Neary: Conversation field detection based on similarity of auditory situation

Toshiya Nakakura
Graduate School of Informatics,
Kyoto University
nakakura@ii.ist.i.kyoto-u.ac.jp

Yasuyuki Sumi
Graduate School of Informatics,
Kyoto University
sumi@i.kyoto-u.ac.jp

Toyoaki Nishida
Graduate School of Informatics,
Kyoto University
nishida@i.kyoto-u.ac.jp

ABSTRACT

This paper proposes a system called Neary that detects conversational fields and is designed to run on mobile machines with a network module and a microphone. Neary operates on a simple algorithm and a light-weight process, so other systems can incorporate it smoothly. In this paper, we show an implementation of our Neary system and its experimental evaluations. Neary can distinguish between two conversation groups one meter apart and dynamically detect the changes in the number and the physical size of conversation fields.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Synchronous interaction, Sound and Music Computing; J.4 [**Social and Behavioral Sciences**]: Sociology

General Terms

Measurement and Design

Keywords

User Context, conversation field detect

1. INTRODUCTION

This paper proposes a system called Neary that detects a conversational field, which is defined in this paper as a physical area where multiple persons join a conversation. Neary detects a conversational field by inferring the members who have joined the conversation. Neary excludes someone may join the conversation; however, when s/he joins, Neary infers his/her participation in the same conversational field.

Neary is designed to be adjustable to transform the conversational field. We achieve this by a simple algorithm that does not depend on the form of the conversational field or the distance between its users, but only uses the similarity of the sound environment.

We previously examined the precision of Neary in meeting environments in its developmental stage. It proved that Neary has high precision, low recall, and is adjustable to transform the conversation field, if there are sufficient spaces. In this paper we verify that Neary is adjustable to transform the conversation field in small spaces.

The many researches that have attempted to record and analyze various kinds of human communication can be divided into two major methods. One is based on an analysis of the situation around the conversation [1] [6], and the other is based on a spectrum analysis of utterances [2].

As for the former type, for example, some researches judge user contexts from position and orientation by estimating an object's position and orientation based on topological information collected using infrared tags [5], WiFi accesspoints [4], [10], [7], etc. Although these methods are adaptive for various purposes, it remains unclear whether they are actually conversations. In addition, the method can hardly recognize how the conversation field is distributed. Neary uses sound information to avoid these weak points.

As for the latter type, for example, even though much research has attempted to recognize conversations using sound information, such studies often assume that conversations are generally performed by those nearby. Of course, many people communicate based on social distances of 120 ∼ or 210 cm [3]; however we believe systems that cling to this view may overlook the following communication forms. Many conversation fields can be found in party and exhibition halls. On the other hand, it is not natural to assume that one conversation field can cover an entire lecture hall It is hard to distinguish these two situations by differences of distance and the number of people. Neary is designed to identify these situations that existing methods can hardly distinguish.

Our approach using sound environments resembles work by Choudhury et. al.[2][9]. However, since their motivation is based on social network analysis, their systems have high precision and researchers receive output after they have finished their experiments. Our motivation is to support other systems, so Neary's performance is light enough for ordinary mobile PCs and outputs conversational situations immediately.

This paper shows Neary's basic idea, its implementation, and the important aspects of the experiment results.

Hotmobile 2009, February 23-24, 2009, Santa Cruz, CA, USA.
Copyright 2009 ACM 978-1-60558-283-2/09/02 ...$5.00.

2. DETECTING CONVERSATION FIELDS

Neary infers the existence of conversation fields using the similarity of sound environments. Its algorithm is based on a simple idea: in the same conversation field, the same sound is heard. For example, consider the two conversation fields shown in Figure 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 record A's voice louder than D's. On the other hand, the microphones of C and D record D's voice louder than A's.

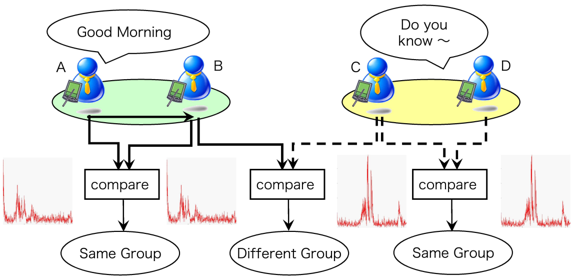

Figure 1: Basic idea of Neary algorithm

If Neary compares their microphones' captured sounds, A's is more similar to B's than C's and D's. By comparing sound input, Neary can recognize the conversation field form.

This algorithm divides the users into groups by specific sounds. If there is one loud sound in the room, whether the room is large or small, the users are unified. If there are some loud sounds, the users are divided into groups. Due to this property, Neary adapts itself to various situations, including presentations, parties, etc.

We tested this algorithm in a meeting environment where a presenter and the audience used Neary. In this test, the precision of Neary was 96.6%(917/949) and its recall was 67.9% (917/1349).

If there are many conversation fields in a small room, some conversation fields may merge, or someone may leave the conversation. Robustness to these changes is desirable. In this paper, we verified whether Neary successfully judged such situations.

3. IMPLEMENTATION OF NEARY

3.1 Basic architecture

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 send necessary information for detecting conversation fields. 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. This time we used the 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 communication. Figure 2 shows Neary's users and devices, and Figure 3 shows its system chart and data flow.

Figure 2: User and device

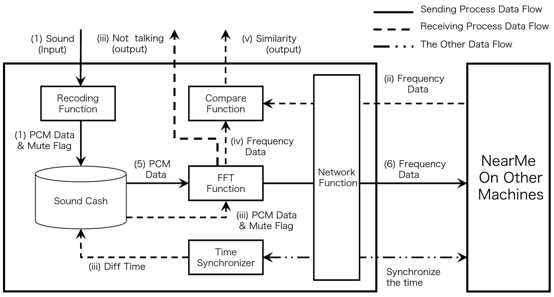

Figure 3: Neary's system chart and data flow

3.2 Detection algorithm

Neary's algorithm compares the frequency of input sound. If the input sound is silent, clearly nobody is talking, so Neary assumes there is no conversation field without calculating. If the input volume is smaller than the average input volume (Figure 4), Neary regards the input sound as silence. The detailed algorithm is as follows:

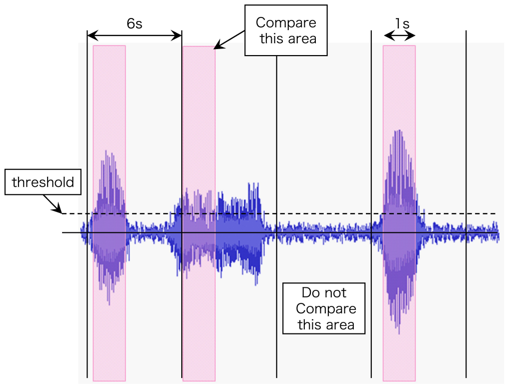

Figure 4: Sound input and area compared by Neary

Sending process

1. Create 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 Fast Fourier Transform, and put a timestamp and user name on the output.
6. Send the output to all 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 Fast Fourier Transform, and calculate the cosine similarity of these two bits of data.
If the value is over a threshold, Neary outputs a true value and otherwise outputs a false value.

v. Go to i.

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

Neary uses not only human voices but also other sound information such as music. If persons are listening to the same music and seem to begin to talk, Neary regards them as being in the same conversation field. Neary uses 50 $\sim$ 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. EXPERIMENT

If there are many people in a room, various forms of conversation fields may be found. For example, the number of conversation fields may change, or some people may move to another conversation field. Since Neary's main purpose to keep up with these changes, we examined whether Neary can detect them by designing a task for a conversation 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 Figure 5. 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.

First, each team discussed separately what equipment they needed (Figure 6-1). To avoid conversation between teams, we put a 120-cm wide, 5-cm deep, and 200-cm high partition between the groups and determined before this experiment that this partition did not affect Neary's decision. There were two conversation 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 (Figure 6-2) 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 conversation fields changed and to estimate Neary's accuracy.

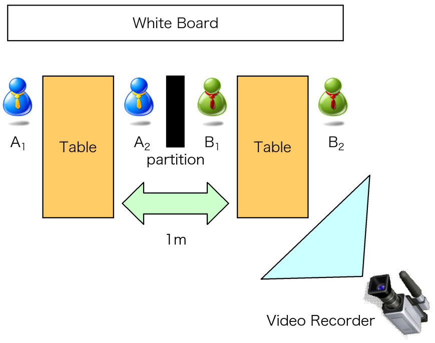

Figure 5: Experiment setting

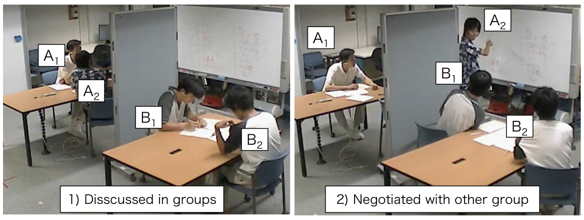

Figure 6: Experiment state

4.2 Results and discussion

Two group conversations

This phase featured two conversation fields, as shown in Figure 6-1. Neary's goal was to divide users into groups A and B(Figure 7).

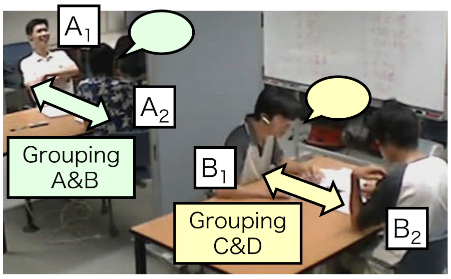

Figure 7: Two group conversations and Neary output

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

Table 1: Neary output when participants harmonized group opinion

Terminal	<inferred communication partners: frequency>
A_1	**<A_2:11>** <B_1:0> <B_2:0> <A_2,B_1:0> <A_2,B_2:3> <B_1,B_2:0> <A_2,B_1,B_2:1>
A_2	**<A_1:21>** <B_1:0> <B_2:0> <A_1,B_1:0> <A_1,B_2:1> <B_1,B_2:0> <A_1,B_1,B_2:1>
B_1	<A_1:0> <A_2:0> **<B_2:19>** <A_1,A_2:2> <A_1,B_2:0> <A_2,B_2:0> <A_1,A_2,B_2:2>
B_2	<A_1:0> <A_2:0> **<B_1:14>** <A_1,A_2:0> <A_1,B_1:3> <A_2,B_1:1> <A_1,A_2,B_1:0>

Table 1 summarizes the frequency of the combination partners inferred by Neary and displayed on each user's screen during the session. For example, <A_2, B_2:3> in the second row indicates that there were three times when Neary guessed A_2 and B_2 as conversational partners of A_1.

In this phase, they talked with their teammate. If Neary detected A_1 and A_2, and B_1 and B_2 in two groups, the output is true.

This table shows that Neary approximately made correct outputs. For example, Neary placed B_2 and B_1 in the same group 14 times. Neary also made three mistakes with B_2, A_1, and B_1 and one with B_2, A_2, and B_1. These mistakes were caused by various reasons. A loud voice caused everyone to be mistakenly put in the same group. When all members of a team were silent, the other team member's voices were deemed mistaken output. A_1's voice was too loud for Neary to often guess that A_1 and someone were in the same group. If someone spoke loudly, he may be speaking to everyone. Inferring intention only by the volume of voice is difficult. We believe that estimating by context is the only way to infer correctly.

Negotiation phase

We considered this phase during their negotiations. Table 2 shows samples of conversations and Neary output when A_2 was negotiating with group B. Neary almost detected the group that continued to talk at this table.

When A_2 negotiated with group B, Neary detected that A_1 is not in the conversation (Figure 8). When A_1 joined the conversation by a loud voice from behind the partition, Neary detected the group as $A_1, A_2, B_1, and B_2$(Figure 9).

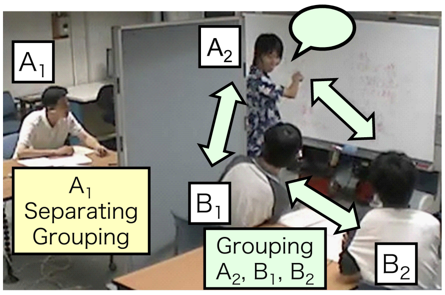

Figure 8: Neary detected three people's conversation

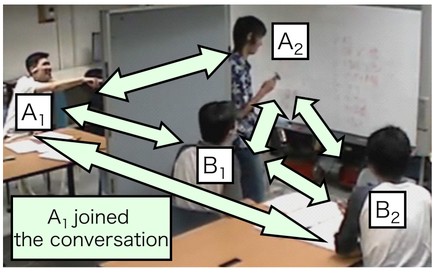

Figure 9: Neary detected A_1 join the conversation

We got an interesting case. Although A_1 joined the conversation by a loud voice from behind the partition, Neary detected the group as A_2, B_1, and B_2(Figure 10).

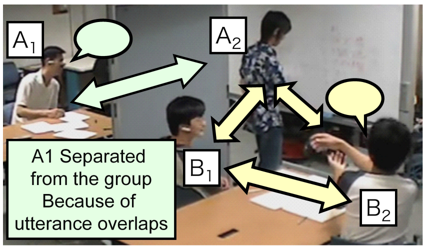

Figure 10: Neary separated A_1 from the conversation because of utterance overlaps

This is because A_1 and B_1's utterances overlapped each other. A member's utterance had the strongest influence on her own microphone. If members simultaneously talked, their microphones received completely different sounds. Because Neary compared the sounds, they were divided into other groups.

Table 2: Display on each user's Neary terminal during sample session

Conversation		display on A_1's	display on A_2's	display on B_1's	display on B_2's
	A_2 is negotiating with B_1 and B_2. A_1 is apart from them.				
$A_2 \rightarrow B$	"First, we have two large desks. That's unfair."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_1 \rightarrow A_2$	"True, but..."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \rightarrow A_2$	"No, no. We don't need such desks."	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_1 \rightarrow A_2$	"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	..	..	..	..
$A_2 \rightarrow B$	"Then you should take this large-sized display."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \rightarrow A_2$	"No. We already have the same display."	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_2 \rightarrow B_2$	"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
$A_2 \rightarrow B$	"We don't need it!"	..	B_1, B_2	A_2, B_2	A_2, B_1
$A_2 \rightarrow B$	"We'll take this display, and you take this desk."	..	B_1, B_2	A_2, B_2	A_2, B_1
$B_2 \rightarrow A_2$	"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.				
$A_1 \rightarrow B$	"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
$A_2 \rightarrow A_1$	"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
$B_2 \rightarrow A_2$	"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
	...				

In this case, B_1 and the member listening to B_1's utterance did not think that A_1 was in the same conversation group, so we believe it is a collect recognition. However, if the utterances of multiple members happen to overlap, this property may cause mistakes in Neary.

This problem cannot be solved only by using sound. One solution is removing a short overlap as a noise by estimating the tempord context. In one conversation, it seems some utterances kept overlapping for a long time, so outputs must be merged before they can be recovered.

5. APPLICATION EXAMPLE

This system works on a machine with a microphone and a network function. PhotoChat [8] uses Neary. Figure 11 shows samples of PhotoChat and Neary output tags.

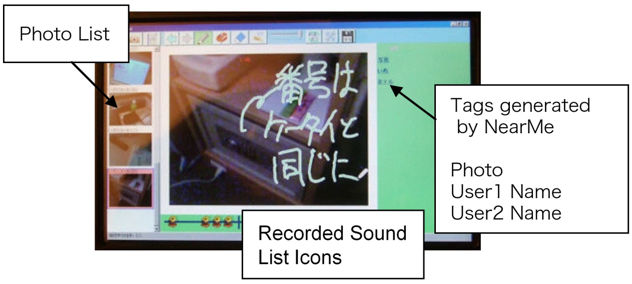

Figure 11: Tags generated by Neary on PhotoChat interface

PhotoChat facilitates communication among users who want to share experiences by enabling them to exchange photos and notes. When the amount of photographs increases, PhotoChat suffers from poor photo retrieval. If they only use the default tags (date and time) provided by PhotoChat, users often have difficulty finding photos. Neary relieves this problem by automatically attaching the names of other users who talk about photos as tags. These tags make searching more instinctive. Neary is more suitable for such use than ordinary proximity sensors, because people a user talked to before and after taking photos have more powerful connections with the photos than people who are just near by.

Neary also helps analyze the logs. We performed an PhotoChat experiment in a zoo (Figure 12) with eight participants who freely used PhotoChat.

Users called each other in front of various animals using PhotoChat and went their own ways. Analysts can't understand how users communicate with each other. Neary shows how PhotoChat logs encourage communication among users and what kind of conversation encourages users to take photos.

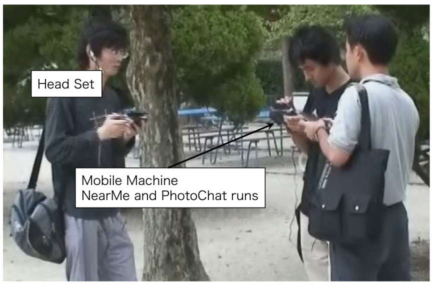

Figure 12: PhotoChat at a zoo

Neary may help users share photos and other data. If they want to share photos, they usually go home and use social media sites such as Flickr. They can categorize their photos into "Photos for Public," "Photos for Friends," "Photos for Family," etc. This series of tasks is very hard, especially remembering who is concerned with which photos. Neary can categorize photos based on conversations and help other systems share photos as soon as users take them.

We classified photos by batch processing after the zoo experiment. The following are some examples.

First, Figure 13 shows the photos shared by almost all users. They were taken when we explained our experimental configuration and how to use PhotoChat. We consider photos taken in such situations to have know-how and to be shared.

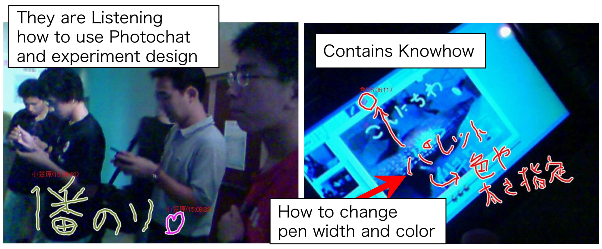
They are Listening how to use Photochat and experiment design
1番のり
Contains Knowhow
パレット
色や太さ指定
How to change pen width and color

Figure 13: Neary shares this photograph between many people.

Figure 14 shows the photos shared by a few users. They consist of personal photos and photos with less information about what the photographer means. They should share these photos with the users who talked with the photographers of these photos. Using Neary output, we can control this.

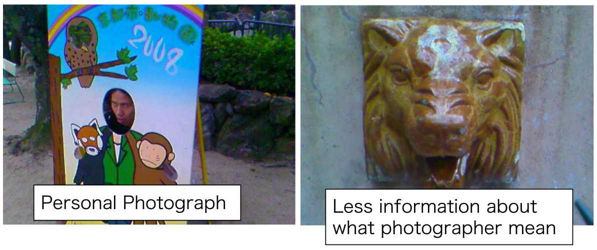

Figure 14: Photos should not be shared

Users talked the longest after taking photos (Figure 15) because the subject is very attractive. This photo suggests that we can guess the photo's priority and its property.

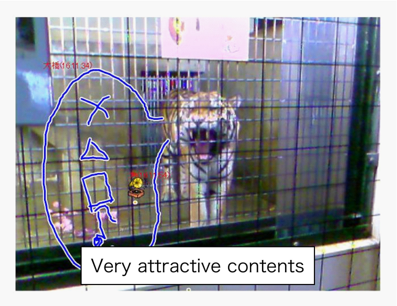

Figure 15: One photo about which users talked a long time

For example, the photos of a presenter and a screen in a workshop may be useful for discussion about the presentation. Such photos should be shared by all the members in the hall. After the presentation, some may take personal photos. These photos should only be shared by the members concerned. In these cases, Neary has an advantage over other proximity sensors too. Since Neary is not based on distance, it can follow conversation situations even if the conversation field becomes very large. Neary supports these requests.

6. CONCLUSION

In this paper, we showed an implementation of our Neary system and experimental evaluations.

Neary can correctly divide members into conversation fields and adjust the merger and division of conversations. However, Neary sometimes makes mistakes if utterances overlap. This problem may be solved by calculating frequency information not as one huge task but as several small, separate tasks. This is future work.

7. REFERENCES

[1] 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 CSCW'98*, pages 159–168. ACM, 1998.

[2] T. Choudhury. Sensing and Modeling Human Networks. *Doctoral thesis, Massachusetts Institute of Technology*, September 2003.

[3] E. T. Hall. *The Hidden Dimension*. Doubleday & Company, Inc., 1966.

[4] J. Hong, G. Borriello, J. Landay, D. McDonald, B. Schilit, and D. Tygar. Privacy and Security in the Location-enhanced World Wide Web. *In Proceedings of Ubicomp 2003*, October 2003.

[5] 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. *In Proc of International Conference on Networked Sensing Systems (INSS2007)*, pages pp.187–195, 2007.

[6] Y. Nakanishi, T. Tsuji, M. Ohyama, and K. Hakozaki. Context Aware Messaging Service: A Dynamical Messaging Delivery using Location Information and Schedule Information. *In Proceedings of Ubicomp 2003*, Vol.4:pp.221–224, August 2000.

[7] J. Rekimoto, T. Miyaki, and T. Ishizawa. LifeTag: WiFi-based Continuous Location Logging for Life Pattern Analysis. *3rd International Symposium on Location- and Context-Awareness (LOCA2007)*, pages pp.35–49, 2007.

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

[9] D. Wyatt, T. Choudhury, J. Bilmes, and J. Kitts. Towards Automated Social Analysis of Situated Speech Data. *Proceedings of Ubicomp 2008*, Spetember 2008.

[10] H. Yoshida, S. Ito, and N. Kawaguchi. Evaluation of Pre-Acquisition Methods for Position Estimation System using Wireless LAN. *The Third International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2006)*, pages pp.148–155, 2006.