SocioPhone: Everyday Face-to-Face Interaction Monitoring Platform Using Multi-Phone Sensor Fusion

Youngki Lee[1], Chulhong Min[2], Chanyou Hwang[2], Jaeung Lee[3], Inseok Hwang[2,4], Younghyun Ju[2], Chungkuk Yoo[2], Miri Moon[3], Uichin Lee[5], Junehwa Song[2]

[1]School of Information Systems, Singapore Management University
[2]Computer Science Department, KAIST, [3]Web Science and Technology Division, KAIST,
[4]Center for Mobile Software Platform, KAIST, [5]Knowledge Service Engineering Department, KAIST

[1]youngkilee@smu.edu.sg,
{chulhong, chanyou, leejai, inseok, yhju, ckyoo, miri.moon, junesong}@nclab.kaist.ac.kr, [5]uclee@kaist.edu

ABSTRACT

In this paper, we propose SocioPhone, a novel initiative to build a mobile platform for face-to-face interaction monitoring. Face-to-face interaction, especially conversation, is a fundamental part of everyday life. *Interaction-aware applications* aimed at facilitating group conversations have been proposed, but have not proliferated yet. Useful contexts to capture and support face-to-face interactions need to be explored more deeply. More important, recognizing delicate conversational contexts with commodity mobile devices requires solving a number of technical challenges. As a first step to address such challenges, we identify useful *meta-linguistic contexts* of conversation, such as turn-takings, prosodic features, a dominant participant, and pace. These serve as cornerstones for building a variety of interaction-aware applications. SocioPhone abstracts such useful meta-linguistic contexts as a set of intuitive APIs. Its runtime efficiently monitors registered contexts during in-progress conversations and notifies applications on-the-fly. Importantly, we have noticed that *online turn monitoring* is the basic building block for extracting diverse meta-linguistic contexts, and have devised a novel *volume-topography*-based method. We show the usefulness of SocioPhone with several interesting applications: *SocioTherapist*, *SocioDigest*, and *Tug-of-War*. Also, we show that our turn-monitoring technique is highly accurate and energy-efficient under diverse real-life situations.

Categories and Subject Descriptors

K.8 [**Personal Computing**]: General; C.3 [**Special-Purpose and Application-based Systems**]: Real-time and embedded systems

Keywords

Interaction, Conversation, Social, Platform, Volume Topography

1. INTRODUCTION

Face-to-face social interaction is an integral part of human life; everyday, people dine with family, have meetings with colleagues, and spend time with friends. A promising new direction for mobile sensing lies in capturing and utilizing sophisticated social contexts during daily face-to-face interactions. Early *interaction-aware applications* have been emerging and show its potential usefulness [22][27]. For example, *MeetingMediator* [22] displays the skew of individuals' verbal participations to promote group brainstorming. Another application helps a user remember the name of the person he is talking with, to help avoid the awkward experience of forgetting a name [27]. However, building interaction-aware applications involves severe challenges without system-level support. First of all, such applications are still in an early stage and most developers do not know which contexts to leverage during daily conversations. Furthermore, monitoring conversations requires implementing complicated inference logics, repetitive learning and testing to improve recognition accuracy, and significant optimization of battery use.

In this paper, we propose *SocioPhone*, a mobile platform for face-to-face interaction monitoring. Ideally, a full-fledged interaction monitoring platform would capture a variety of communicative cues expressed during face-to-face interaction such as verbal cues (spoken words and sentences), aural cues (tones, pitch), and visual cues (gesture, eye contact). As a first step, SocioPhone focuses on monitoring *meta-linguistic contexts* that provide useful information about conversations without requiring computation-intensive semantic inference on conversation contents. SocioPhone provides applications with a set of intuitive APIs to monitor rich meta-linguistic contexts on the fly (See Section 2); applications can submit simple monitoring requests to obtain contexts of interests. The SocioPhone runtime monitors registered contexts in a highly-efficient and precise manner, based on our new *volume-topography-based turn monitoring* technique (See Section 4).

In its core, SocioPhone monitors *conversational turns,* the basic unit of conversation; a *turn* is a continuous speech segment where a person starts and ends her speech [3][10]. We have noticed that monitoring turns is a first crucial step to deriving many interesting aspects of a conversion, e.g., how long and often one talks, how quickly she responds, who talks more or less, and how fast a conversation progresses. More interestingly, turn analysis enables high-level social inference, such as one's role in a conversation and problematic situations [9][16][35]. Future mobile applications will be tightly interwoven with sophisticated interactions, e.g., dynamic conversational flows and relational behaviors, in-situ; this will enrich and broaden the set of potential applications, from interaction facilitations to collaborative decision making, and even to psychological care. In a broader view, monitoring turns can also serve as the prerequisite for speaker-specific vocal inference and content analysis in real-time, such as assessing a speaker's emotional state and performing deep semantic analysis.

Online turn monitoring is a primitive building block, but it is challenging to implement it on everyday personal mobile devices. Existing voice recognition techniques such as speaker recognition

[1] This work was done while this author was at KAIST.

MobiSys'13, June 25–28, 2013, Taipei, Taiwan.

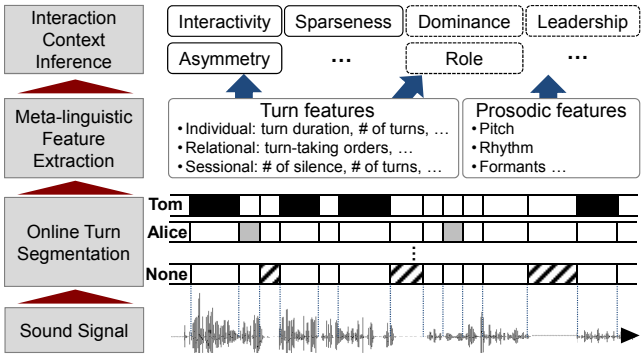


Figure 1. Online turn segmentation and meta-linguistic conversation monitoring

[7] and speaker diarization [2][4] rarely consider the challenges of mobile environments, e.g., unconstrained acoustic situations, real-time monitoring, and battery limitations. A potential approach to turn monitoring would be to continuously execute crafted speaker recognition logic, as in SpeakerSense [27] (See Section 3.1). However, this has a number of shortcomings. First, short-lasting turns (1-2 seconds) are common in casual conversations [2], but cannot be detected reliably. Existing techniques mostly require long speech segments (e.g., 3-8 seconds.) for reliable recognition to ensure statistical confidence of the windowed voice samples with respect to the speaker-specific pre-constructed spectral model [27][40]. More challenging, daily conversations do not occur in an ideal setting; dynamic ambient noises inevitably distort one's vocal signatures, leading to poor recognition accuracy. Furthermore, running speaker recognition on smartphones consumes significant power, (> 400 mW) for high-rate sound sensing and heavy computation [27][31].

To address the challenges for online turn monitoring, we propose an *on-the-spot multi-phone sensor fusion* approach; multiple smartphones work together to detect turn changes and associated speakers, along with a short in-situ training. Naturally placed phones belonging to conversation group members simultaneously sense a speaker's voice signals, but capture the signals with different strengths depending on their positions. Such relative sensory readings can be fused in realtime to form a *volume topography*, i.e., a signature vector of volume values sensed over different phones. Our key observation is that such a topography is unique to each speaker, showing enough discrimination power to identify turns and associated speakers. With a short training period, e.g., 30-60 seconds at the beginning of a conversation, frequent turn-taking of speakers can be very quickly and precisely traced through simple vector matching.

Our volume-topography-based technique has important advantages for online turn monitoring. First, volume parameters can be instantly and reliably estimated, even with a very short sensing window, e.g., 0.3 seconds; this allows us to monitor dynamic turn-taking behavior in a highly agile way. Second, volume-topography is less susceptible to diverse environmental noises as it is built in-situ to reflect the current noise characteristics. Third, our approach is computationally much lighter than existing techniques [7][27]; it does not require complex signal processing such as MFCC extraction and GMM matching. Finally, we note that the method works well even at very low sampling rates (as low as 500 Hz), which has the potential to reduce users' privacy concerns.

SocioPhone shows the potential to transform a personal mobile device into a social device that is aware of fine-grained face-to-face interaction contexts. So far, a number of mobile sensing systems have been proposed; yet, most of them focus on sensing personal status [25][28][29]. A few systems aim at capturing social contexts to facilitate interaction, but they provide only coarse-grained contexts such as encounters or presence of conversation [11][27].

We now summarize the contribution of this paper. First, we propose SocioPhone, a novel mobile interaction monitoring platform; it provides useful APIs to monitor '*turn*' and turn-derived meta-linguistic contexts. Second, as a key building block, we propose a new online turn-monitoring technique based on the volume topography constructed on the spot by collaborative sound sensing. In addition, we adopt and craft other supporting components to build SocioPhone as a working platform. Third, we prototype three promising applications, *SocioTherapist*, *SocioDigest,* and *Tug-of-War* on SocioPhone, and show their potential use. Finally, through extensive experiments, we show that our technique outperforms the state-of-the-art techniques in terms of accuracy, noise-resiliency, and resource usage.

The rest of the paper is organized as follows. Section 2 motivates face-to-face interaction monitoring and introduces the SocioPhone API and our applications. Section 3 presents the technical challenges of daily conversation and online turn monitoring. Section 4 describes the volume-topography-based technique in detail, and Section 5 presents the platform implementation. In Section 6, we show the effectiveness of our technique, and discuss potential issues in Section 7. We present related work in Section 8, and conclude the paper in Section 9.

2. SOCIOPHONE API And APPLICATIONS

2.1 Meta-Linguistic Interaction Monitoring

Developing a mobile platform to monitor everyday face-to-face interaction opens a broad spectrum of design considerations. First of all, it is important to identify core system requirements for interaction monitoring and abstract them as common interfaces. In addition, we need to devise techniques to support diverse real-life interaction situations that are often disorderly, noisy, and dynamic. Unconstrained mobile environments make it difficult to simply adopt existing technologies that were mostly developed for rather orderly and lab-like environments. Finally, the issues of computation- and energy-efficiency are further intensified in mobile environments.

In this paper, we take a first step toward an *online conversation monitoring platform*; it supports diverse applications with *meta-linguistic conversational contexts* in unconstrained mobile environments. While there has been much work on conversation analysis from various angles [3][10][17][35], it is important to note that they focus on offline analysis of collected records. The challenges of online monitoring have not been thoroughly explored yet. Figure 1 shows the high-level process of meta-linguistic conversation monitoring composed of two layers: online turn segmentation and meta-linguistic context inference.

Online turn segmentation: Online turn segmentation forms a common basis for any conversation-monitoring system. As the core technical effort, we focus on executing online turn segmentation using smartphones. As a conversation progresses, it identifies turns continually; each turn is annotated with a triple, (*speaking person*, *start time*, *end time*).

Table 1. Key APIs of SocioPhone

Monitoring conversation sessions and turns
registerSessionStartListener (callback(Session), conditions) *registerTurnChangeListener* (callback(Turn)) * conditions = TARGET_PERSON | TARGET_PLACE class Session{ /* seeTable 2 */}; class Turn{ /* see Table 3 */};
Monitoring prosodic features & interaction characteristics
enableProsodicFeature (session_id, /* features to enable */) * Feature ={energy_avg, energy_var, pitch_avg, energy_var, …} *getSparsity* (window_time | window_turns) *getInteractivity* (window_time | window_turns) *getAsymmetry* (window_time | window_turns) *registerDominanceListener* (callback(Interactant), Inferrer) *registerLeadershipListener* (callback(Interactant), Inferrer)
Querying interaction history
getOnGoingSessionHistory("SQL_Query_Statement"); *getPastInteractionHistory*("SQL_Query_Statement");

Table 2. Session table

sID	Interactants	start_time	end_time	place	…
1	Sheldon, Leonard	Nov-6 19:20	Nov-6 21:05	Office	…
2	Wife	Nov-6 22:50	Nov-6 23:08	Home	…
…	…	…	…	…	…

Table 3. Turn table

sID	tID	speaker	start_time	end_time	prosodic_ptr	…
1	1	Sheldon	19:20:35	19:20:39	pointers to Prosodic table entries	…
1	2	Myself	19:20:39	19:21:04		…
1	3	NOBODY	19:21:04	19:21:11		…
…	…	…	…	…	…	…

Meta-linguistic conversation monitoring: Based on the online turn segmentation, we also develop a *light-weight meta-linguistic interaction monitor* that tracks non-verbal elements during conversations such as voice tone and speaking style. Such elements are combined with turn information to infer behavioral and relational characteristics of the conversation participants.

To be more specific, the monitor extracts a number of useful *turn features* from identified turns and complementarily *prosodic features* from sound samples to infer high-level interaction contexts. First, turn features are largely classified as those describing individual participants (e.g., speaking length, number of turns, duration statistics), relations among participants (e.g., turn taking orders, pair-wise turn-taking frequencies), and the whole interaction session (e.g., duration of speaking and non-speaking turns). *Prosodic features* are also useful indicators of social behavior [38] and complement the turn features. Example features are pitch, energy, loudness, rhythm, as well as spectral features like formants, bandwidths, spectrum intensity.

When these simple features are combined, high-level interaction contexts can be further inferred, which are essential for delivering rich interaction-aware applications. For example, a fast-paced conversation can be identified from turn durations. Also, the sparseness of a conversation could be measured from the length and the distribution of the non-speaking turns, which an application may correlate with the progress or troublesome status of the on-going interaction. More complicated inference can be performed using the features. For example, one can understand the most (or the least) dominant person, the roles of participants, their role-playing patterns, and emergent leaders (See Section 5).

2.2 SocioPhone API

Table 1 shows the key SocioPhone APIs to facilitate monitoring rich meta-linguistic information in daily face-to-face interactions.

Monitoring sessions and turns: The two key primitives are *registerSessionStartListener*() and *registerTurnChangeListener*(), with which applications can trace conversational sessions and turns on-the-fly. Once the former is registered, SocioPhone notifies applications of the Session structure upon the start/end of a conversation and join/leave of a participant. See Table 2 for the Session structure. Applications may designate people or places of interest with the "CONDITION" clause. Upon notification of a session start, applications can further request turn monitoring with *registerTurnChangeListener*(). Then, SocioPhone provides the Turn information (Table 3) continuously upon each turn-taking event, i.e., alternation of a speaker or occurrence of pause.

Monitoring meta-linguistic interactions: Applications also can retrieve rich prosodic features associated with each turn using *enableProsodicFeatures*(); such features are provided only with explicit requests to save resources. The API currently provides volume, energy, and pitch features with their means and variances. SocioPhone also provides a set of convenient APIs for informative turn features and their patterns. For example, *getSparsity*() returns how far the speaking turns are separated by non-speaking turns. *registerDominanceListener*() encapsulates complex social inference to find someone with dominance over the conversation. Note that applications can replace the built-in inference engine with custom implementations.

Querying interaction history: In addition to real-time monitoring, SocioPhone supports querying the interaction history of a user. Applications can use *getOnGoingSessionHistory*() to query the on-going session, and *getPastInteractionHistory*() to query completed sessions. Example queries are "*How many turns has John taken within last 10 minutes*" and "*Which three friends has John spoken to the most this week?*" SocioPhone provides a conventional SQL interface to support flexible and easy querying of stored *Session* and *Turn* information.

2.3 Example Applications on SocioPhone

To demonstrate the usefulness of SocioPhone and its APIs, we designed and prototyped three interaction-aware applications.

SocioTherapist: Nonverbal social interaction and turn-taking deficits are a specific characteristic of young autistic children [32]. In speech therapy sessions for autistic children, the therapist often employs a stimulus, e.g., a toy, to evoke verbal turn-takings from a child. Upon a successful response, the child is reinforced with small rewards such as verbal encouragement or a snack [24].

SocioTherapist is a smartphone application for children with a mild degree of autism, and is designed to mimic stimuli and reinforcements *in-situ* during daily social interactions. The motivation and design have been largely advised by a local kindergarten in collaboration with us [18]. The symptoms of those mildly autistic children are not so severe to require full-time treatment in a special education facility. Instead, they attend regular kindergartens as well as periodic dedicated sessions with a speech therapist. However, in daily interactions out of the clinic without the therapist's guidance, they often experience difficulties with turn-taking when chatting or playing with other non-autistic children. Delayed or failed turn-taking may discontinue their interactions, or even result in eventual social isolation.

(a) (b) (c)

Figure 2. Daily report by SocioDigest. (a) Cumulative conversation time within the user's social circle (b) Relative per-person talking times in a session (c) Relative number of turns exchanged in a session

We prototyped SocioTherapist on top of SocioPhone APIs; a callback is triggered for every turn-taking event, i.e., when the speaker has been switched. Through the Turn instance, SocioTherapist easily obtains the properties for the newly started turn, e.g., its speaker, the timestamp it started, etc.

To implement a few basic criteria for desirable turn-taking behaviors, we consulted a speech therapist for autistic children. Accordingly, our initial prototype of SocioTherapist looks for *initiations*, *long-lasting turns*, and *rapid responses*. An initiation is a newly begun turn breaking a long silence. A long-lasting turn indicates a completed turn which lasted for a sufficient duration of time, ruling out short utterances like "Wow!" or "I got it." A rapid response is a newly begun turn immediately after another person's turn. When such turn-takings occur, SocioTherapist displays small rewards on the phones, i.e., well-known robotic characters for children gradually upgraded upon desirable turn-takings.

Our pilot deployment was encouraging. A group of three children played together for 15 minutes with SocioTherapist, including one with a mild degree of autism. The deployment was entirely supervised by a child education professional, who acknowledged clearly noticeable increases of utterances from the autistic child in both frequency and duration of turns.

SocioDigest: The ubiquity of mobile sensing allows us to digitally capture and archive what we see and what we do everyday. This is also known as *life-logging* [36]. As highly social beings, we believe that it is a natural expansion of life-logging to archive our fine-grained interactions around our daily social circles.

In this light, we have been developing SocioDigest, an application providing daily report on a user's 24/7 face-to-face conversations. Figure 2(a) shows a daily report for a PhD student, illustrating relative times he talked to his colleagues and family. SocioDigest further reports the detailed anatomy of each conversation session. Figure 2(b) shows the relative total time durations for which each participating person talked in a conversation session. Based on the report, SocioDigest gives the user a small suggestion as well. SocioDigest is implemented with SocioPhone APIs and easily retrieves the turn-wise durations from the timestamp attributes of the Turn instances. Figure 2(c) reports even more details, the turn-taking graph. Each vertex denotes a participant of the conversation, and the edge thickness denotes the numbers of turns exchanged between the pair. A thick edge implies that this person would be the most respondent to me, or I was to him/her as well.

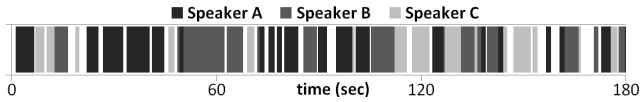


Figure 3. Turn-taking patterns in a sample conversation case

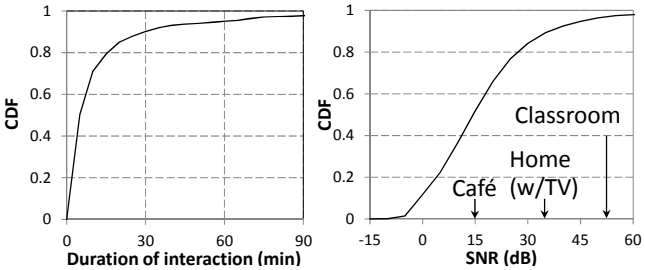


Figure 4. Distribution of daily conversation durations (left)
Figure 5. SNR distribution during daily conversations (right)

We conducted a mini-deployment study of a preliminary version of SocioDigest; Section 6.4 discusses the settings and lessons.

Tug-of-War: In group meetings or brainstorming, the level of participation of each individual may vary greatly; there might be someone who mostly remains silent, whereas a few might talk excessively, unwantedly giving others few chances to talk. It was reported that encouraging balanced participations from all individuals yields better outcomes in brainstorming [22].

Tug-of-War is a smartphone application that monitors turn-takings of participants and provides in-situ graphical feedback of how long each has talked so far. It is inspired by SensibleOrb [33], which employed dedicated wearable sensors called Sociometric Badges to monitor individuals' utterances. While we do not claim that its design is novel, the objective is to provide the key features of SensibleOrb on commodity mobile devices in everyday group-meeting setting. Using SocioPhone APIs and the participants' own smartphones enables convenient, rapid, and low-cost development of the monitoring functionalities of SensibleOrb. The lines of code of our prototype is only 75 (without counting those for GUI), demonstrating the effectiveness of SocioPhone to facilitate the development of interaction-aware applications.

3. CHALLENGES IN DAILY CONVERSATION MONITORING

We studied characteristics of daily conversations in real-life settings to understand the key requirements for our platform. To this end, we collected real-life conversation data using a custom smartphone logger that continuously recorded sound and performed off-line extraction of conversation through a voice-activity detection tool [37]. We deployed the software to five university students and collected data for ten days (total 753 user-hours). Although our dataset is limited in size and population, analyzing such real-life data gives us valuable insights into the challenges of daily-interaction monitoring.

Interaction patterns: The following observations strongly influenced the design of SocioPhone. First, we found that participants spend 4.5 hours a day, on average, in face-to-face conversations. This shows that new mobile applications to support our daily interactions have the potential to appeal to many developers and users. Also, we can see that conversation monitoring should be performed in an energy-efficient way to support such long interaction times. Second, conversations consist of many short speaking turns. Figure 3 illustrates a turn-taking history of speakers in a sample conversation that we collected using throat microphones

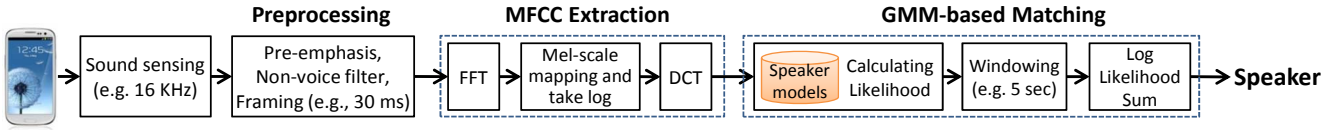


Figure 6. Typical speaker recognition pipeline

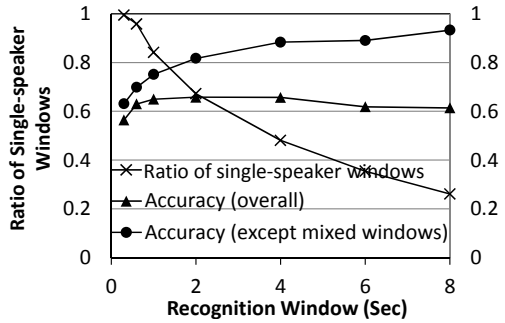


Figure 7. Tradeoff of window size

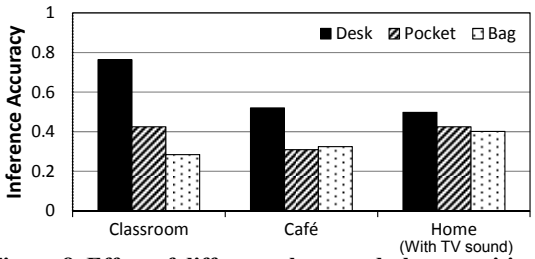


Figure 8. Effect of different places and phone positions

Table 4. Power cons. of speaker recognition on Galaxy Nexus

Component	Idle	Sensing	Preprocessing	MFCC	GMM	Total
Avg. power (mW)	13.5	160.9	4.0	54.2	204.2	437

(see Section 6.1). In the figure, we find that short, spontaneous turns dominate the conversation. Thus, daily-conversation monitoring must capture such short turn-takings. Third, Figure 4 shows that more than 50% of conversations last more than 5 minutes. Moreover, conversations lasting longer than 5 minutes account for 83% of the total conversation time, and conversations longer than 10 minutes do for 70% of the total time. Separating the short, active learning phase and the long, energy-efficient monitoring phase is a key aspect of our design. We will describe this in Section 4.2.

Environmental characteristics: Real-life acoustic environments are largely different from ideal lab environments, especially in terms of noise, making everyday conversation monitoring challenging. To understand these noise characteristics, we initially analyzed Signal-to-Noise Ratios (SNRs) during conversations; we measure the SNR values by applying the WADA-SNR library [21] to the conversation periods. Figure 5 shows the broad range of SNR values in real-life situations, mostly from -5 dB to 45 dB. The quality of recorded sound could vary greatly according to place (e.g., a silent meeting room, a noisy coffee shop), phone positions (e.g., on a table, in a pocket), and performance of microphones. This implies that conversation monitoring should be robust enough to handle noisy real-life environments.

3.1 Limitations of Existing Techniques

As a baseline approach, we can consider a representative speaker-recognition method [7][27] that has been well-established over several decades. Figure 6 shows its processing pipeline. It first splits continuous sound data into fixed frames, extracts cepstral features (MFCC) from each frame, and matches them with pre-built Gaussian mixture models (GMM) of MFCCs, containing unique vocal features of speakers. We now summarize key limitations of this approach for daily conversation monitoring.

Slow, inaccurate speaking turn detection: The baseline speaker-recognition pipeline hardly detects the highly-interactive turn-takings of daily conversations. This is because it generally requires 3-8-sec windows for reliable recognition, while turn-taking events often occur within smaller windows; note that a study reports two seconds of average turn length [2]. Figure 7 shows that as the window size increases, a window is more likely to contain multiple people's speech, degrading the accuracy of speaker recognition. (see Section 6 for the definition of accuracy) One may consider reducing the window size, but the accuracy drops significantly when a window size is too short. As the pipeline relies on the spectral signature of a person's speech, it must listen long enough to obtain statistically representative spectrum from the speaker and thereby identify who he is reliably. Instantaneous spectrum largely varies even for a single speaker, depending on his intonation and which consonants he pronounces [40]. With a short window, such so-called "atypical" sounds easily dominate the overall spectrum, making model matching difficult.

Vulnerability to real-life acoustic environments: The accuracy of a speaker-recognition pipeline can be easily compromised by background noises and phone positions in real-life situations. Figure 8 shows the effect of noise in different places, i.e., a quiet classroom, a noisy café, and a living room with TV sound, as well as different phone positions, i.e., on the desk, in the pocket, and in the bag (see Figure 5 for SNR of each place); we used a 4-second window, which provides the highest overall accuracy. The results are mainly attributed to several factors, namely poor SNR, noise-vulnerability of MFCC [5], and GMM-mismatch in real, distorted data. While there are solutions to handle these problems such as noise cancellation, in-situ model building, and collaborative sensing [4][31], their improvements are known to be limited.

High energy consumption: The baseline pipeline consumes a significant power. As shown in Table 4, the overall recognition process consumes 437 mW on a Galaxy Nexus phone; its 1750 mAh battery will drain in about 14 hours to only perform the recognition. In particular, MFCC extraction and GMM matching require 54 mW and 204 mW, respectively. A system could filter out non-voice parts to avoid frequent execution of resource-demanding recognition logic [27][29]. However, the logic still needs to examine entire conversations, which are long enough (e.g., 4.5 hours a day) to significantly impact the battery life.

Limitation of existing collaborative sensing approaches: Recent work exploits collaboration opportunities with nearby phones for effective context monitoring [26][31]. These approaches can be applied for conversation monitoring. One may execute the recognition pipeline on co-located phones and aggregate their inference results for better accuracy [31]. Alternatively, for resource saving, only one phone may run the pipeline and share the results with the others [26]. However, since these systems still use a conventional approach to speaker recognition, they suffer from low accuracy when detecting frequent, short turn-takings.

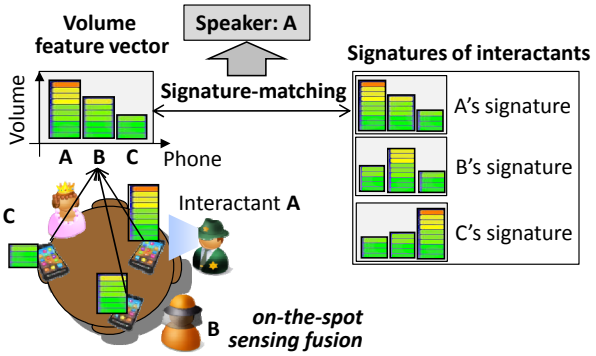


Figure 9. Illustration of online turn monitoring

4. IN-SITU TURN MONITORING

To address aforementioned challenges, we devised a novel turn monitoring technique. In this section, we present the details of our turn-detection algorithm and practical implementation issues.

4.1 Overview

Consider a group conversation scenario with three people as in Figure 9. When a person speaks, multiple phones acting as wireless receivers can capture the sound signals that the person (or transmitter) generates. Each phone measures a speaker's voice signal strength (or volume in μPa). When a speaker's phone is placed right next to the speaker (mostly true in practice), this phone is likely to measure the strongest signal strength among all the neighboring phones. A simple approach to speaker recognition is then to select a phone (and its owner) that has the strongest signal strength; this naïve method is called a *Volume-peak-based algorithm*. In real-life situations, however, this approach has the following limitations: (1) location and placement of phones are not controllable (e.g., a phone may be placed in a pocket), (2) some of the phones may not be available (e.g., due to limited resources or poor recording quality), and (3) peak detection is susceptible to background noise.

To handle such limitations, we devise a *Volume topography-based method* that leverages the relative difference of recorded signal strengths over multiple phones. As in Figure 9, speaker A's voice has been recorded over three phones with different volumes (represented as a volume vector). Due to relative position differences, each speaker will have a unique volume signature (or topography) over three phones. These phones can collaboratively build a topography database a priori (say during a learning phase), and we can identify the speaker by matching a newly measured volume vector with the topography database.

Our method is advantageous in several ways. First, it is much lighter than existing speaker recognition systems like [27], since we limit complex signal processing only in the learning phase. Second, volume vectors can be reliably obtained even with a very short sensing window, e.g., 300 ms, and thus enable turn-taking monitoring in a highly agile way; a turn is simply extracted by aggregating consecutive results. Third, the volume topography is less vulnerable in noisy acoustic environments; the background noises easily distort the users' voice spectra, but the topography itself is mostly consistent as long as the spatial placements of the phones and the speakers are consistent. In addition, the volume topography can be quickly re-trained in-situ to update phone positions and noise characteristics. Such in-situ topography also enables our method to work even when some phones may not be available (i.e., number of monitoring phones < number of users).

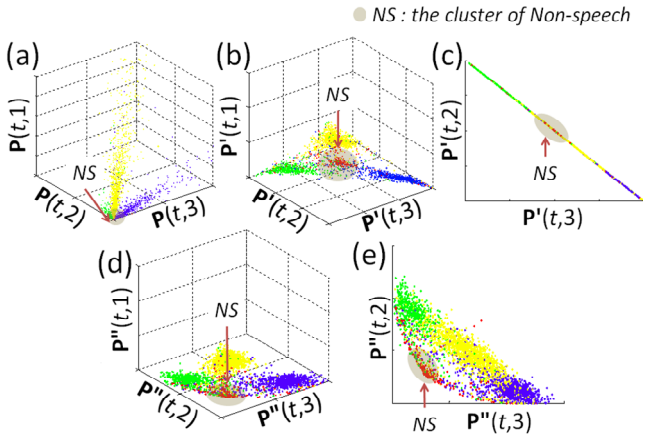


Figure 10. Distribution of feature vectors

4.2 Volume Topography-based Algorithm

Training data collection: During the learning phase, each phone samples the incoming sound at the rate of 8 kHz. The sampled audio stream is segmented into 300 ms-frames (i.e., 2,400 samples). For a given time t, each phone i calculates $p(t,i)$, the power of the frame from phone i at time t, i.e., the average of the square of the audio signals. Thus, we have a feature vector, $\mathbf{P}(t) = (p(t,1), p(t,2), \ldots, p(t,n_p))$, where n_p is the number of monitoring phones; note that n_p may not be equal to the group size. For adequate learning, phones collect the feature vectors for L seconds, where L is a system parameter for a learning period. We use L=60 seconds in three-user experiments, obtaining 200 vectors in total.

Feature vector transformation: One of the key challenges is to define the feature vector so that it has discrimination power. Our initial approach was to simply use $\mathbf{P}(t)$ itself. Figure 10(a) plots $\mathbf{P}(t)$ for a three-user group as in Figure 16(a). In this case, we were able to differentiate three users, but we found that this approach performs poorly in discriminating non-speech turns (or silent turns). Our alternative was to normalize the vector as $\mathbf{P'}(t) = \mathbf{P}(t) / \mathrm{E}(t)$, where $\mathrm{E}(t)$ is an average of a vector $\mathbf{P}(t)$. Figure 10(b) plots $\mathbf{P'}(t)$ for the same situation. This approach distinguishes human speech from non-speech well. However, we find that discrimination is weak when the number of phones is less than the group size due to loss of degrees of freedom (i.e., the sum of $\mathbf{P'}(t)$ is always 1). Figure 10(c) shows $\mathbf{P'}(t)$ with one fewer phone.

To overcome this, we define the feature vector as the product of $\mathbf{P'}(t)$ and the decibel measured on phone i, i.e., $\mathbf{P''}(t) = \{\mathrm{D}(t,1) \times \mathrm{p}(t,1) / \mathrm{E}(t), \ldots, \mathrm{D}(t,n_p) \times \mathrm{p}(t,n_p) / \mathrm{E}(t)\}$, where $\mathrm{D}(t,i)$ is defined as follows

$$D(t,i) = 20\log_{10} {}^{p(t,i)}/_{p_{ref}},$$

where p_{ref} is the standard reference sound pressure level, i.e., 20 μPa. In addition to the second approach, it discriminates better even with fewer phones. Figure 10(d) and (e) show $\mathbf{P''}(t)$ using three and two phones, respectively.

Topography generation: From the training dataset, we build a set of audio-signal signatures, i.e., *volume topographies*, for each group member plus the non-speech case (the moment when no member speaks). For an n-member group, we use k-means clustering where k is set to n+1.

Classifier training and classification: The input dataset collected during the learning phase is used, namely feature vectors labeled with a cluster-ID. We select a multi-class SVM classifier, known as

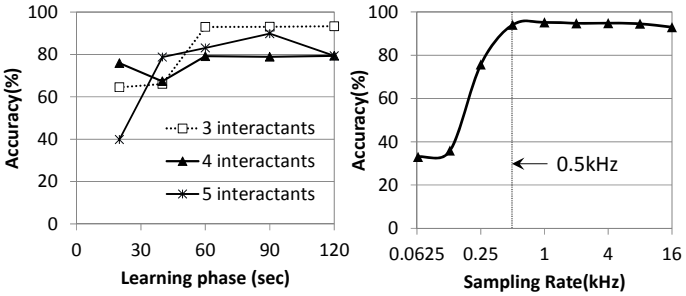


Figure 11. Effect of learning phase duration (left)
Figure 12. Effect of sampling rate (right)

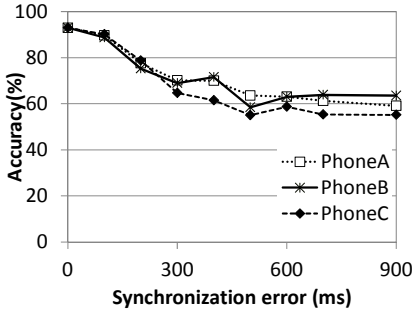


Figure 13. Effect of synchronization error

one of the best performing classifiers [1]. After training has completed, SocioPhone segments turns online by simply mapping incoming frames into cluster-IDs using this classifier. According to our experiences, training duration is short (around one minute), and turn monitoring can start shortly after a conversation starts.

Turn recognition: A turn is detected if two consecutive frames belong to different clusters. We do not consider non-speech turns in a user's speech of less than 300ms; they are regarded as small pauses and often ignored [2].

Mapping audio signatures (cluster-IDs) to group members (member-IDs): In the learning phase, we also build a mapping table that converts cluster-IDs to member-IDs. We use a conventional speaker recognition technique [27]. Each phone trains the recognition algorithm for its owner a priori by building a reference speech model. At the end of learning phase, each phone uses all original frames that belong to each audio signature to generate MFCC and compute GMM likelihood. The cluster head collects the GMM likelihoods from its members and determines the mappings of cluster-IDs onto member-IDs.

4.3 Other Practical Issues

Energy-efficient conversation detection: The first step of turn monitoring is to detect whether a group conversation has started. Given that a conversation starts when people talk with one another, SocioPhone periodically monitors ambient sound to detect voice activity (e.g., analyzing about 2-sec-long audio signals in every 30 seconds). To be precise, the incoming sound wave is sampled at the rate of 8 kHz and the samples are segmented into frames. The duration of a frame is 2,048 ms (i.e., 16,384 samples per frame). For a given frame, we calculate two metrics, root mean square (RMS) and zero-crossing rate (ZCR). Then, we decide whether the sound is human speech using an offline-trained decision tree, which is commonly used in human-speech detection [27].

Group formation and head selection: Upon the detection of voice activity, SocioPhone discovers nearby friends by performing Bluetooth scanning. It retains MAC addresses of a user's friends; the list can be collected using conventional peer introduction mechanisms (only once per friend) [13]. If it finds any registered friends, a group network is formed. In a group, one phone is selected as a head and coordinates the collaborative turn detection; it collects volume features from other phones, matches them to the topography, and shares the results. The head is randomly selected; the difference of resource consumption between a head and a member is marginal (See Section 6.3).

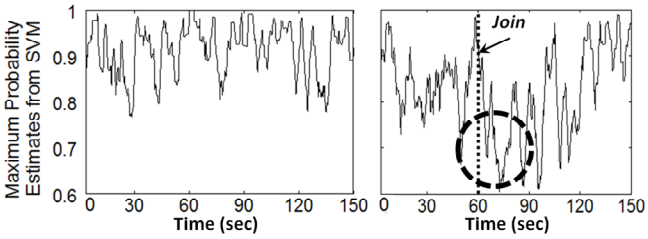
Duration of a learning phase: Each user should speak at least once in a learning phase. To determine the proper duration, Figure 11 shows the accuracy while varying group sizes and learning durations. For the group sizes of three or four, accuracy tapers off around the 60 seconds, whereas it tapers off around the 90 seconds for group sizes of five. Reasonably assuming that the learning duration is proportional to the group size, we set the duration to the group size $n \times 20$ seconds; our experiment shows 95% of speaking turns are shorter than 20 seconds (see the details in Section 6.2). During the learning phase, the topography-based turn monitoring will not be available on-the-fly. SocioPhone can apply the volume-peak-based algorithm in parallel, which does not require any training a priori. Also, it is worth noting that many daily conversations last quite long as discussed in Section 3.

Sampling rate selection: Figure 12 shows that SocioPhone achieves highly stable accuracy at sampling rates as low as 500 Hz (see Section 6 for configurations). Using low sampling rates has two major benefits: energy saving by reduced computation and privacy preserving even if the sampled speech is temporally stored. We elaborate the latter one. By the Nyquist sampling theorem, with the speech signal sampled at 500 Hz, we can reconstruct only the signals whose frequencies are no higher than 250 Hz, i.e., half the sampling frequency. Then, we refer to the *articulation index* (AI), a value quantifying the intelligibility of a given speech signal [14], where AI = 1 for most intelligible, zero for completely unintelligible. For example, AI is 0.9 for a low-passed speech signal cut off at 5000 Hz, 0.1 at 500 Hz and zero at 250 Hz or lower. Therefore, the speech that SocioPhone samples is largely unintelligible, which potentially preserves users' privacy.

Time synchronization: To align feature vectors at the same time, we synchronize the phone clocks by well-known means (e.g., GPS or NTP). Note that phones may be out of sync by 1-2 seconds in WCDMA network. To investigate the required level, we performed an experiment with a 3-user group by deliberately making synchronization errors in one of the phones. Figure 13 shows that our algorithm tolerates about 100ms of errors. SocioPhone periodically checks the availability of GPS (once a day when a user is outdoors) and fixes the time from the GPS receiver whose time is accurate to 200 nanoseconds.

4.4 Discussion on Potential Improvements

Detection of a conversation group: In our current design, SocioPhone simply identifies conversation group members by an initial Bluetooth scan. We assume a single-group interaction among collocated friends. This assumption holds in many daily life situations, but sometimes groups may be partitioned into subgroups. We admit that further study is required to enable robust and practical detection of conversation groups, especially to deal with such multiple sub-group situations. One possible way would be to dynamically divide the sub-groups by analyzing overlapping speech patterns [6]; note that overlapping speech is limited within a single conversation group as people often speak once at a time whereas



(a) w/o dynamic event (b) w/ dynamic event

Figure 14. SVM probability estimates over time

overlapping frequently occurs among different conversation groups within a place.

Selection of monitoring phones: In real situations, participation from all available phones does not always guarantee the best accuracy; excluding a phone may achieve higher accuracy if that phone shows poor recording quality. However, it is challenging to estimate the expected accuracy in advance. One possible approach is to check if the signal-to-noise ratio (SNR) is above a certain threshold, but reliable SNR calculation is difficult and also consumes much power. Another alternative approach is to leverage phone placements. For example, from our empirical studies, we observe that turns are monitored more accurately by excluding phones in bags. To apply this method, we need to incorporate phone-placement detection, such as [30]. Besides accuracy, we can further consider the available power of phones to select the monitoring phones. For example, SocioPhone can exclude phones with little battery remaining, e.g., < 10%, if the number of phones is greater than three and most are qualified.

Noise reduction: Our technique is resilient to some forms of noise such as *ambient noise* that may persist or fluctuate but uniformly applies to all participating phones. An example is the background human utterances in a restaurant where people at the surrounding tables are chatting in similar tones. However, volume-topography may be vulnerable to nearby *point-sourced noise*; for example, an announcement from a nearby loud speaker or a cup rattling next to a specific phone. Given such a point-sourced noise, the large variance of phone-to-source proximity significantly distorts the volume topography. To improve robustness against such point-sourced noises, pre-filtering of non-human vocal spectrum at the recording stage would narrow down the vulnerable bandwidth. Techniques like spectral subtraction and Wiener filtering [8] could be leveraged for this purpose.

Handling dynamic situations: Our technique properly operates when the relative positions of users and their phones are mostly fixed. However, diverse events may dynamically occur during a conversation, e.g., join and leave of a new member, moving phones, turning on a TV, which potentially compromises the monitoring accuracy. First, if the topography is successfully built in the learning phase but such dynamic events appear during the monitoring phase, we believe that the probability estimates of the SVM classifier can be used to handle the events. Figure 14 shows empirical behaviors of the probability estimates in the case of a group conversation with three users. The value mostly remains above 0.8 without any dynamic event (Figure 14(a)). When a fourth person joined the group and started speaking (at the 60 second mark), the values dropped to around 0.6 for about 10-20 seconds, as shown in Figure 14(b). For such a sudden drop within a predefined duration, the topography can be retrained in the background (during which the old ones are still used). Second, the topography training and associated classification can be spoiled when dynamic events

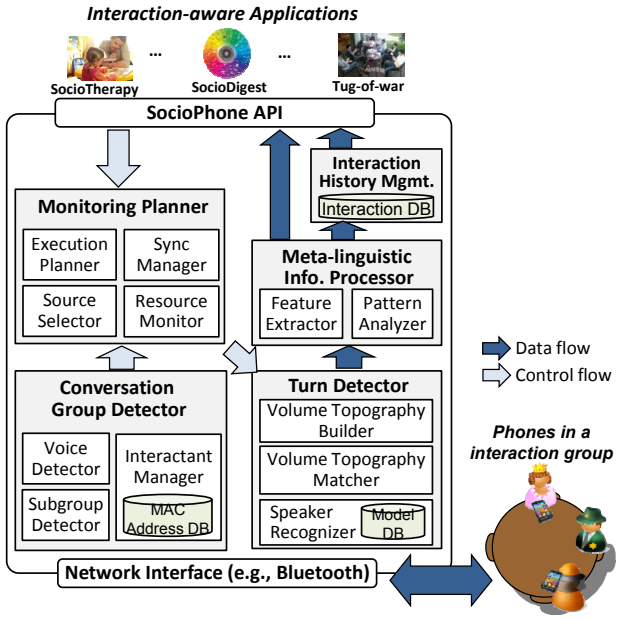


Figure 15. SocioPhone system architecture

occur during the learning phase. In such a case, the clues suggesting a retraining may be found from multiple sources, e.g., erratic turn-taking patterns which are unlikely in normal conversations, considerably low probability estimates, etc.

Detection of overlapping speeches: As our technique classifies each speech frame into a single speaker, it fails to detect overlapping turns in which the multiple speakers talk at the same time. However, the portion of such overlapping speeches is not significant in our daily conversation. From our experiment with three people in a café (Figure 17 (c)), the total time of overlapping speech is under 10% of the total conversation time; a study also reports the overlapping ratio from 6% to 14% [39]. Also, most overlapping speeches are short, less than 2 seconds. Accordingly, meta-linguistic features can be extracted properly regardless of the overlap. For some applications, however, overlap detection can be useful; for example, the successful interruptions are considered to infer the leadership in the group discussion [35]. Note that even in field of speech diarization, identifying overlapping speech and the associated speakers remains an on-going challenge.

5. PLATFORM IMPLEMENTATION

We have implemented a SocioPhone prototype in Java using Android SDK 4.0. It runs as a middleware and fully supports the SocioPhone APIs. Figure 15 shows the system architecture of the prototype. We implemented *Turn Detector* and *Conversation Group Detector* using the techniques introduced in Section 4. Here, we briefly explain the role of other system components.

Monitoring Planner decides how to perform turn monitoring. Its key role is to determine the feasibility of collaborative turn detection. *Source Selector* first figures out how many phones participate; it checks if the phone has sufficient battery power and if its sound signals are clear enough for discriminative volume topography. If there are sufficiently many sources available, *Execution Planner* performs turn monitoring with the volume-topography-based method. Otherwise, it performs conventional speaker recognition. Note that SocioPhone may ask users to place their phones in a better position when the collaborative method is not possible.

Table 5. SocioPhone evaluation parameters

Parameters	Default values	Other values used
# of interactants	3	4, 5
Place	Seminar room	Home, café
# of avail. phones	3	2, 4, 5
Phone Position	On a table	In a pocket / a bag
Mic. Direction	To the owner	Reversed

Meta-linguistic Information Processor computes rich meta-linguistic contexts, based on the turns computed by *Turn Detector*. Additionally, *Feature Extractor* processes prosodic features such as volume, pitch, and their variation over segmented sound signals. *Pattern Analyzer* infers a number of meaningful social contexts by combining turn information and prosodic features. In the current prototype, it supports the following contexts: dominance and leadership in a conversation group, conversation asymmetry, interactivity, and sparseness. To infer the dominance and leadership, *Pattern Analyzer* applies a supervised SVM over the turn and prosodic features [3][17]. It also identifies interactivity, sparseness, and skewness, applying heuristic metrics as follows:

- *Level of interactivity*: # of speaking turns per minute
- *Level of sparseness*: # of non-speaking turns over three seconds per minute
- *Level of skewness*: standard deviation of # of speaking turns for all participants

Besides the above examples, *Pattern Analyzer* can flexibly incorporate other algorithms to infer diverse contexts. For example, emergent leaders in a conversation group can be further inferred using the method in [35]. Another method can infer expressiveness from volume and pitch [38]. We leave detailed evaluation of these derived contexts as a future work.

Interaction History Manager supports SQL queries from applications. To support the queries, it stores 'conversation session' and 'turn' information in an internal database. For efficiency, SocioPhone holds the turn information for the on-going session in the memory, while flushing it to persistent storage when the conversation completes. Internally, it is implemented using SQLite, a light-weight database in Android.

Network Interface: SocioPhone uses Bluetooth for peer discovery and communication. We considered using Wi-Fi Direct since it provides adequate features such as ad-hoc peer discovery and message broadcasting. However, it consumes too much power to use in everyday monitoring, as it is designed for short-term high-bandwidth communication. According to our measurements, exchanging messages every second through Wi-Fi Direct requires about 413 mW of power. For the same, Bluetooth communication requires only 138 mW.

6. EXPERIMENTS

Our goal is to fully evaluate SocioPhone's performance under a range of real-life situations. However, since real-life sound sensing is affected by a number of factors simultaneously, a direct, fully unorganized deployment would make isolating the root causes of performance changes extremely challenging. As an initial step, we carefully select representative real-life scenarios, and we identify independent parameters that may largely affect SocioPhone's performance, as shown in Table 5. Then, we rehearse diverse variants of the scenarios by applying different combinations of the parameters to understand the causality of performance inclines or declines. Based on such understanding, we also describe our experiences and lessons learned from subsequent unorganized real-world deployments of SocioPhone.

Figure 16. Experimental setup

6.1 Experimental Setup

Scenarios and parameters. For performance evaluation of SocioPhone, we consider three conversation situations in different places, i.e., seminar room, home, and café (See Figure 16(a)-(c)). We vary the following parameters to reflect diverse real situations: the group size, the number of available phones, the phone positions, and the direction of microphones. By default, we assume a casual conversation with three participants. Each participant's phone is placed on a table and the microphones are directed to their owners. Table 5 lists the default values and variations. Each conversation is 15 minutes of unscripted, free talking. For all experiments, we use Galaxy Nexus phones.

Alternative techniques we developed for comparisons:
SinglePipe is a conventional speaker recognition system, as shown in Figure 6. Each phone runs its own recognition pipeline and uses the results separately. The performance is reported as the average value measured over all phones.

CombinePipe is developed based on DarwinPhones [31]. It runs SinglePipe on every phone and makes the final inference by combining GMM likelihoods to improve accuracy.

SharePipe applies the idea from CoMon [26]. Among multiple phones, only a single phone runs SinglePipe and shares the results with other phones to save energy. We omit the inference accuracy of SharePipe, since it is expected to be the same as SinglePipe.

All the alternatives are built on conventional speaker recognition methods. We attempt to carefully select their parameters to show their best performance. First, from the previous lessons [31], we apply well-crafted speaker models for each situation and use only the models of the interactants participating in the conversation session. Also, as these methods generate results over a fixed-size sensing window, we apply a four-second window by default, which provides the best accuracy in our experiments.

Evaluation metrics: We adopt two key evaluation metrics: accuracy and resource efficiency in terms of energy and CPU. We measure energy consumption using a Monsoon PowerMeter.

Turn-monitoring accuracy: We adopt the duration-weighted accuracy used for speaker diarization [2]. It is the ratio of correctly

Table 6. Monitoring result in default situation

System	Monitoring accuracy (%)		
	Accuracy	Precision	Recall
SinglePipe	76.3	76.3	85.9
CombinePipe	80.4	80.4	90.6
SocioPhone	92.9	97.1	94.1

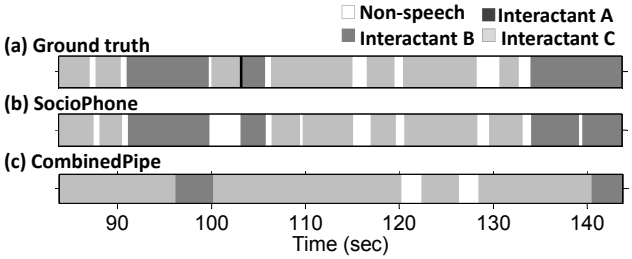


Figure 17. Turns over time

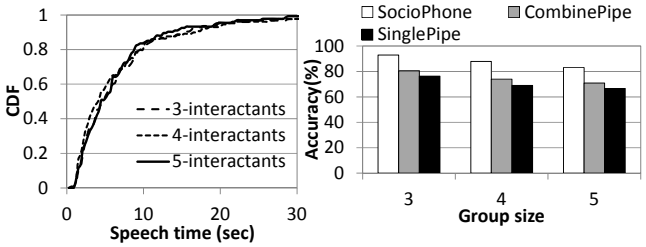


Figure 18. CDF of speech duration (left)

Figure 19. Accuracy with different group size (right)

inferred time to the total time of the conversation. A conversation session is segmented by the start and end times of the turns that are either annotated in ground truth or inferred by the evaluating technique. Each segment is labeled as true-positive (TP), false-negative (FN), false-positive (FP), and true-negative (TN). If the speaker of a segment in ground truth is identical to the inferred one, it is labeled as *TP*; in the case of non-speech, *TN* is tagged. *FN* means a speaker in ground truth is not found or incorrectly inferred. *FP* means a speaker by the inference is not in the ground truth. Based on the label and the segment duration, we define three metrics as followings:

- Accuracy = {D(*TP*) + D(*TN*)} / total time
- Precision = D(*TP*) / {D(*TP*) + D(*FP*)}
- Recall = D(*TP*) / {D(*TP*) + D(*FN*)}

where D(*tag*) is the total time of segments labeled as *tag*.

Ground-truth annotation: Correct ground truth is a precondition for the integrity of monitoring accuracy. For accurate and fine-grained annotation, we use throat microphones (See Figure 16(d)). The throat microphone records only its wearer's voice while suppressing external sound; it directly senses throat vibrations instead of vibrating air molecules. We also videotaped all conversations for post-hoc analysis. Note that manual tagging did not work properly due to the highly interactive nature of real-life conversations and difficulty of accurately tagging the start and end of a turn. Also, manual tagging was inconsistent across persons.

6.2 Turn-Monitoring Accuracy

We investigate turn-monitoring accuracy at the default setting. Table 6 summarizes the results. SocioPhone shows the highest accuracy, 92.9%; it accurately and quickly detects turn-takings by inspecting volume vectors every 0.3 seconds. The others show around 80%. They hardly segment the turns precisely due to their larger 4-second window for reliable inference. Note that CombinePipe slightly outperforms SinglePipe, since phones are in a similar situation and the combined inference benefit is marginal.

To see the detailed differences, we plot partial results from SocioPhone and CombinePipe over the ground truth as in Figure 17. SocioPhone captures the overall turn-taking pattern well. CombinePipe also recognizes speakers well in long-speaking turns, but often misses short, interactive turns. Figure 18 plots the CDF of speaking-turn durations. 45% of turns are less than four seconds, which is the window size of SinglePipe-based pipelines. More than 80% of speeches are less than 10 seconds, implying the importance of fine-grained turn segmentation for casual conversation. We find similar patterns in the 4- or 5-interactant conversations. Note that the topic or type of conversation could change the distribution, but the general trend would remain stable.

We observe that the interactants' speaking turns are sometimes overlapped. In our experiments, overlapped speech accounts for 1%-10% of the whole session time; the average duration of overlapped turns is 0.8-1 second. Interestingly, all the techniques mostly choose one speaker among the actually speaking speakers.

6.2.1 Effect of Number of Phones

We investigate the effect of the group size and the number of available phones on the turn-monitoring accuracy. In addition to the default setting, we consider two more situations with four and five interactants. Figure 19 shows the results with different group sizes. SocioPhone outperforms the others regardless of the group size by 12-19%. Even with 5 interactants, it shows the accuracy of 83%, while the accuracies of other techniques are below 70%.

We further examine the accuracy while varying the number of phones actually monitoring. We report the average accuracy over all possible combinations; e.g., in the case of three phones for five interactants, we calculate the average accuracy for all 10 combinations. We exclude SinglePipe since it runs on one phone. As shown in Figure 20, SocioPhone outperforms CombinePipe except when only two phones are available. This shows that the volume topography-based method works well even if a small portion of phones is unavailable, e.g., 78% accuracy with only three phones and five interactants. When the number of available phones is much smaller than the group size, our method performs worse than CombinePipe, e.g., two available phones for a 5-interactant conversation. In these cases, SocioPhone had better use the conventional speaker recognition method.

6.2.2 Effect of Phone Placement and Direction

We then investigate the effect of phone placement and direction. To equalize external variables such as noises and conversation patterns, we simultaneously deploy multiple phones on each interactant (Figure 16(d)). In this section, we omit the results with 3 and 4 interactants since their results are similar to a 5-interactant.

Effect of phone placement: Figure 21(a) depicts the accuracy in 5-interactanat conversation by increasing the number of phones in pockets. Except the case of (5,0) and (0,5), we report the average accuracy over all possible combinations. SocioPhone shows around 75% of accuracy even with three phones placed in a pocket and two phones on a table. This is similar to the accuracy of CombinePipe with all five phones on the table. CombinePipe also outperforms SinglePipe since a few phones with better sound quality disproportionately contribute to the results.

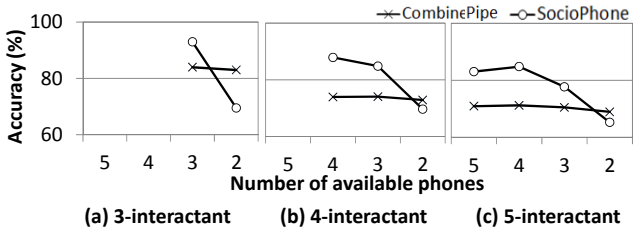


Figure 20. Effect of number of available phones

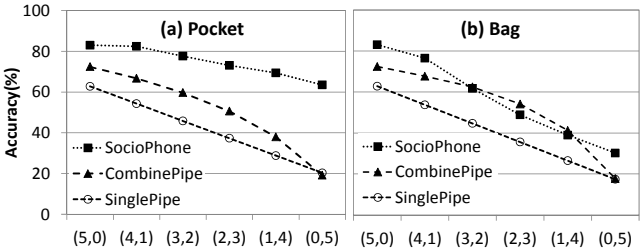


Figure 21. Effect of phone position; (X,Y), X: number of phones on a desk, Y: number of phones in a pocket or in a bag

We find an interesting result in the 3-interactant conversation (see Figure 23 for the setting). Figure 22 shows that the F1 score of each interactants depends on which phone is placed in a pocket. An F1 score is the harmonic mean of precision and recall. When a user *B* in Figure 23 puts his phone in his pocket, the accuracy is much higher and F1-score of interactants are more balanced, compared to other cases. From the video review, we find that the relative distances between interactants are different. The distance between *A* and *C* is much shorter than B. Interestingly, it is unexpected that uniformly distributed phones on the table will be more helpful. We speculate that an imbalance of recoding volume makes inference more difficult for users with relatively close positions such as A and C when C's phone is in C's pocket. This implies that the relative position of interactants as well as the phones will be a key to estimating the expected accuracy.

Figure 21(b) shows the accuracy putting some phones in a bag. In some cases, CombinePipe outperforms SocioPhone, but the overall trend is similar to the previous experiment. The accuracy of all pipelines is much lower than the previous experiment due to lower quality audio.

Effect of phone direction: We next measure the accuracy by varying phones' direction. The direction hardly affects accuracy. This is because the length of the smartphone is much shorter than the relative distance among smartphones, and thus the volume level or frequency-domain features are well maintained.

6.2.3 Effect of Places

We next examine the effect of background noise on turn-monitoring in different places: seminar room, home and café. In a café, background music is played and other guests are chattering. For home, we experimented in a living room with a TV turned on. To quantify the sensed audio quality, we use the SNR, with the method presented in Section 3.

Figure 24 depicts the average SNR in the three places. As expected, the SNR in the classroom, 40.7, is much higher than those in the home and café, 28.4, and 11.8, respectively. The SNR in home is also different from that in café. It might be due to the directivity of the microphone as well as the ambience of the noise. Café noise is spread out, whereas TV sound at home has more directivity.

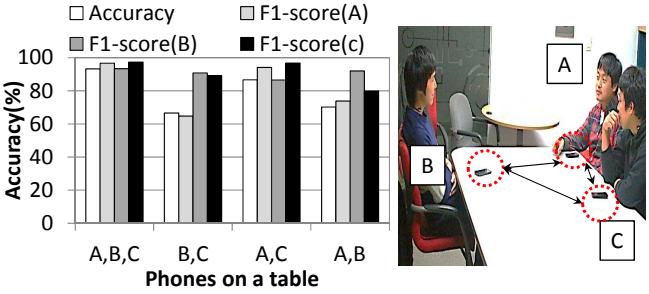


Figure 22. Effect of phone position (left)
Figure 23. Relative position of interactants and phones(right)

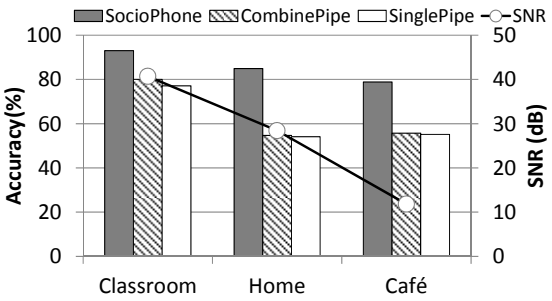


Figure 24. Accuracy on different places

Due to the degradation in audio quality, the monitoring accuracy at home or in the café drops compared to the seminar room, for all techniques. However, even in the café, which is an uncontrolled, noisy situation, we could observe that SocioPhone performs effectively at about 80% accuracy, whereas SinglePipe and CombinePipe show accuracy under 60%. The reason why SocioPhone is more robust against background noise may result from the different characteristic of MFCC features and the volume topography. Due to the logarithms, MFCC is easily influenced by low energies from noise. However, even in a noisy situation, people tend to speak louder than the background noise. Thus, a phone can still record an interactant's voice louder and thus, the volume topography will be maintained more stably.

6.3 Resource Usage for Turn Monitoring

6.3.1 Cost of Turn Monitoring

We evaluate the system cost for turn monitoring in terms of power consumption and CPU utilization. Table 7 shows the results of SocioPhone and the other techniques with the default setting in Table 5. All techniques but SinglePipe operates in two modes, head and member. A head takes charge of the coordination and final inference. A member transmits the required information to the head. Overall, SocioPhone consumes much less power at higher accuracy in turn monitoring; it consumes about 280 mW, whereas others range from 436 to 512 mW; a 1750 mAh battery would last about 23 hours with SocioPhone and 12-14 hours with others. The difference of power consumption between a head and a member on SocioPhone is marginal because of the light-weight matching. For SharePipe, a member consumes much less power, i.e., 89.3 mW since it performs no recognition-related processing. Interestingly, CombinePipe's member consumes 35.3 mW more than the head, since Bluetooth consumes more power for transmission than for reception. SocioPhone also uses much fewer CPU cycles by avoiding complex speaker recognition.

Table 7. Energy consumption and CPU utilization

Pipeline	Head		Member	
	Power(mW)	CPU(%)	Power	CPU
SinglePipe	436.8	15	N/A	N/A
SharePipe	481.9	17.3	89.3	< 2
CombinePipe	476.8	19.2	512.1	18.3
SocioPhone	282.1	< 2	278.6	< 2

Table 8. Power breakdown

Operation	Power(mW)	Ratio
Idle	13.5	0.05
Recording	160.9	0.57
Feature computation	5.7	0.02
Classification	3.5	0.01
Communication	98.6	0.35

Table 9. CPU time in the learning phase

# of interactats	3	4	5
(1) Clustering	3.4 ms	13.6 ms	15.2 ms
(2) SVM training	89.0 ms	125.8 ms	147.2 ms
(3) Speaker labeling	4624.8 ms		

6.3.2 Cost Breakdown of Turn Monitoring

For turn monitoring, SocioPhone continuously performs three common operations: (1) sound recording, (2) feature computation, and (3) feature and result transmission. A head performs extra operations for learning and matching.

Table 8 shows the power breakdown of SocioPhone. The top power consumer is sound recording, i.e., 57% of the entire power. The sound recording includes acquiring the wake lock in Android, about 30 mW. Bluetooth communication consumes about 98.5 mW due to frequent messaging every 0.3 seconds. However, logging applications such as SocioDigest may not need instantaneous turn results, and messages can be buffered and bulk-transmitted. We measured that Bluetooth consumes only 58.7 mW when messaging every 10 seconds. We omit the CPU breakdown since the total CPU usage is less than 2%.

We measure the training time of a *head* of SocioPhone. In the learning phase, a *head* performs the following three operations: (1) K-means clustering, (2) SVM model generation, and (3) speaker labeling using MFCC and GMM. Table 9 shows the CPU time for different group sizes. Since speaker labeling is executed on every phone for the same workload, we report the average value. Interestingly, (1) and (2) take only 160 ms even with 5-interactant data for 60 seconds. The bottleneck is (3), mapping the clusters onto speakers, which takes four seconds. Offloading such complex processing into the server might be useful to further optimize SocioPhone.

6.4 Deployment Experience

We conduct additional experiments to observe SocioPhone's performance under more natural interaction situations. Here, we do not attempt to show general performance characteristics but present notable lessons on the performance and user experiences.

6.4.1 Deployment in Natural Situations

For experiments, we recruited four frequently-interacting graduate students (P_1, P_2, P_3, and P_4) who did not know about SocioPhone in advance, and installed SocioPhone on their own smartphones; all the subjects were males in their twenties. For a natural setting, we let them freely have conversation sessions at school for a weekday. For ground truth, we asked them to wear throat microphones upon the start of a conversation and also to video-record the conversation sessions by themselves using a tripod; each participant was compensated with KRW 50000 (about USD 45) to participate in the study.

Table 10. Precision and recall in a natural situation

	Precision(%)	Recall(%)	Time(sec.)
P_1(on the desk)	96.4	89.0	244
P_2(on the desk')	97.4	63.1	190
P_3(in the pocket)	98.0	54.0	98
P_4(in the bag)	98.5	76.4	199
Non-speaking	19.2	88.8	48
Total	97.4	74.1	779

We first look at a case in which they go to a seminar room for brainstorming. We could see that P_1 and P_2 put their phones on the desk, P_3 put his in his pant pocket, and P_4 put his in his backpack; in the case of P_2's phone, the microphone was not facing towards him. In this natural brainstorming, the overall accuracy of SocioPhone is about 75%. This is 13% lower than the case of four phones on the desk as in Section 6.2.1 (88%). Table 10 shows the results per participant including non-speaking turns. Interestingly, while the precision for speaking turns is very high overall (> 96%), the recall is not as good, especially for P_3 and P_4 whose phones are not in open-air positions; SocioPhone often misses their speaking turns. In the case of P_2, the recall is also quite low, 60%, even though his phone is on the desk; this is different than our previous observations, indicating that the direction of the microphone hardly affects the accuracy. From our video review, we strongly suspect that this is because P_2 speaks in a calm tone so that his turn is often identified as non-speaking turns; interestingly, others asked him several times to speak again.

We investigate another case of three of the participants going to a café to have a casual conversation. Unlike when brainstorming, all participants comfortably put their phones on the coffee table. Unfortunately, in this setting, the ground truth collected by throat microphones was not accurate as the participants did not wear them tight enough. Instead of investigating accuracy, from the video recordings and SocioPhone logs, we found several scenes that cause notable performance problems. In brief, very short, instantaneous noises often led to misclassification. First, after taking a sip of coffee, putting the cup on the table makes a loud noise, especially to the nearby phone. Second, P_3 sometimes shakes his leg and his leg touches the table. In addition, we notice that when P_2 uses his phone to check his Facebook (we asked him afterwards), his screen taps also create loud noises to the very phone, causing instant misclassification. We expect that the noise reduction techniques discussed in Section 4.4 can be further incorporated to filter out such instantaneous noises.

6.4.2 Experiences with SocioDigest

We conducted a mini-deployment of SocioPhone for three consecutive days, to encompass a broader set of our daily interactions and find lessons regarding further in-the-wild issues and user experiences. We recruited 15 users for SocioDigest introduced in Section 2.3, many of whom are within the same social circle, i.e., lab members.

This mini-deployment enlightened us about future considerations for SocioPhone to work fully robustly in-the-wild. For example, SocioPhone did not perform well under some conditions, such as when everyone keeps their phones in their pants pockets. This often

occurred when participants make unplanned small talk while standing. To circumvent it, we distributed wind vests to the participants and recommended that they put their phones in its chest pocket instead of the pants one. Besides, SocioPhone needs to be further improved to isolate the true conversation groups in some situations such as multiple independent conversation groups talking at the same time very closely.

On reviewing his daily conversation report, a user who is 31-year-old Ph.D. student, was surprised that most of his daily talks are concentrated on a few people. An interesting observation is that, on the first day SocioDigest reported that his wife is not highly ranked in terms of conversation length. On the next day, the total length did not change much but his wife and he exchanged quite more turns; he said it was his small effort to make better use of his limited amount of time at home.

7. DISCUSSION

Privacy: Privacy is a primary concern when audio recording is involved. Our SocioPhone design addresses privacy issues as much as possible. As SocioPhone works on mobile devices only, it does not provide any raw audio recording or rich features like MFCC to the third-party servers from which original speech can be inferred. More importantly, SocioPhone limits its sample frequency to 500 Hz and shares only simple volume features from which it is almost impossible to recover the original linguistic contents. Within a conversation group, partial exposure concerns still remain, such as the size of a conversation group or emotional tone of speech. We expect sharing of this kind of information would be reasonably acceptable among people in the same group.

In addition, malicious applications running on SocioPhone have the potential to secretly report any private meta-linguistic contexts. SocioPhone provides users with an access control interface by which users can easily ensure that only trusted applications access SocioPhone. We also expect future mobile OSs would incorporate real-time information tracing facilities to monitor unexpected usage of private data, as proposed in TaintDroid [12]. Finally, regardless of its privacy-preserving techniques, we admit the inherent limitations that nearby people might perceive intrusive from being audio-recorded itself [23].

Beyond meta-linguistic contexts: SocioPhone can incorporate existing speech recognition techniques to additionally provide semantic information like topics. For example, an application can recommend YouTube videos based on what a group has talked about so far. Supporting these advanced functions requires a better understanding of the resource requirements involved. The basic interaction awareness provided by SocioPhone can be an initial clue to determine when to selectively conduct heavy speech recognition given a device's limited resources.

We can further extend SocioPhone to capture visual cues such as gesture or eye contact. It is possible to adopt a gesture monitoring system like *E-Gesture* featuring energy-efficiency and resilience to activity-generated noises [34]. An interesting direction would be capturing eye contacts with new hardware like Google Glass.

8. RELATED WORK

Conversation analysis: Everyday social interaction has been a long-studied area in sociology. They studied formal models and methods to understand everyday interactions, such as video-taping a conversation and structuring a schematic by turns and their orders [15]. Our platform can provide a way to bring these research efforts and findings onto real-life services, enabling a variety of useful interaction-aware applications.

Interaction-aware applications: Initial applications are emerging to leverage social contexts during face-to-face interactions. For example, Pentland et al. infers meaningful social relationships by analyzing large volumes of daily social interaction data collected by mobile devices (e.g., Bluetooth scanning). Also, they propose several applications such as Sensible Orb [33] and Meeting Mediator [22] for workplace meeting situations. As a mobile platform, SocioPhone facilitates such applications in real-time.

Speaker recognition and diarization: In the fields of artificial intelligence, there have been significant efforts to infer diverse information from sound signals, including speaker, words, and emotions [2][4][7][8][17][39][40]. However, daily conversation monitoring on mobile devices imposes new requirements such as highly-interactive turns, dynamic acoustic situations, real-time processing, and the resource limitations of the mobile devices.

We may also consider using speaker diarization techniques to extract fine-granule "who spoke when" information [2][4]; example applications include conversation-structure analysis for meeting records or automatic index building on media contents. However, it is difficult to directly apply these techniques in our environments. First, they are designed for post-conversation analysis; they hardly support real-time monitoring, which is the key to enable timely interaction-aware services as in *Tug-of-War* and *SocioTherapist*. For applications based on offline profiling such as *SocioDigest*, diarization techniques might be useful but requires careful consideration. It requires huge power and storage (2.5 GB per day at 16-bit 16kHz PCM) to capture and store raw sound data. Also, the error rates of such techniques needs to be further investigated in daily-interaction situations, as most of the previous studies are based on highly quality-controlled sounds.

Mobile context monitoring systems: Some previous works propose mobile platforms to facilitate monitoring of user contexts on-the-fly [19][20][25][28][29]. Most work focuses on efficiently monitoring personal contexts such as location, activity, and emotion. SocioPhone expands the scope of context-awareness toward daily face-to-face interactions.

9. CONCLUSION

In this paper, we propose the design and implementation of *SocioPhone*, a mobile interaction-monitoring platform. It provides a set of APIs to monitor *turn* and turn-derived meta-linguistic contexts during conversations in progress. In its core, it incorporates highly-efficient online turn-monitoring techniques based on the volume topography collaboratively created by conversation participants' phones. We built several interesting applications on top of SocioPhone: *SocioTherapist*, *SocioDigest*, and *Tug-of-War*. Moreover, we showed that our turn monitoring technique offers significant advantages over comparative techniques in terms of both accuracy and battery usage. We believe SocioPhone is a first crucial step to build a full-fledged mobile platform for daily face-to-face interaction monitoring.

10. ACKNOWLEDGEMENT

We thank our shepherd, Prof. Landon Cox, for his valuable comments to improve the quality of this paper. We also thank Taiwoo Park, Haechan Lee, Yuhwan Kim, and Changhoon Lee, for their great help on building and demonstrating interesting applications, and Dr. Hyojung Shin for his valuable inputs for the project. This work was supported by the NRF of Korea grant

(MEST) (No. 2012-0005733) and the SW Computing R&D Program of KEIT(2012-10041313, UX-oriented Mobile SW Platform) funded by the Ministry of Knowledge Economy of Korea.

11. REFERENCES

[1] Alpaydin, E. *Introduction to Machine Learning, 1st edition*. The MIT Press, 2004.

[2] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. Speaker Diarization: A Review of Recent Research. In *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, issue 2, pp. 356-370. 2012.

[3] Aran, O., and Gatica-Perez, D. Analysis of Group Conversations: Modeling Social Verticality. *Computer Analysis of Human Behavior*, pp. 293-322. 2011. Springer London.

[4] Barras, C., Zhu, X., Meihner, S., and Gauvain, J. Multistage Speaker Diarization of Broadcast News. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, Issue 5. 2006.

[5] Boil, S., Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoustics, Speech, and Signal Processing*. Vol 27, Issue 2, pp. 113-120. 1979.

[6] Brdiczka, O., Maisonnasse, J., and Reignier, P. Automatic Detection of Interaction Groups, In *ICMI*, 2005.

[7] Campbell, J.P., Jr. Speaker recognition: a tutorial. *Proc. of the IEEE*, Vol. 85, Issue 9, pp. 1437-1462. 1997.

[8] Chen, J., Benesty, J., Huang, Y., and Doclo, S. New insights into the noise reduction Wiener filter. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, Issue 4. 2006

[9] Choudhury, T, and Pentland, A. Sensing and Modeling Human Networks using the Sociometer. In *ISWC*, 2003.

[10] Cowley, S. J. Of timing, Turn-Taking and Conversations, *Journal of Psycholinguistic Research*, Vol. 27. Nov. 5. 1998.

[11] Efstratious, C., Leontiadis, I., Picone, M., Rachuri, K. K., Mascolo, C., and Crowcroft, J. Sense and Sensibility in a Pervasive World. In *Pervasive*, 2012.

[12] Enck, W., Gilbert, P., Chun, B., Cox, L. P., Jung, J., McDaniel, P., and Sheth, A. N. TaintDroid: An Information-Flow Tracking System for Realtime Privacy, In *OSDI*, 2010.

[13] Ford, B., Strauss, J., Lesniewski-Lass, C., Rhea, S., Kaashoek, F., and Morris, R. Persistent Personal Names for Globally Connected Mobile Devices. In *OSDI* 2006.

[14] French, N. R. and Steinberg, J. C. Factors Governing the Intellligibility of Speech Sounds. Journal of the Acoustical Society of America, vol. 19, no. 1, pp.90-119. 1947.

[15] Goffman, E. The Interaction Order. *American Sociological Review*, vol. 48, pp. 1-17. 1983.

[16] Hawkins, K. Some Consequences of Deep Interruption in Task-oriented Communication. In *Journal of Language and Social Psychology*, vol. 10, no. 3, pp. 185-203. 1991.

[17] Hung, H., Huang, Y., Friedland, G., Gatica-Perez, D. Estimating Dominance in Multi-Party Meetings Using Speaker Diarization. In *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 4. 2011.

[18] Hwang, I., Jang, H., Nachman, L., and Song, J. Exploring Inter-child Behavioral Relativity in a Shared Social Setting: a Field Study in a Kindergarten. In *UbiComp* 2010.

[19] Ju, Y., Lee, Y., Yu, J., Min, C., Shin, I., and Song, J. SymPhoney: A Coordinated Sensing Flow Execution Engine for Concurrent Mobile Sensing Applications, in *SenSys*, 2012.

[20] Kang, S., Lee, Y., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y., and Song, J. Orchestrator: An Active Resource Orchestration Framework for Mobile Context Monitoring in Sensor-rich Mobile Environments, in *PerCom*, 2010.

[21] Kim, C., and Stern, R. M. Robust Signal-to-Noise Ratio Estimation Based on Waveform Amplitude Distribution Analysis. In *InterSpeech*, 2008.

[22] Kim, T., Chang, A., Holland, L., and Pentland, A. Meeting Mediator: Enhancing Group Collaboration using Sociometric Feedback. In *CSCW*, 2008

[23] Klasnja, P., Consolvo, S., Choudhury, T., Beckwith, R., and Hightower, J. Exploring Privacy Concerns about Personal Sensing. In *Pervasive* 2009.

[24] Koegel, R. L., O'Dell, M. C., and Koegel, L. K. A Natural Language Teaching Paradigm for Nonverbal Autistic Children. *Journal of Autism and Developmental Disorders*, vol. 17, no. 2, pp. 187-200, 1987.

[25] Lee, Y., Iyengar, S. S., Min, C., Ju, Y., Park, T., Lee, J., Rhee, Y., Song, J. MobiCon: Mobile Context Monitoring Platform, in *Communications of ACM* (*CACM*), 2012.

[26] Lee, Y., Ju, Y., Min, C., Kang, S., Hwang, I., and Song, J. CoMon: Cooperative Ambience Monitoring Platform with continuity and benefit awareness. In *Mobisys*, 2012.

[27] Lu, H., Brush, A. J. B., Priyantha, B., Karson, A. K., and Liu, J. SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones. In *Pervasive*, 2011.

[28] Lu, H., Pan, W., Lane, N. D., Choundhury, T., and Campbell, A. T. SoundSense: Scalable Sound Sensing for People-Centric Application on Mobile Phones. In *MobiSys*, 2009.

[29] Lu, H., Yang, J., Liu, Z., Lane, N. D. Choudhury, T., and Campbell, A. T. The Jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, 2010.

[30] Miluzzo, E., Papandrea, M., Lane, N. D., Lu, H., and Campbell, A. T. Pocket, Bag, Hand, etc. – Automatically Detecting Phone Context through Discovery. In *PhoneSense* 2010.

[31] Miluzzo. E., Cornelius, C. T., Ramaswamy, A., Choudhury, T., Liu, Z., Campbell, A. T. Darwin Phones: the Evolution of Sensing and Inference on Mobile Phones. In *MobiSys*, 2011.

[32] Mundy, P., Sigman, M., Ungerer, J. and Sherman, T. Defining the Social Deficits of Autism: The Contribution of Non-verbal Communication Measures. *Journal of Child Psychology and Psychiatry*, vol. 27, no. 5, 1986.

[33] Olguin, D, O., Waber, B. N., Kim, T., Mohan, A., Ara, K., and Pentland, A. Sensible Organizations: Technology and Methodology for Automatically Measuring Organizational Behavior. *In IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 39, Issue 1, pp. 43-55. 2009.

[34] Park, T., Lee, J., Hwang, I., Yoo, C., Nachman, L., and Song, J. E-Gesture: A Collaborative Architecture for Energy-efficient Gesture Recognition with Hand-worn Sensor and Mobile Devices, In *SenSys*, 2011.

[35] Sanchez-Cortes, D., Aran, O., Mast, M. S., and Gatica-Perez, D. Identifying emergent leadership in small groups using nonverbal communicative cues. In *ICMI* 2010.

[36] Sellen, A., and Whittaker, S. Beyond Total Capture: A Constructive Critique of Lifelogging. *Communications of the ACM*, vol. 53, no. 5, pp. 70-77. May 2010.

[37] Sohn, J., Kim, N. and Sung, W. Statistical model-based voice activity detection. *IEEE Signal Processing Letters*, Vol. 6, Issue 1, pp. 1-3. 1999.

[38] Wang, D. and Narayanan, S. S. Robust Speech Rate Estimation for Spontaneous Speech. In *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.15, Issue 8. Pp. 2190-2201. 2007.

[39] Wrigley, S. N., Brown, G. J., Wan, V., and Renals, S. Speech and Crosstalk Detection in Multichannel Audio. *IEEE Transactions on Speech and Audio Processing*, Vol. 13, Issue 1, pp. 84-91. 2005.

[40] Wyatt, D., Choudhury, T., Bilmes, J., and Kitts, J. A. Inferring Colocation and Conversation Networks from Privacy-sensitive Audio with Implications for Computational Social Science. *ACM Trans. Intelligent Systems and Technology*, vol. 2, 2011.