SensePresence: Infrastructure-less Occupancy Detection for Opportunistic Sensing Applications

Md Abdullah Al Hafiz Khan, H M Sajjad Hossain, Nirmalya Roy
Department of Information Systems, University of Maryland Baltimore County
mdkhan1@umbc.edu, riaj.sajjad@umbc.edu, nroy@umbc.edu

Abstract—**Predicting the occupancy related information in an environment has been investigated to satisfy the myriad requirements of various evolving pervasive, ubiquitous, opportunistic and participatory sensing applications. Infrastructure and ambient sensors based techniques have been leveraged largely to determine the occupancy of an environment incurring a significant deployment and retrofitting costs. In this paper, we advocate an infrastructure-less zero-configuration multimodal smartphone sensor-based techniques to detect fine-grained occupancy information. We propose to exploit opportunistically smartphones' acoustic sensors in presence of human conversation and motion sensors in absence of any conversational data. We develop a novel speaker estimation algorithm based on unsupervised clustering of overlapped and non-overlapped conversational data to determine the number of occupants in a crowded environment. We also design a hybrid approach combining acoustic sensing opportunistically with locomotive model to further improve the occupancy detection accuracy. We evaluate our algorithms in different contexts; *conversational*, *silence* and *mixed* in presence of 10 domestic users. Our experimental results on real-life data traces collected from 10 occupants in natural setting show that using this hybrid approach we can achieve approximately 0.76 error count distance for occupancy detection accuracy on average.**

I. Introduction

Smartphone based participatory and citizen sensing applications have attested the promise of microphone sensor based several audio inference applications. The most obvious benefits from microphone sensor based applications are assessment of social interaction and active engagement among a group of people [1], speaker identification and characterization of social settings [2][3][4] by leveraging their conversational contents. Recently speaker counting has been investigated to enumerate the number of people in conversational episodes like social gatherings, interactive lecture sessions or in a restaurant or shopping mall environment [5][6][7]. Most of the recent studies focus on the conversational data to extract the high level occupancy information assuming that all the users are taking turns to speak. This may occur in a controlled environment (seminar, meeting, classroom etc.) but overlapped or concurrent speaking is the most frequently occurred event in our day to day life. On the other hand most of the previous studies are obtrusive which proposed to use arrays of ambient microphone sensors, video cameras or motion sensors for inferring the real time occupancy information [8][9].

Taking turns in conversation is albeit feasible but we move one step further considering a more naturalistic uncontrolled environment where people may spontaneously participate in any conversational phenomenon without any a-priori intuition. While smartphones' microphone sensor-based acoustic sensing approach holds great promises in inferring the number of occupants and promoting scalable infrastructure-less opportunistic sensing but it fails in absence of any conversational data from the surrounding environment. Motivated by this we propose to augment locomotive sensing in absence of any conversational episode with acoustic sensing being considered as a de facto audio inference in our model to precisely synthesize the characteristics of a natural environment and accurately estimate the occupancy related information. In pursuit of these goals we propose an opportunistic collaborative sensing system called, *SensePresense*, which opportunistically exploits both the audio and motion data respectively from smartphones' microphone and accelerometer sensor to infer the number of people present in a gathering.

In *SensePresence*, we opportunistically combine smartphone-based acoustic and motion sensing to determine the number of people in a partially conversational and non-conversational environment. When multiple people are present and a subset or a group of people are conversing, how do we identify who are involved in the conversation, or belonged to a specific conversational clique, and who are not, i.e., opportunistically exploiting smartphones' microphone and motion sensors to infer number of people present therein. Such hybrid sensing approach could potentially furnish fine-grained occupancy profiling for better serving many participatory sensing applications while saving smartphones' battery power by advocating a distributed sensing strategy. In this paper, we propose an adaptive acoustic sensing based linear time people counting algorithm based on the real-life conversational data. Our algorithm follows an unified strategy in presence of both overlapped and non-overlapped conversational data as naturally evolved from a crowded environment. Our proposed algorithm relies on a dynamic length of the audio segmented data compared to a predefined static audio segment length [6]. We investigate a locomotive sensing model and augment it with our proposed acoustic sensing based people counting algorithm to make our system work on extreme modality of either of the data sources, whether it is *acoustic* or *locomotive*.

II. Related Work

Smartphones' microphone sensor has been used extensively to opportunistically analyze audio for context characterization. For example, SpeakerSense [4] performs speaker identification and SoundSense [10] classifies sounds from macro to micro contexts. All these work have often in common the use of supervised learning technique. In contrast, SensePresence's occupancy counting process is entirely unsupervised. The authors of [11] used unsupervised techniques to perform speaker clustering using distances of the feature vectors extracted from different speakers. However this occupant estimation has been done only on telephonic conversational data where our proposed system, SensePresence performs speaker counting without any staged conversational setup. It collects data from natural conversation and performs clustering to infer the number of people. The most closely related research to SensePresence speaker counting is *Crowd++* [6] where counting has been done in a controlled scenario with all the participants speaking actively. [6] used a fixed length audio segment (3 sec) where each segment corresponds to an individual but we performed this audio segmentation dynamically to increase the accuracy of occupancy inference. [6] also classified a few segments as undetermined but our system never discards segments as undetermined which is achieved only through employing dynamic segmentation. Therefore SensePresence tackles a richer problem, where none of the speakers are discarded for handling the computational challenges. *Crowd++* [6] proposed to combine pitch with MFCC to compute the number of people with an average error distance of 1.5 speakers. On the other hand SensePresence improved average error distance by a factor of two (0.76 Speakers). Next we briefly discuss some specific occupancy sensing applications.

Building occupancy monitoring applications rely on the deployment and installation of a bona fide system or device inside the building environment necessitating the high retrofitting and management costs. For example, a wireless sensor network system consisting of PIR sensor, reed switch and CC2530 radio has been deployed for collecting real time occupancy information inside a building environment [12]. Non-intrusive occupancy monitoring algorithm has been proposed to infer binary occupancy from smart meter data in a home environment [13] which helps detect occupancy using average and standard deviation of the power usage and power range. However, ambient and infrastructure sensors have been deployed there to infer occupancy information and neither of them used mobile context data for inferring the number of people in building environment [14]. In this work, we present an opportunistic infrastructure-less zero-configuration hybrid system exploiting the ubiquitous availability of smartphone sensors on the horizon to shift the traditional occupancy monitoring paradigm from an infrastructural device based system to fluid mobile sensing based system.

III. Overall SensePresence Framework

We envision to develop a minimally invasive and low-cost mobile system for counting the number of people present in any environment. We propose an opportunistic collaborative sensing approach which exploits multiple sensors on smartphone - microphone for acoustic sensing and accelerometer for locomotive sensing. Our system as shown in Fig. 1, comprises of two subsystems, one deployed on smartphone and other deployed in server. In the mobile part of our proposed client-server architecture, sensed data both for the acoustic and locomotive sensing are being stored in a *data sink* on the smartphone itself and transferred to the server in a regular interval for triggering the opportunistic sensing among the multiple smartphones and posterior data analysis (sink in Fig 1). Acoustic data from each smartphone is first fed to the filter to collect Acoustic Fingerprints (AFP), of any conversation consisting of content based audio. The AFPs being collected from all the smartphones are sent to the "*Estimate Proximity*" module residing on the server- which helps distinguish the audio signals in vicinity and helps approximate opportunistically the inclusion of a group of smartphones to form single clique. Finally, "*Optimum Node*" module elects the clique leader (most informative smartphone) to record the conversational audio data and notifies the condition of deactivation to the other smartphones from capturing the duplicate audio signal. It also helps in sorting the smartphone list based on their audio signal strength which is eventually utilized by locomotive "*Signature Collection*" module to opportunistically check-on and trigger the accelerometer sensor on the smartphones [15].

The Occupancy Context Model (OCM) which resides on the server-side has two main sub-components: *i) Acoustic Context Model*, and *ii) Locomotive Context Model*. These models together form the inference engine consisting of opportunistic occupancy context module.

A. Acoustic Context Model (ACM)

Our acoustic context model comprises of the following three logical components.

Pre-processing: This is the most trivial phase for acoustic signal processing. This module helps to perform the filtering and select the audio segment length dynamically. It finally helps remove all the noises, silences and produce smooth conversational data which is later passed to the feature extraction module.

Feature Extraction: This module is the main basis for extracting all types of features which is utilized in the speaker estimation module. It has been briefly discussed in section V-C.

Speaker Estimation: This serves as a core processor for occupancy counting. (Details in IV-A).

B. Locomotive Context Model (LCM)

It consists of *i) Signature Collection*, *ii) Feature Extraction*, and *iii) Occupancy Estimation* modules. Signature collection module receives total number of people count from ACM module and the sorted smartphone list from the *optimum module* to opportunistically select microphone sensors. Based on these two inputs, *LCM module* makes decision on which smartphones' sensors are needed for further occupancy estimation. Feature extraction module calculates accelerometer sensor magnitude and feeds that into occupancy estimation module, which helps to infer binary occupancy for each smartphone sensed data and finally helps count the total number of people present in a conversational, silent or mixed environment.

IV. Design Methodology

In this section we describe the details of our SensePresence design framework. We present an acoustic sensing based algorithm for counting the number of people present in a conversing environment (such as group meeting, brain storming session etc).

A. Occupancy Estimation Using Acoustic Signature

In this section, we describe occupancy estimation using our proposed acoustic sensing model. We look into the specific

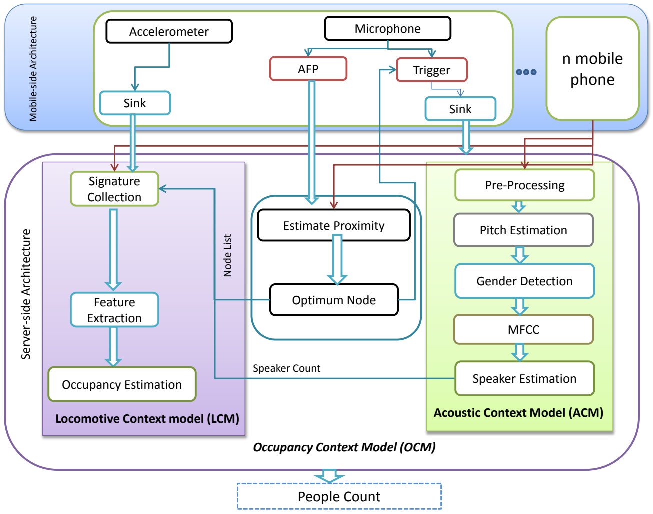

Fig. 1: SensePresence Architectural Overview

cases where all the occupants are conversing. We first attempt to calculate the number of speakers engaged and consider three different phases to compute the number of personnel present. We first propose to create dynamic segments from the supplied raw audio data and assume that each segment belongs to an individual person. We attempt to detect every speaker change point in the entire audio signal spectrum and assign one segment to one person to increase the counting performance of our occupancy detection algorithm. Speaker change point depicts the stopping point of one speaker and starting point of another speaker. Speaker change point detection algorithms have been investigated extensively [16][17][18], however, it is a complex process to detect speaker change point in conversational speech because utterance lengths can be extremely short, speaker changes may occur frequently, there may be some overlaps between the speakers, and surrounding environment can be noisy.

We first calculated confidence score for the entire supplied audio which represents the probability of finding pitch within a segment. We start finding confidence score from a small segment (32 ms) and increase the step size in the next iteration (16 ms), and repeat this iteration for up to 10 seconds audio segment. We calculated the variance of this confidence score and based on a lower variance associated with a specific segment we have selected that segment length as one unit of conversation. If a segment has over 90% confidence, we admit it or otherwise we reject it. As there are many audio segments with different segment lengths, we have chosen a segment length corresponding to a single person unit associated with a higher confidence score and greater number of audio segments with lower segment length. Fig. 2 shows various confidence scores for different segment lengths. We selected 2.72 sec as segment length instead of 3.36 sec when both have a confidence score of 1, but first segment length admits greater number of segments than the latter one. We have calculated this confidence score using YIN [19] algorithm by using non-overlapping frames and skipped the best local estimate step. This help to determine on real time the unit audio segment which solely depends upon the recorded audio.

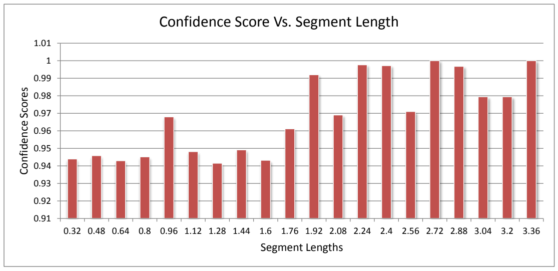

Fig. 2: Confidence Scores for different segment lengths of a sample audio

```
Procedure People-Count (input: set of segments ($S$), total
number of segments($N$); output: number of distinct speakers)
For ($i$ from $1:N$)
    Compute MFCC vectors $m_i$ = Compute_MFCC($S_i$);
    Insert(M,$m_i$);//Insert $m_i$ into MFCC set M
End-For
Sort(M) //sort MFCC set and keep sorted MFCC set
    into the same Set M
PS = {} //Initialize Persons Set
    which contains similar person in sets $PS_j$
For ($i$ from $1:N$)
    For ($j$ from $(i+1):N$)
        angle = Cosine_Similarity($M_i$,$M_j$);
        If (angle < $\theta_{th}$) then
        Insert($PS_i$,$M_j$);
        Else    i=j;    $break$;
        End-If
    End-For
    Insert(PS,$PS_i$); // PS denotes person Set
End-For
$N_S$ = Count_Elements(PS);
return $N_S$;
```

Fig. 3: The People Count Algorithm

As human voice ranges approximately 300 Hz to 4000 Hz, we filter each of the segments based on that frequency range using band pass filter. After filtering the raw audio we have applied Hamming window to reduce the spectral leakage while creating audio segments. Consider a segment which contains m frames and each segment consists of frames $\{F_1,\ F_2,\ .\ .\ .\ ,\ F_m\}$. We calculated MFCC for each frame where each segment has corresponding MFCC feature vectors as $\{M_1, M_2, ..., M_m\}$. We also computed pitch for each segment to apprehend gender in the conversational data. Segment pitches are represented as $\{P_1, P_2, ..., P_m\}$, where the average pitch for male falls between 100 to 146 Hz whereas female pitch is within 188 to 221 Hz, as demonstrated in [20]. Segments which fall within male frequency are marked as male and similarly for female. These two sets are then passed to our proposed people counting heuristic algorithm. Before passing these male and female segments for checking similarity measures, we calculated intra cosine angle of each segment to sort out both male and female segments. Next we have checked the similarity among inter-segments if it falls within our predefined threshold, θ_{th} or not. If these segments have been similar, we have merged them to make a new segment and continued to check for the next segment with this newly created segment. If those segments have been dissimilar then we have moved forward and picked another segment to check similarity with the next one. The pseudo code of our proposed people counting heuristic has been shown in Fig. 3.

B. Occupancy Estimation Using Accelerometer Signature

In this section, we discuss our locomotive sensing model in absence of any conversational data or in a mixed environment where a group of people may talk and other listen silently. If a smartphone is stationary for a significant amount of time, on-board accelerometer sensor produces steady state signature which has no variation or spikes in terms of signal amplitude, whereas if there is a movement it generates a spike or corresponds to a steady-state signal alteration. To detect this abrupt changes in locomotive signal amplitude we propose to use change point detection based technique [21].

Change point detection helps find the abrupt variation in the movement data stream. Our motivation in this work is to use change point to find the stray movements by finding abrupt changes in the accelerometer signals. These changes help inferring binary people counting (whether people present or not). We developed and used offline Baysian changepoint [21] detection based algorithm for inferring occupant's presence in $\mathcal{O}(n^2)$. It has three fold methods. First, we calculate a-priori probability of two successive change points at a distance d (run length). We use Gaussian based log-likelihood model [22] to compute log-likelihood of the data in a sequences $[s, d]$, where no change point has been detected. Second, we calculate log-likelihood for the entire signal $S[t, n]$, log-likelihood of data sequence $S_s[t, s]$ where no changepoint has been occurred between t and s and $\pi[i, t]$, the log-likelihood that the i-th changepoint occurs at time step t. Finally, We calculate the probability of a changepoint at time step t by summing up the log-likelihoods for that sequence. Fig. 4 shows the change-points and their probabilities as being detected successfully. We filter those changepoints based on empirically determined threshold probability (δ_{th}) and infer presence of the occupant based on the admitted changepoint sequence. We also count the number of changepoints in the the data sequence which indicates movement score that represents how frequent the person moves. The overall algorithm has been summarized in Fig. 5.

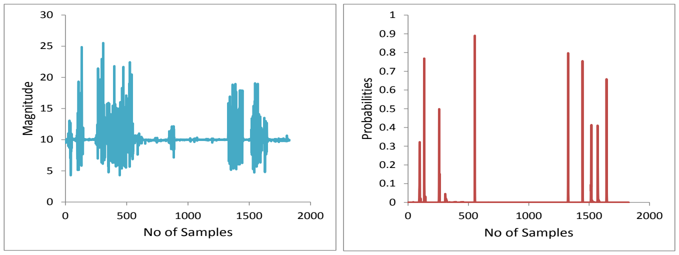

Fig. 4: Magnitude of accelerometer signal (Left) and changepoints with probabilities of that signal (Right) due to a person's random movement patterns

V. System Implementation and Evaluation Results

We now discuss the detailed implementation and evaluation of our SensePresence framework.

A. Tools and Resources

We used Google Nexus-5 with built in microphone and three axes accelerometer sensor for our experiments. Our entire system comprises of two parts: *i)* sensing, and *ii)* classification and clustering, first one was implemented on Nexus-5 and latter on the server. Application software was written in Java which utilizes Android Programming Interface (API) to sense microphone and accelerometer signals. Classification and clustering algorithms and our occupancy counting algorithm have been implemented on the server using python.

```
Procedure Binary-occupancy-detection (input: samples ($data$),
total number of data points($n$); output: 1 for occupant
present, otherwise 0)
For ($t$ from $1:n$)
    $g[i]$ = log$(1/(n+1))$;
    If i == 0 then G[i] = g[i];
    Else G[i] = log(exp(G[i-1])+exp(g[i]));
    End-If
End-For
P[n-1, n-1] = Gaussian_log_likelihood(data, n-1, n)
For ($t$ from $n:1$)
    /* get next changepoint probability by computings
    joint distribution $P(r_n, x_{1:n})$, recursively using
    $\sum_{r_{n-1}} \pi(r_n|r_{n-1})\pi(x_n|r_{n-1}, x_{1:n})\pi(r_{n-1}, x_{1:n-1})$ */
    prob_next_changepoint = Cal_Joint_Dist(data,t,n-1)
    P[t, n-1] = Gaussian_log_likelihood(data, t, n)
    Q[t] = log(exp(P_next_run),
    exp(P[t, n-1] + 1 - exp(G[n-1-t])));
End-For
For ($i$ from $1:n-1$)
     changepoint_prob[0, t] = (P[0, i] + Q[i + 1] +
    g[i] - Q[0]);
End-for
num_effective_cp = 0;
occupancy = 0;
For ($i$ from $1:n-1$)
    For ($t$ from $i:n-1$)
        tmp_sum = (changepoint_prob[i-1, i-1:t]
        + P[i:t+1, t] + Q[t + 1] + g[0:t-i+1]
        - Q[i:t+1]);
        changepoint_prob[i, t] = log(sum(exp(tmp_sum)))
        If (changepoint_prob[i, t] > $\delta_{th}$) then
        num_effective_cp = num_effective_cp + 1;
        End-If
End-For
End-For
If num_effective_cp > 0 then occupancy = 1;
End-If
return $occupancy$;
```

Fig. 5: The Binary Occupancy Detection Algorithm

B. Data Collection

We implemented the acoustic sensing and collected conversational data from different places at different times in naturalistic settings. Conversational data have been collected and properly anonymized during the spontaneous lab conversation among the students (without making the occupants aware of it), lab meeting, and general discussion in the lobby/corridor in presence of a variety of surrounding noise levels. The demographic for our conversational data collection was 1-10 persons (with 5 females and 5 males) in age group of 18-50 years. The acoustic data were collected at a mono sampling rate of 16kHz at 16bit pulse-code modulation (PCM).

C. Acoustic and Locomotive Feature Extraction

We discuss different features relevant to our acoustic and locomotive sensing techniques in this section.

Acoustic Features: We generated two basic features which are used in the speaker identification - MFCC and Pitch. Each feature has been described in details in the following. *i) MFCC* is one of the most significant features which is used for acoustic processing. We followed the following steps to process it. 1. Take the Fourier transform of (a windowed excerpt of) a signal, 2. Map the powers of the spectrum obtained above onto the Mel scale using triangular overlapping windows, 3. Take the logs of the powers at each of the Mel frequencies, 4. Finally, take the discrete cosine transform of the list of Mel log powers. We excluded the first co-efficient of MFCC and then chose 20 coefficients as feature vectors. *ii) Pitch* is defined as the lowest frequency of a periodic waveform. It is the discriminative features between man and woman. Human voice pitch interval falls within the range 50Hz to 450Hz [20]. We calculated pitch of different segments using YIN [19] algorithm. We used 32 msec hamming window with 50% overlap for computing the Pitch and MFCC feature.

Locomotive Features: We calculated the magnitude of accelerometer data. We considered magnitude in order to mitigate calibration.

D. Accuracy Metrics Definition

To evaluate and compare the performance of our system, we computed the average error count as the normalized predicted occupancy metric represented by $\frac{|EC-AC|}{N}$, where EC, AC, N denote the estimated people count, actual people count and number of samples respectively. We presented only the absolute value in order to avoid any positive or negative contribution.

E. Occupancy Counting Results

We evaluated our opportunistic occupancy counting algorithm in four scenarios. *i)* No conversation among occupants, *ii)* All occupants are conversing in a single clique, *iii)* Occupants are conversing in multiple cliques, and *iv)* Mixed conversing and non-conversing occupants.

For the first scenario, when no occupants are involved in conversation we used the accelerometer to count the occupancy. Each accelerometer sensor provides binary occupancy indication based on our change point detection algorithm as discussed in section IV-B which computes the total number of people present in the environment. Fig. 6 shows the total number of people successfully counted using our locomotive sensing model. We note that our locomotive sensing model achieves 80% accuracy (8 out of 10 people) in predicting occupancy when most of the users carry their smartphones with them.

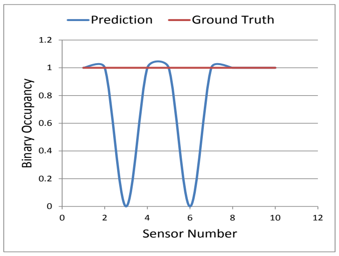

Fig. 6: Locomotive Sensing-based Occupancy Count

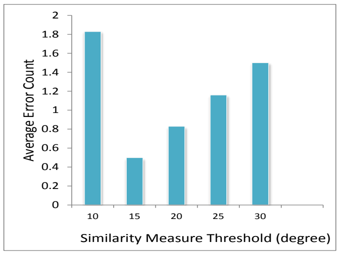

Fig. 7: People Counting performance with different cosine measures

Our opportunistic sensing system plays a critical role when all occupants have been conversing in a single clique. Our system helps activate a single microphone for occupancy counting and deactivate all other microphone and accelerometer sensors based on the server feedback (details are omitted due to space constraints). Fig. 7 depicts the effect of cosine distant similarity measures on our occupancy counting algorithm as shown in Fig 3. We notice that similarity distance angle measures (in degree) play a pivotal role on reducing the error count of occupancy inference. In our experiments with 3 people conversing, we found that 15 degree similarity measure threshold is an appropriate choice for consideration to reduce the error count for our proposed adaptive people counting algorithm.

We also have run experiments in an uncontrolled environment (completely in a natural setting) without imposing any restrictions on smartphones relative positions and distance from each other or from the server. Fig. 8 reports the average error count distance ≈ 0.5 with respect to different positions of the phone. It is noted that when smartphone is placed on the table and two persons speak the error count becomes zero, but when three persons start speaking, error count tends to become slightly higher due to the ambient noise and overlapped conversation.

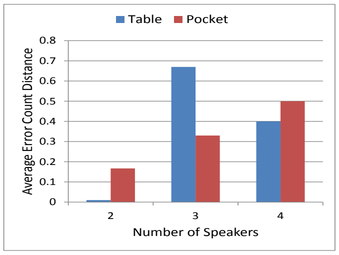

Fig. 8: Occupancy count over different phone positions

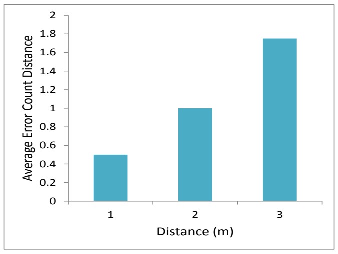

Fig. 9: People counting depends on phone distance

Fig. 9 depicts that error count increases as single clique leader's distance from other occupants increases. We note that for a 3 meter distance error count becomes close to two which confirms that even for a large internal distance separation among the conversing occupants our acoustic sensing model performs quite well.

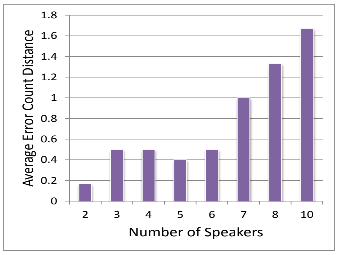

Fig. 10: Accuracy vs. Number of People

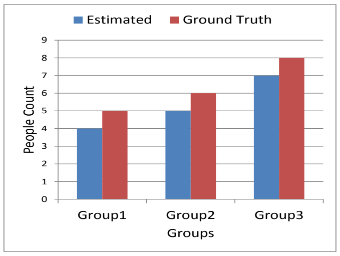

Fig. 11: People Counting vs. Multiple Co-located Group of Speakers

Fig. 10 presents the performance of our people counting algorithm where users speak naturally with overlapped conversations. It is observed that average error count is 0.1 for 2 people and 1.7 for 10 people when conversing together. Thus the overall average error count is 0.76 with number of users present varying from 2 to 10 establishes that our acoustic-based occupancy counting algorithm performs well even in a crowded environment.

In our third scenario, where occupants are conversing in multiple cliques (assume three cliques in our experiment) we deployed three microphones and accelerometer sensors which are chosen based on the proximity measure from the server to infer the occupancy. Fig. 11 shows the intra-group count in presence of conversational data with distinct clique formation. In our experiments, first group has 5 occupants (2 men and 3 women), second group has 6 occupants (3 men and 3 women) and last group has 8 occupants (4 men and 4 women). We observe that the mean error count is ≈ 1 for even our group based acoustic sensing model which attests the promise of our occupancy detection model in different real life scenarios.

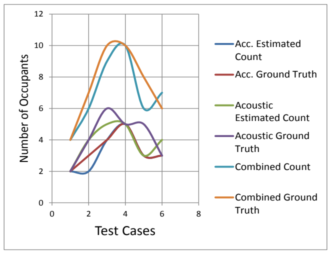

Fig. 12: Locomotive Augmented Acoustic Occupancy Count

Number of Speakers	Crowd++ (Error Count)	Sense Presence (Error Count)
2	0.5	0.167
4	2.33	0.5
6	2.5	0.83
Average	1.78	0.5

TABLE I: Comparison (Average Errro Count) between Crowd++ and SensePresence

In our last scenario, where some people speak and some people remain silent arise challenges for estimating total number of occupants present in that specific place. In this case, we propose to utilize our hybrid locomotive cum acoustic sensing model to infer total number of occupants. For example, consider a scenario where six persons are involved in conversation while four remain silent. For conversing population, we activate either a single microphone sensor if there is a single clique or multiple microphone sensors if there are multiple conversing cliques as determined by our "*Estimate Proximity*" module implemented on the server. We use mean error count estimation to infer the number of people conversing. To estimate the number of people who are not involved in that conversation, we utilize our locomotive sensing model which postulates binary occupancy using change point detection applied on the accelerometer's signal and finally infers the total number of silent people. Fig 12 plots overall occupancy counting performance based on our hybrid approach. For example, when there are ten people and 6 persons converse in a single clique and 4 persons remain silent, our acoustic sensing estimates 5 people out of 6 and locomotive sensing estimates 4 people out of 4, resulting in total of predicting 9 people out of 10. We have compared the performance of our SensePresence system with Crowd++ framework [6] for counting the number of people. Table I shows that the average error count distance for Crowd++ is 1.78 where as for *SensePresence* it is 0.5, more than a three fold increase in accuracy for inferring the total number of people.

Acknowledgement

This work is supported partially by the NSF Award #1344990, and Constellation E^2: Energy to Educate Grant.

VI. Conclusions

In this paper, we present SensePresence, an innovative system to infer number of people present in a specific location. We exploit opportunistically the smartphone based accelerometer and microphone sensor for people counting. We propose an acoustic sensing based unsupervised clustering algorithm addressing the underpinning challenges evolving from naturalistic overlapped and sequential conversation to infer the occupancy of an environment. We posit a change point detection based locomotive sensing model to infer the number of people in absence of any conversational episode. We implement an opportunistic context-aware client-server based architecture to leverage smartphones' microphone and accelerometer sensors and combine our acoustic sensing model with locomotive to better predict the people counting. Our experimental results hold promises in a variety of natural settings with an average error count distance of 0.76 in presence of 10 users. We will explore if additional modality of smartphone based sensors such as location information from magnetometer, or barometer sensor could help improve the accuracy of our proposed occupancy detection system.

References

[1] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, 1993.

[2] T. Choudhury and A. Pentland, "Sensing and modeling human networks using the sociometer."

[3] D. Jayagopi and et al., "Modeling dominance in group conversations using nonverbal activity cues," *In Proc. of IEEE TASLP (2009).*

[4] H. Lu, A. J. B. Brush, and et al., "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones," in *Proc. of PerCom (2011).*

[5] R. Sen, Y. Lee, and et al., "Grumon: Fast and accurate group monitoring for heterogeneous urban spaces," in *Proc. of SenSys (2014).*

[6] C. Xu, S. Li, and et al., "Crowd++: Unsupervised speaker count with smartphones," in *Proc. of UbiComp (2013).*

[7] Y. Lee, C. Min, and et al., "Sociophone: Everyday face-to-face interaction monitoring platform using multi-phone sensor fusion," in *Proc. of MobiSys (2013).*

[8] E. Hailemariam and et al., "Real-time occupancy detection using decision trees with multiple sensor types," in *Proc. of SimAUD (2011).*

[9] R. Tomastik and et al., "Model-based real-time estimation ofbuilding occupancy during emergency egress," in *Pedestrian and Evacuation Dynamics 2008.*

[10] H. Lu, W. Pan, and et al., "Soundsense: Scalable sound sensing for people-centric applications on mobile phones," in *Proc. of MobiSys (2009).*

[11] A. N. Iyer, U. O. Ofoegbu, R. E. Yantorno, and B. Y. Smolenski, "Blind Speaker Clustering," in *Proc. of IEEE ISPACS (2006).*

[12] Y. Agarwal, B. Balaji, and et al., "Duty-cycling buildings aggressively: The next frontier," in *In Proc. of IPSN (2011)*, 2011.

[13] D. Chen, S. Barker, and et al., "Non-intrusive occupancy monitoring using smart meters," in *Proc. of BuildSys (2013).*

[14] W. Kleiminger, C. Beckel, and S. Santini, "Opportunistic sensing for efficient energy usage in private households."

[15] L. A. Castro, J. Favela, and et al., "Collaborative opportunistic sensing with mobile phones," in *Proc. of ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (2014).*

[16] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *In Proc. of IEEE Signal Processing Letters (2004).*

[17] D. Liu and F. Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Proc. of EuroSpeech (2009).*

[18] L. Lu and H.-J. Zhang, "Real-time unsupervised speaker change detection," in *Proc. of IEEE Pattern Recognition (2002).*

[19] A. de Cheveignéb and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am (2002).*

[20] R. J. Baken and R. F. Orlikoff, *Clinical measurement of speech and voice.* Cengage Learning (2000).

[21] P. Fearnhead, “Exact and efficient bayesian inference for multiple changepoint problems,” in *Proc. of Statistics and computing (2006)*.

[22] K. M. Xuan Xiang, “Modeling changing dependency structure in multivariate time series,” in *Proc. of ICML (2007)*.