.

4.1 Experiment Design

We divided four participants into two groups to debate the distribution of lab equipment between two rooms. The equipment included both appropriate equipment and some too large for the room. In this experiment, we used a whiteboard, a partition, and two desks, as shown in Fig. 7. After this, we refer to these four participants as $A_1, A_2, B_1,$ *and* B_2. A_1 *and* A_2 are group A, and B_1 *and* B_2 are group B.

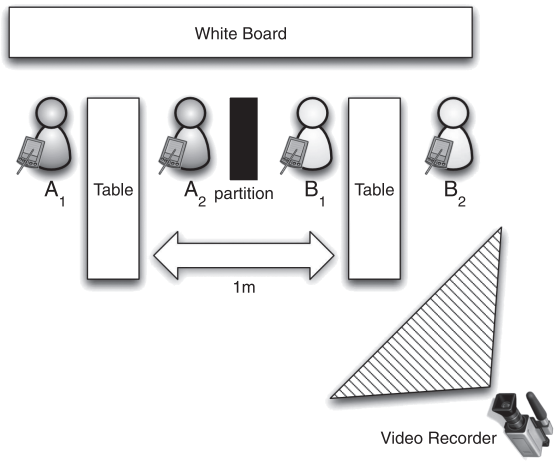

Fig. 7 Experiment setting.

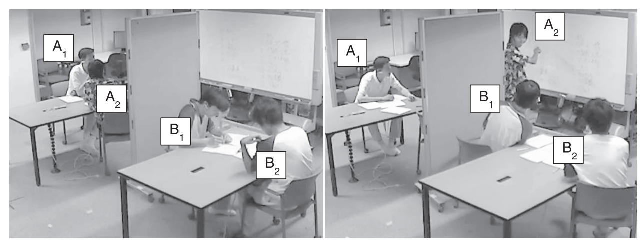

Fig. 8 Experiment states.

First, each team discussed separately what equipment they needed (Fig. 8: left). To avoid conversation between teams, we put a 120-cm wide, 5-cm deep, and 200-cm high partition between the groups. There were two conversational fields in the room. Neary was supposed to detect the two groups: A_1 *and* A_2 and B_1 *and* B_2.

After this discussion, the teams negotiated with each other (Fig. 8: right) in phases that consisted of four turns in the order of $A_1, B_1, A_2,$ *and* B_2. In A_1's turn, A_1 went to the B group table and negotiated with them. A_2 stayed at the A group table, but could join the negotiation by speaking loudly. In this task, there was either a group of three or four. Neary's goal was to detect the group.

We recorded this experiment with a camcorder to show how the conversational fields changed and to estimate Neary's accuracy.

4.2 Results and Discussion

Two group conversations

This phase featured two conversational fields, as shown in Fig. 8: left. In such situation, Neary was expected to divide

Table 2 Conversation samples and Neary's estimation of conversation partners for each user during the negotiation phase.

Conversation		A_1	A_2	B_1	B_2
A_2 is negotiating with B_1 and B_2. A_1 is apart from them.					
$A_2 \to B$	"First, we have two large desks. That's unfair."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_1 \to A_2$	"True, but... "	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \to A_2$	"No, no. We don't need such desks."	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_1 \to A_2$	"Our room doesn't have enough space."	A_2	A_1, B_1, B_2	A_2, B_2	A_2, B_1
	long silence	..	..	..	..
$A_2 \to B$	"Then you should take this large-sized display."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \to A_2$	"No. We already have the same display."	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_2 \to B_2$	"OK. I'll stop forcing this display on you. Take this desk."	..	B_1, B_2	A_2, B_2	A_2, B_1
	laughter	A_2, B_1, B_2	A_1, B_1, B_2	A_1, A_2, B_2	A_1, A_2, B_1
$A_2 \to B$	"We don't need it!"	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_2 \to B$	"We'll take this display, and you take this desk."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \to A_2$	"Not our job. Give up. They were in your room from the beginning."	..	B_1, B_2	A_2, B_2	A_2, B_1
From behind the partition, A_1 joins the conversation in a loud voice.					
$A_1 \to B$	"The initial condition was unfair. We had lots of large equipment in our room."	A_2, B_1, B_2	A_1, B_1, B_2	A_1, A_2, B_2	A_1, A_2, B_1
$A_2 \to A_1$	"I agree."	A_2, B_1, B_2	A_1, B_1, B_2	A_1, A_2, B_2	A_1, A_2, B_1
$B_2 \to A_2$	"If you force that large stuff on us, I'll do the same to you on our next turn."	..	B_1, B_2	A_2, B_2	A_2, B_1
	...				

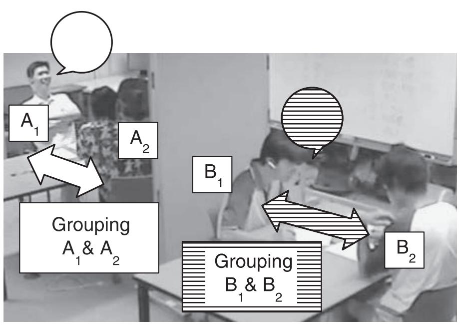

Fig. 9 Two group conversations and Neary output.

Table 1 Neary's estimation of conversation partners for each user during the group conversation phase.

User	Number of times that Neary estimated the below users as a conversation partner			
	A_1	A_2	B_1	B_2
A_1	-	15	1	4
A_2	23	-	1	2
B_1	4	4	-	21
B_2	3	1	18	-

users into groups A and B (Fig. 9).

In five minutes, Neary generated 52 outputs, as shown in Table 1.

Table 1 presents the number of the Neary's detection of conversation partners for each user during the group conversation phase. For example, the first row in the second column indicates that there were 23 times that Neary detected A_1 as conversational partners of A_2.

In this phase, they talked with their teammate. If Neary exclusively distinguished the two groups of A_1 and A_2, and B_1 and B_2, we can regard Neary worked correctly.

This table shows that Neary approximately made correct outputs. For example, Neary detected 18 times that B_1 is estimated to be the conversation partner of B_2. Out of the 18, Neary estimated three times not only B_1 but also A_1 as the partners of B_2, and one time that A_2 was also estimated as the partner.

These interfusion were caused by various reasons. A loud voice caused everyone to be mistakenly put in the same group. When the both members of a team were silent, the other team member's voices caused confusion of the two groups. A_1's voice was comparatively loud so that Neary sometimes estimated all four members in the same conversation group. We, however, do not regard the estimation was absolute misjudgment because the such estimation fitted with our impression in the situation.

Negotiation phase

We considered this phase during their negotiations. Table 2 shows samples of conversations and Neary's output when A_2 was negotiating with the group B members.

When A_2 negotiated with group B, Neary detected that A_1 is not in the conversation (Fig. 10). When A_1 joined the conversation by a loud voice from behind the partition, Neary estimated all the four members in the same group (Fig. 11).

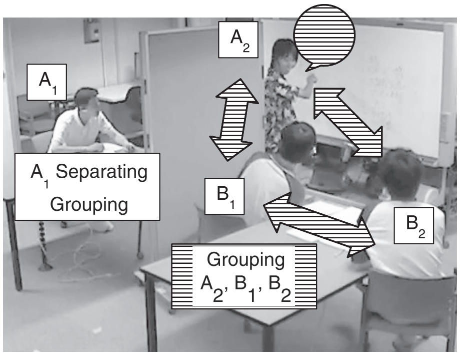

Fig. 10 Neary detected a conversation group organized with $A_2, B_1, and B_2$.

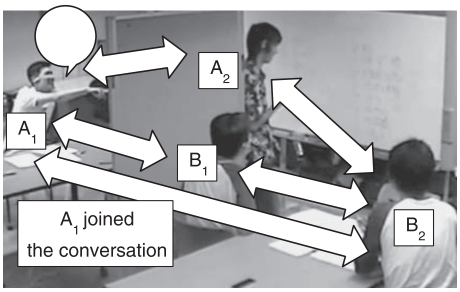

Fig. 11 Neary detected A_1 joined the conversation.

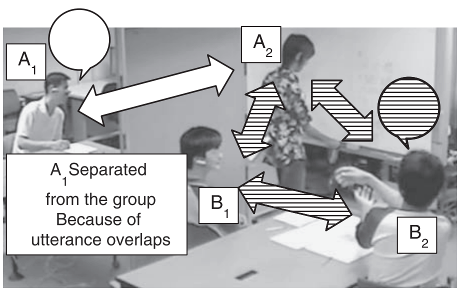

Fig. 12 Neary separated A_1 from the conversation group because of utterance overlap.

An interesting case we observed is that Neary estimated A_1 is out of the conversation group although A_1 joined the conversation by a loud voice from behind the partition (Fig. 12).

This is because A_1 and B_2's utterances overlapped each other. A member's utterance had the strongest influence on his own microphone. When members simultaneously talked, their microphones received completely different sounds, which caused the estimation that they were divided into other groups.

In this case, B_1 and the member listening to B_2's utterance did not think that A_1 was in the same conversation group, the estimation is regarded as a correct recognition.

5. Application Examples: Detecting Group Context on Photochat

This section presents our attempt to deploy Neary for context-awareness of a ubiquitous computing application. Here we introduce a system called PhotoChat, a tool to facilitate communications among nomadic users in social events like touring in museums, attending academic conferences, visiting trade shows, etc. By using PhotoChat, the users can intuitively and quickly exchange their interests and experiential knowledge by sharing photos and hand-written notes [11].

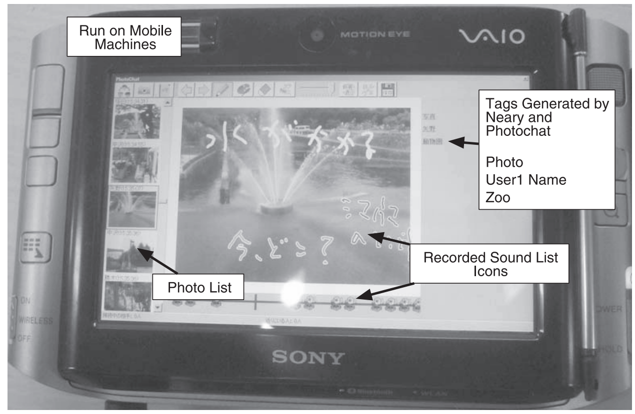

Fig. 13 Appearance of the PhotoChat system with Neary.

PhotoChat simply distributes all photos and written notes to all users' devices as soon as possible. This enables the users to easily notice other users' activities and interests. However, when the numbers of users increases and they disperse, sharing of all photos and notes is not always useful. When some members form a group for a certain activity (e.g., discussing on an exhibit), it might be annoying that many of no related photos from other groups interrupt their PhotoChat timelines. Also, photos and notes derived from a private chat by a small part of members should not be distributed to all users. In this section, we verify Neary enables to keep up with the changes of grouping of the PhotoChat users according to their conversational activities.

Figure 13 shows an appearance of the PhotoChat device. The main area of the screen is used for browsing photos, writing notes on those as well as camera finder. The left side of the screen shows chronological list of all photos including the user's self and other users'. Neary continuously estimates whom the user forms a conversational group with and associates the conversation partner names with the photos.

5.1 Deployment for Experience Sharing in the Zoo

We deployed PhotoChat with Neary for facilitating experience sharing. The eight participants used PhotoChat during their tour in the zoo (Fig. 14). They freely walked around the zoo, taking photos, and speaking with other participants. They were not asked to tour in a group.

When we want to share photos taken at such social events like sightseeing and touring, we usually use Web sites specialized for sharing photos, e.g., Flickr, Picasa, etc. In such sites, we need to classify our photos into "photos

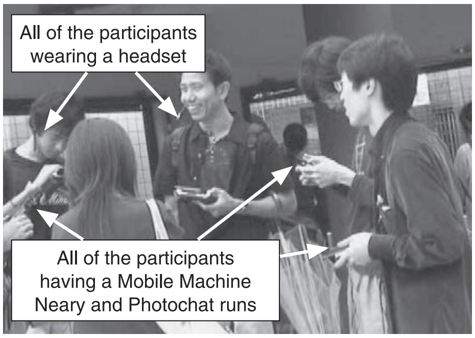

Fig. 14 Using PhotoChat during touring in the zoo.

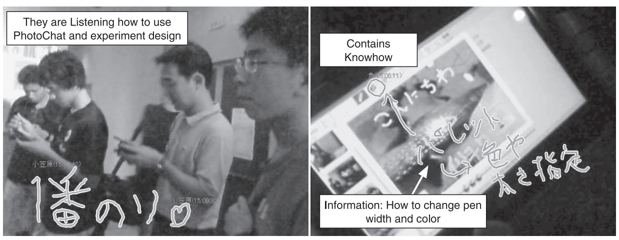

Fig. 15 Photos determined to be shared by many users.

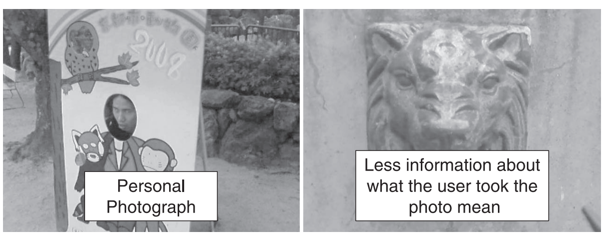

Fig. 16 Photos determined to be personal or for small group of members.

for public", "photos for friends", "photos for family", etc., which is hard task, especially remembering who is associated with which photos. Neary suggests to classify photos according to ad-hoc groups situated in conversations and helps us to appropriately distribute the photos to group members as soon as we take them.

In the following examples of photos taken during the trial, we consider how Neary suggests them to be distributed in the group members.

Figure 15 shows the photos that Neary suggests them to be shared by almost all users: Neary determined almost all users were in the same conversational field when the photos were taken. These photos were taken when we gave the participants a briefing before starting the tour. We consider photos taken in such situations to have know-how and then they should be shared by all members.

Figure 16 shows the photos that Neary suggests them to be shared by a few users. They were personal photos taken while chatting by ad-hoc and small group members. Such photos should be shared by the members who participated in the ad-hoc events with understanding the contexts. Neary could successfully suggest to PhotoChat who were participating in the ad-hoc events when the photos were taken.

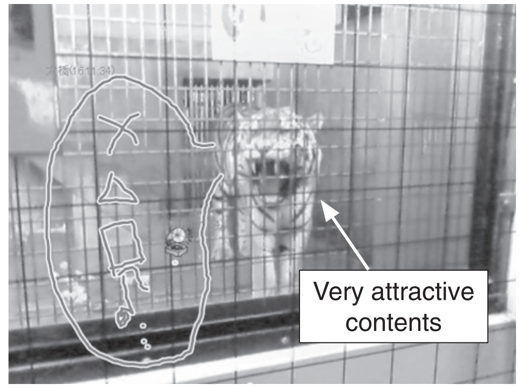

Fig. 17 Example determined to be attractive photo since the users stayed a long time at the scene.

Figure 17 is an example photo showing an event attracting some members for long time (a loudly howling tiger). In such case, Neary detected the members who were there as long as they stayed in the situation. Resultly, the members moving in and out around the scene were naturally associated with the photo and we can track the group dynamics of members' participating to the event by the Neary's log.

5.2 Deployment for Experience Sharing in a Conference

We deployed PhotoChat with Neary for experience sharing in an academic conference with eight participants(A~H) who freely used PhotoChat. After the trial finished, sharing photos by using Neary output.

This conference consists of lecture-style presentations and interactive demo sessions. In presentation sessions, A~H are listening presenter's speech as audience. In this case, all the participants are unified in one conversational field. If participants take pictures, these pictures are suppose to be information of presentations, which are appropriate to be shared by all participants. Neary successfully determined this case.

In demo sessions, participants individually visited demo areas. When they took pictures during been classified in a conversational field, these pictures were supposed to be referred in the conversational filed, and they should be shared by the members in the conversation. When users take photos without participating in any conversational fields, the photos are supposed to be caused by her/his own interest. Neary successfully suggested PhotoChat not to share these photos.

Figure 18 shows photos taken in the first 20 minutes in this trial. These photos are chronologically laied out in the figure. Photos shared by some members is taking place in the same color edge. For example, the blue edge (*1) means the photo was shared by User C, E, F and G.

In the first ten minutes, a lecture-style presentation was made. Neary classified almost all time to one conversational field throughout the hall. Participants took photos of the presentation screen (*2 in Fig. 18) whenever they wanted.

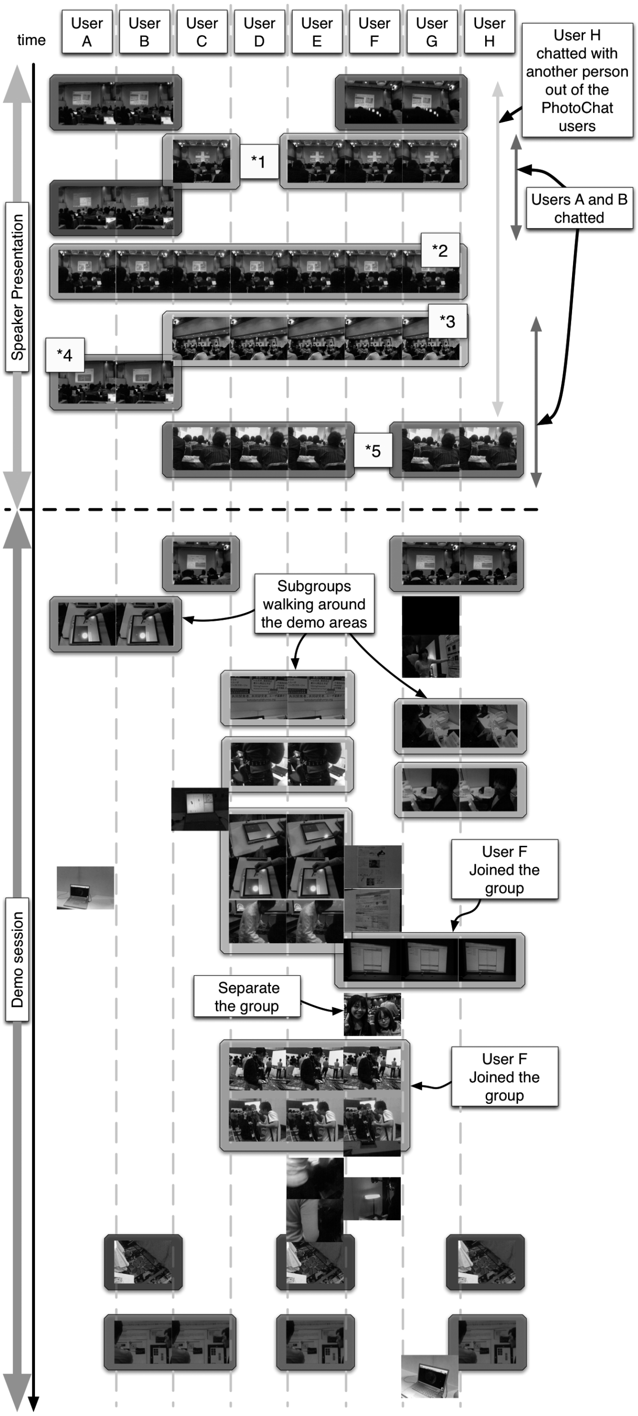

Fig. 18 Photos taken at an academic conference.

In the presentation session, participants were sometime divided into two groups (A and B; C∼H): They were sitting close individually within the same group members. They sometime whispered in the groups. Neary detected these small conversation groups separately associated with *3 and *4 in Fig. 18.

Participants A and B were close friends. They were talking about the presentation and discussed the order to visit demonstrations. Photos they took were related to their private information. Neary delivered these photos only to A and B as we intended.

Participants C∼G were quietly listening the presentation. They took photos of presentation screens and wrote the summary of presentations. Neary suggested these photos to be shared by all audience as we intended. However, some participants did not receive these photos (*1 and *5 in Fig. 18). This was because they made noise, cough and the sound of writing memos. These noise harmed Neary's detection. This is a future issue to be solved.

User H did not attend to the presentation and took no photo. He chatted with another person out of the group during almost of the presentation session. Just before starting the demo session, he joined to the presentation and attended to the demo session with other PhotoChat users. Neary successfully tracked his context.

6. Conclusion

In this paper, we proposed a system called Neary that detects conversational fields based on similarity of auditory situation among users. We presented its implementation and experimental evaluation. The experiment showed that Neary could be adjustable to detect conversational fields which matched with user's impressions in their experiences. We also presented two examples of Neary deployment to track user contexts during experience sharing in touring at the zoo and attending an academic conference. The examples showed PhotoChat successfully delivered photos to appropriate members according to continually changing conversation groups tracked by Neary.

Neary sometimes makes misjudges if utterances overlap. This problem may be solved by calculating frequency information not as one huge task but as several small, separated tasks. This is a future work.

References

[1] P.M. Aoki, M. Romaine, M.H. Szymanski, J.D. Thornton, D. Wilson, and A. Woodruff, "The Mad Hatter's cocktail party: A social mobile audio space supporting multiple simultaneous conversations," Proc. CHI 2003, pp.425–432, 2003.
[2] R. Borovoy, F. Martin, S. Vemuri, M. Resnick, B. Silverman, and C. Hancock, "Meme tags and community mirrors: Moving from conferences to collaboration," Proc. CSCW' 98, pp.159–168, 1998.
[3] T. Choudhury, Sensing and Modeling Human Networks, Doctoral Thesis, Massachusetts Institute of Technology, Sept. 2003.
[4] E.T. Hall, The Hidden Dimension, Doubleday & Company, 1966.
[5] J. Hong, G. Borriello, J. Landay, D. McDonald, B. Schilit, and D. Tygar, "Privacy and security in the location-enhanced World Wide Web," Proc. Ubicomp 2003, Oct. 2003.
[6] S.S. Intille, J.W. Davis, and A.F. Bobick, "Real-time closed world tracking," 1997 IEEE Conference on Computer Vision and Pattern Recognition, pp.697–703, 1997.
[7] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," Computer Vision and Image Understanding, vol.80, pp.42–56, 2000.
[8] T. Nakakura, Y. Sumi, and T. Nishida, "Neary: Conversation field detection based on similarity of auditory situation," 10th Workshop

on Mobile Computing Systems and Applications (HotMobile 2009), 2009.

[9] Y. Nakamura, Y. Namimatsu, N. Miyazaki, Y. Matsuo, and T. Nishimura, "A method for estimating position and orientation with a topological approach using multiple infrared tags," Proc. International Conference on Networked Sensing Systems (INSS2007), pp.187–195, 2007.

[10] J. Rekimoto, T. Miyaki, and T. Ishizawa, "LifeTag: WiFi-based continuous location logging for life pattern analysis," Proc. 3rd International Symposium on Location and Context-Awareness (LOCA2007), pp.35–49, 2007.

[11] Y. Sumi, J. Ito, and T. Nishida, "PhotoChat: Communication support system based on sharing photos and notes," CHI 2008 Extended Abstracts, pp.3237–3242, April 2008.

[12] M. Wirz, D. Roggen, and G. Tröster, "A wearable, ambient sound-based approach for infrastructureless fuzzy proximity estimation," 14th IEEE International Symposium on Wearable Computers (ISWC 2010), pp.125–128, 2010.

[13] D. Wyatt, T. Choudhury, J. Bilmes, and J. Kitts, "Towards automated social analysis of situated speech data," Proc. Ubicomp 2008, pp.168–171, Sept. 2008.

[14] H. Yoshida, S. Ito, and N. Kawaguchi, "Evaluation of pre-acquisition methods for position estimation system using wireless LAN," 3rd International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2006), pp.148–155, 2006.

Toshiya Nakakura is an engineer in NTT Communications. He received his B.Eng. and M.Eng. from Kyoto University in 2007 and 2009.

Yasuyuki Sumi is a professor in School of Systems Information Science at Future University-Hakodate since 2011. He received his B.Eng. from Waseda University in 1990, and his M.Eng. and D.Eng. degrees in Information Engineering from the University of Tokyo in 1992 and 1995. He was a researcher at ATR from 1995 to 2003, and an associate professor in Graduate School of Informatics at Kyoto University from 2003 to 2011. His Research interests include knowledge-based systems, creativity supporting systems, interface/social agents, ubiquitous/wearable computing, Web intelligence, multimedia processing, and their applications for facilitating human interaction and collaborative experiences.

Toyoaki Nishida received his Ph.D. in Informations Science from Kyoto University in 1984. He is currently a Professor at the Department of Intelligence Science and Technology, Kyoto University, Kyoto, Japan. His research centers on artificial intelligence and human computer interaction. His current research focuses on social intelligence design and communicative intelligence. He led several projects, including a JSPS (Japan Society for the Promotion of Science) project on intelligent media technology for supporting natural communication between people, and a RISTEX (The Research Institute of Science and Technology for Society) project on the conversational knowledge process for risk communication.