O-MUSUBI: Ad-hoc Grouping System Enhanced by Ambient Sound – The Similarity based on Information Theoretical Features for Sound-Fields –

Sachio Teramoto, Jun Noda
Cloud System Research Laboratories
NEC Corporation
Kanagawa, Japan
Email: s-teramoto@bx.jp.nec.com, j-noda@cw.jp.nec.com

***Abstract*—The aim of this paper is to achieve ad-hoc grouping systems enhanced by ambient sounds or sound-fields. As an elemental technology of ad-hoc grouping, systems have to be equipped with a search engine with sufficient accuracy to find out users who are in similar contexts. Systems require another similarity criterion for sound-fields. Because search results from well-known similarities, such as cosine-similarity, cannot exclude false negative and cannot restrict false positives. Moreover, in order to cover a wide-variety of mobile devices including smartphones, we have the problem of the deterioration of search accuracy due to differences in microphone performances. We may also have to decrease the system-wide load. This suggests that original sound-field data should be resized as small as possible without losing valuable features to flexibly recognize different contexts. We thus propose a new similarity criterion on sound-fields for ad-hoc grouping. We also show experimental results to ensure all requirements are fulfilled.**

***Keywords*-sensor based system; ubiquitous system; ad-hoc communication; ambient-sound.**

I. INTRODUCTION

To increase the chance of grouping anytime anywhere with the growing popularity of social networking services (SNSs), constituent members for sharing pictures or communicating with each other choose to use various kinds of SNSs. Indeed, ad-hoc group communication services [1] are just beginning to be provided for smartphone users who want to temporarily set up a group consisting of ones immediate circle over a period of time.

Users who want to find constituents to add to their group automatically have usability and operability requirements. To infer constituents appropriately, it is important to guarantee high accuracy for searching for ones who have the same situation or context. We thus concentrate search accuracies on *the false negative exclusion,* and *the restriction of a false positive.* Here we consider a situation in which group constituents have been identified and the members do not want others to join in their group. A false negative is when the search could not find suitable members. A false positive is when the search finds unsuitable members. As another requirements supposed in various situations, a wide area must be covered. This cause that systems should have sensors equipped by mobile devices as their components.

There is a fundamental approach to measure proximity of user's context, such Wellman et al.'s approach [2] for acquiring and using absolute positioning of GPS that gives latitude and longitude. However, approaches based on GPS cannot be applied indoors or underground. Thus, other approaches that apply several sensors have been investigated; for example, Cricket [3] and ActiveBat [4] based on ultrasonic waves, Ekahau [5] based on wireless LAN, and LuxTrace [6] based on building illumination. However, even if these approaches were combined with GPS, it still might be difficult to apply to ad-hoc grouping systems, since we have to arrange many sensors broadly. Thus, equipment costs (deployment cost and maintenance cost) are comparatively high. Moreover, an approach may be also desired that is applicable even in places or situations in which sensors are difficult to deploy.

On the other hand, there exist approaches on positioning inferred by comparing each pair of sensor data. We refer such approaches to certification matching for descriptive purpose. Some certification might be created simply by an action occurring that can be sensed by devices: vibrating devices that enable an accelerometer [7], or clicking the same button simultaneously. The certification based on occurrence has a weakness in terms of search accuracy. The main factor for its deterioration comes from increasing probabilities of collisions occurring; users who are in different contexts doing the same action at the same time. When an ad-hoc grouping system has about 10,000 users exist, we can confirm theoretically that probabilities approach infinitely to 1, by analogies of the birthday's paradox.

In this paper, we resolve subjects mentioned above by introducing a new similarity. Specifically, the proposed similarity measures information theoretical features in ambient sound or sound-fields. The sound-fields can be sensed by a microphone equipped on many mobile devices, including smartphones. Therefore, by considering microphones equipped on many mobile devices, the requirements for covering a wide area and various scenes and equipment costs are simultaneously solvable. Moreover, utilizing sound-fields may be tractable for search-accuracy requirements related to the restriction of false positives because sound-fields have

 ISBN: 978-1-61208-246-2

many features and variations, rather than simple certifications based on occurrence of actions.

The rest of this paper is organized as follows. In Section II, we deal with related works for context proximity inference methods based on ambient sound and describe requirements for similarity between sound-fields. In Section III, we overview the architecture of ad-hoc grouping system enhanced by ambient sound and describe in detail procedures of each mobile device and the cloud server. In Section IV, we report experimental results on the accuracy of information retrieval for the proposed similarity and discuss advances fulfilling each requirement described in Section II. Finally, in Section V, we summarize the results that have been achieved and detail future works.

II. Context proximity inferences based on sound-fields

A. Related works and their problems

To infer proximity of contexts on the basis of a sound-field, Sturm et al. [8] proposed an approach for recognizing trajectories of several moving sound-sources by using microphone-arrays. However, this also requires that many microphones be deployed to cover a wide area.

As methods for no equipment costs, Tarzia et al. [9] proposed that a positioning system for single user, based on finger-prints of a sound-field. This system can allow to recognize rooms where user visited. However, it is not so easy to get a high accuracy on location retrieval. We discuss on accuracies in section IV.

Lu et al. [10] proposed a method inferring user's context, using ambient sounds. This method leverages machine learning on several features of sound-field, to classify ambient sounds into attributive categories with high accuracies. However, Lu et al. did not indicate whether their method have a capability to distinguish sound-fields in a same category.

Nakamura et al. [11] provide thorough knowledge of specific sound-fields, especially conversation-fields. They designed architectures that recognize appropriately different conversation-fields, using cosine-similarity.

However, to cover various sound-fields, cosine-similarity has a weakness in terms of search accuracies: higher *precision* or lower *false-positive.* The precision and the false-positive are defined as follows. Let R be the number of users who are constituent members by just grouping and N the number of users found out by searching. Thus, precision $p := \frac{R}{N}$; false-positive $f := \frac{N-R}{N}$. We also have a relation $p = 1 - f$. Thus we consider only the false-positive in this paper.

Here, we describe a number of disadvantages derived from measuring with cosine-similarity for sound-fields. First, search results by using cosine-similarity may contain a certain amount of false-positives. Sound-fields have mainly two discriminable sounds; *event-sounds*, which occur in an unexpected fashion, and *ambient-noise*, which are stationary background sounds. Note that a definition of ambient-noise includes sounds in which one can be observed anywhere else, such as cafeterias, offices, and also calm places though contradicting this term with noise. Comparing duration between an event-sound with a ambient-noise, ambient-noise occupy a considerable amount of sensing time, but an event-sound is rare. This implies that temporal coincidences of ambient-noise severely affect context proximity, but coincidences of event-sounds do not. This is because cosine-similarity treats ambient-noise and event-sounds evenly. Therefore, results of searching with the cosine-similarity tend to have higher false-positives rate, since results contain many false users who only are in similar ambient-noise. Therefore, similarities measuring sound-fields should give higher grades for temporal coincidence with event-sounds, but not for ambient-noise.

Second, there is a vulnerability in differences of microphone performance. Usually, according to the type of mobile devices, microphones' performances differ dramatically from each other. Differences are especially observable in sound-pressure levels. That is, cosine-similarity will misjudge users who have distinct contexts, even if both microphones sense just the same sounds. Since such unfortunate cases will be caused by cosine-similarity, inferring proximity of context will err frequently or be unable to except the false-negativeness.

Next, we consider a practical system for ad-hoc grouping enhanced by a sound-field. Then problems on system-width load emerge. Since the data size of any sound-pressure series sensed by mobile devices is not very small, data sent from each mobile device should be as small as possible. In particular, a sound-pressure series contains much more information or many more various features than required. Thus, it is inefficient from the viewpoint of both of system-wide loads and network communication costs. Specifically, receiving rather large size of data will cause high network I/O loads, as a consequence, restrict availabilities of systems. Thus, the communication cost become a bottleneck. This is a problem that users could not join a group within applicable timings. Therefore, for the sound-pressure series, a contraction method is required that has valuable features to recognize different contexts.

B. Requirements and technical idea

We describe requirements derived from problems in the previous section, and its technical idea.

i) **Higher accuracies on information retrieval**
Proposed similarity estimates temporal coincidences between singular value (derived from event-sounds) included in each sound-fields from aspects of information theoretical features. In particular, we introduce concepts of mutual information. That is, if event-sounds, which have practically lower probabilities of occurring in sound-fields, coincide, then we add higher

 ISBN: 978-1-61208-246-2

estimation to their similarity. Furthermore, we can confine contributions of ambient-noise to similarities, and then the false-positive ratio tends to decrease. Therefore, we could guarantee higher accuracy in information retrieval, and thus, could appropriately associate users with others in similar environments.

ii) **Flexible treatments for different microphones**
Performance differences are observed between equipped microphones, especially in small and large of sound-pressure values. To resolve the differences in sound-pressure values, we generate a collection of multiple *feature vectors* extracted from frequency spectrum, considering redundancy for sound-pressure values. Then, we compute information entropy, estimating coincidences between one collection and the others. This implies that proposed similarity is relatively tolerant of differences in microphones performance. Therefore, we could accept a wide variety of mobile devices, since proposed similarity may resolve flexibly.

iii) **Low communication costs**
Now we consider "large scale and real-time" ad-hoc grouping (cloud) systems enhanced by ambient sound. In such cloud systems, network communication costs may be problem for system availability, since the data size of any sound-pressure series is not very small. This also implies higher system loads. Considering an availability of a system, we also have to avoid that network I/O loads will become a bottleneck. Thus, to reduce communication costs, we apply a contraction procedure to sound-pressure series sensed originally by microphones. Note that applied contraction procedures also have to be guaranteed to meet the above two requirements simultaneously. To tackle this issue, we apply a low-pass filter like a finite impulse response (FIR). Applying a FIR filter, a sound-pressure series is shortened and consists of components only with low frequency bands. We refer any sound-pressure series applied FIR as series of beats, and show that any series of beats still holds enough features by experimental results in section IV.

III. O-MUSUBI: AD-HOC GROUPING SYSTEM ENHANCED BY AMBIENT SOUND

We show overview of proposed ad-hoc grouping system enhanced by ambient sound in Figure 1. We entitle this system O-MUSUBI, which is acronym stands for Organization scheme Measured by Universal **S**ensor data like ambient sound for UBIquitous machines. Another meaning is derived from combination of two Japanese words, "Oto" (sound in English) and "Musubi" (connection or nexus in English). cf. O-MUSUBI is Japanese traditional riceball.

Each mobile device sends sensing data consisting of a sound-pressure series to the cloud server. Then the cloud server computes similarity of each pair of given sound-pressure series.

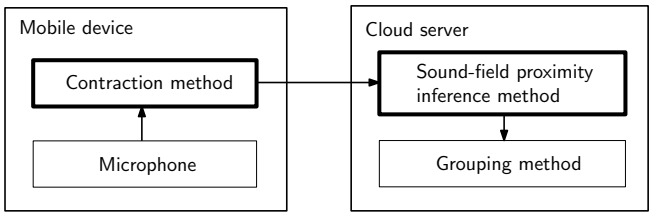

Figure 1. Overview of O-MUSUBI system.

The overview of procedures in each mobile device can be described as follows. The contraction method contracts original sound-pressure series by removing unnecessary information. Then each mobile device sends contracted data to the cloud server. We assume that sensing time is separated by a unit time (e.g. three seconds) that is determined and shared in the whole system. We also assume that the unit time is continuously resumed until a group is found or a termination message is received.

The main procedure in the cloud server is to infer each pair of users' context proximity. More specifically, sound-field proximity inference method computes similarities, or information entropies, between two of each sound-pressure series. Then, grouping method update information for groups and may notify users of new groups they should join.

A. The mobile devices side

The procedure of mobile devices mainly consists of three phases: sensing sound-pressure series, contracting the original series, and sending the contracted series to the cloud server. Here, we have known that there exist temporal gaps between any pair of sound-pressure series. Although feature vectors are created in the basis on frequency spectrum, we cannot correct the gaps by using any pair of spectrum component series. This is because spectrum component series does not have temporal information. Thus, we have to correct or synchronize at the cloud server side. This temporal synchronization procedure have been described at sub-section III-B. This is also desirable to cover various kinds of devices whose computational resources are poor.

In this section, we describe the procedure of the contraction method. To contract the original data, we borrow an idea from applying low-pass filers such as FIR. Although applying filters might loss much information contained in the originals, valuable features are still retained, such as beats. This series of beats implicitly provides occurrences of tempos of event-sounds, and coincidences of tempos with spectrum powers, or joint probability, have higher information entropies. Thus, we could ensure the accuracy of information retrieval, in spite of considerably shrinking the original sound-pressure series.

 ISBN: 978-1-61208-246-2

Algorithm 1 shows a pseudo-code of contraction method. For input, Algorithm 1 is given an array `buf` stored sound-pressure series that is sensed with a sampling frequency specified by parameter `samplingFrequency` for a given length of time specified by parameter `duration`. For output, Algorithm 1 ensures an array `contractData` stored contracted sound-pressure series whose elements are maximal ones in `buf` for each sliding time-window. Additionally, we would assume that two consecutive time-windows share overwrap each other. We specify a fraction of overwrap with `overwrapRate`, where $0 \leq$ `overwrapRate` < 1.

We note that information that should be shared by both mobile devices and the cloud server is represented by two parameters: the sensing time `duration` in mobile devices and the size of the sound-pressure series `sendDataSize`.

Algorithm 1: Contraction Procedure in Mobile Device

input : An array `buf` stored the time series-data of sound pressure values.
output: A contracted array `contractedData`

1 $w = \left\lfloor \frac{\texttt{buf}.length}{(1-\texttt{overwrapRate})\times\texttt{sendDataSize}+\texttt{overwrapRate}} \right\rfloor$;
2 $w' = w \times$ `overwrapRate`;
3 **for** $i = 1, k = 0;\ i <$ `buf`$.length;\ i{+}{+}, k{+}{+}$ **do**
4 **if** `buf`$[0] <$ `buf`$[i]$ **then**
5 `buf`$[0] =$ `buf`$[i]$;
6 **if** $k{=}{=}w$ **then**
7 `contractedData`$.push\_buck($`buf`$[0])$;
8 $k = 0; \quad i \mathrel{-}= w'$;
9 `buf`$[0] =$ `buf`$[i]$;
10 **return** contractedData;

In the case of a smartphone, the sampling frequency is configured by its specification and is sensed with at least 8 kHz. For example, when any smartphone sensing for 3 seconds, and the system configures `sendDataSize` $= 300$ and `overwrapRate` $= 0.5$, the time-window size `windowSize` becomes 162, and each element stored in output array `contractData` is a maximal among 162 values in each sliding time-window.

B. The cloud server side

In this section, we describe the procedure in the cloud server on the basis of requests from the first two requirements described in Section II-B.

In the cloud server, the sound-field proximity inference method computes a degree of similarity between any pair of sound-fields, and then, in accordance with similarities, the grouping method updates grouping information to create a new group or find a group when one user can find other users having higher similarities.

Algorithm 2 shows a pseudo-code of sound-field inference method. The procedures in the cloud server consist of two main steps: executing synchronizations in chronological order between sound-pressure series, and for each sliding time-windows, generating feature vectors and computing information entropies.

Algorithm 2: Computation of Information Entropy on Sound-Fields

input : A pair of two sound-pressure series $\{s_0, s_1\}$.
output: The similarity measured between s_0 and s_1.

1 TimeSynchronous(s_0, s_1);
2 $w = (1 -$ `FFToverwrap`$) \times$ `FFTwinSize`;
3 **for** $t = 0; t < s_1.length -$ `FFTwinSize`$; t{+}{=}w$ **do**
4 **for** $i = 0; i \leq 1; i{+}{+}$ **do**
5 $S_i =$ FFT$(s_i, t,$ `FFTwinSize`$)$;
6 $V_i =$ SpectrumQuantization(S_i);
7 CommonVectorAggregation(H, V_0, V_1);
8 **foreach** $v \in H$ **do**
9 $p_v = \frac{H[v]}{H.count}$;
10 `entropy`$+{=} p_v \log(p_v)$;
11 **return** $|$`entropy`$| \times H.count$;

Executing synchronization in chronological order: The procedure TimeSynchronous, Algorithm 2 (line 1), corrects small gaps for a given two sound-pressure series. Here, we describe an algorithm that corrects gaps in time-series as follows. First, we generate two arrays M_0 and M_1 consisting of maximal values for each given sound-pressure series s_0 and s_1. We define maximal as being the maximum among three for just previous and next ones, and itself, and their differences are larger than a threshold. Note that any sound-pressure series may have a number of maximal. Second, we obtain moving factors (gap size and direction.) For example, part of the procedure is as follows: for each maximal x_i in M_0, finds $y_i \in M_1$ which is the nearest to x_i; then, memorizes the minimum of $|x_i - y_i|$ and its sign of $x_i - y_i$ as the moving factor. Finally, temporal synchronization is executed on the basis of the moving factor.

Generating feature vectors and computing entropies: Here we describe the procedure of the first for-loop shown in Algorithm 2 (lines 3–7). Given two sound-pressure series s_0 and s_1 (which may be applied a synchronization), the procedure computes average information entropy and its summation for each sliding time-window whose size is specified by parameter `FFTwinSize`. We assume that `FFTwinSize` is the power of 2, since `FFTwinSize` corresponds to data size input towards fast Fourier transform (FFT). Two consecutive time-windows should share an overwrap, to prevent missing features such as event-sounds occurring at a boundary of non-overwrap consecutive time-windows. Let

 ISBN: 978-1-61208-246-2

FFToverwrap be the parameter specifying the fraction of overwrap, or overwrap ratio, for two consecutive time-windows, where $0 \leq$ FFToverwrap < 1. As described in line 2 of Algorithm 2, for example, when we configure the size of a time-window FFTwinSize with 64 and the overwrap ratio FFToverwrap with $\frac{1}{8}$, the actual size w of overwrap is equal to $56 = (1 - 0.125) \times 64$.

We generate feature vectors for each inputs s_0 and s_1 respectively, as shown in lines 3–7. The process of generating feature vectors consists of two steps as follows. First, for each s_0 and s_1, we obtain frequency spectrum S_0 and S_1, respectively, by FFT. Second, we generate a collection of multiplied feature vectors considering a redundancy by SpectrumQuantization. The domain of any feature vectors is defined by two parameters: the cut-off frequency cutOffFreq, which defines the upper bound on frequency we use, and the number of quantization levels quantLevel. For example, when cutOffFreq $= 11$ and quantLevel $= 4$, an arbitrary feature vector is defined on a finite space $[1, 10] \times [1, 4]$. Figure 2 shows a simple example.

Here, we refer to the relationship between sendDataSize of mobile devices with cutOffFreq of the cloud server. That is, cutOffFreq gives a lower bound of sendDataSize. When we generate a frequency spectrum with lower frequency than cutOffFreq by FFT, we have to hold a condition on value of parameters at least sendDataSize ≥ 2 cutOffFreq. We also note the tradeoff between the size of data sent and the strength of information entropies computed from given data. This is because the number of time-windows defined in given data increases if the size of data sent is larger.

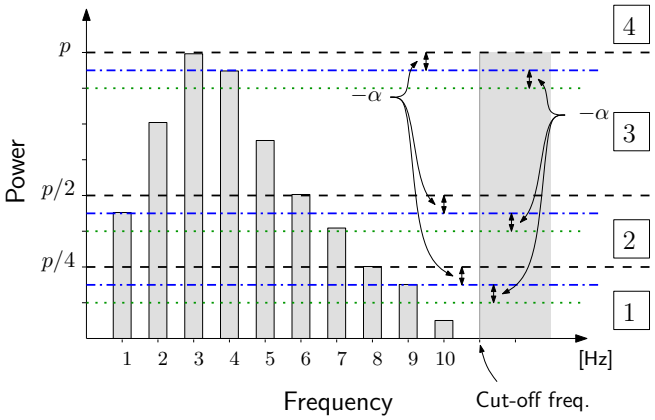

Figure 2. The characteristic-vector generation based on quantization frequency spectrum.

We describe the procedure of quantization of each frequency component in a frequency spectrum. In this description, we assume that if quantLevel $= k$, then $k - 1$ horizontal lines (quantization levels) are drawn as shown in Figure 2, and then each spectrum components are quantized into either one of $\{1..k\}$. We draw each $k - 1$ quantization level such that we settle the first or highest level with power p which has the maximum power among frequency components in frequency domain $(1, 2, ..., \texttt{cutOffFreq} - 1)$; then afterwards, the second or later levels with recursively are defined as $\frac{p}{2^{i-1}}$. This $k - 1$ quantization level separates the range of power in a frequency spectrum into k intervals $[0, \frac{p}{2^{k-2}}), [\frac{p}{2^{k-2}}, \frac{p}{2^{k-3}}), \ldots, [\frac{p}{2}, p), [p, \infty)$. Then we relate each interval with quantization values in increasing order from the start-point. We quantize each frequency components on the basis of power and intervals by finding an interval containing the power and then quantize with the value related. Figure 2 shows a situation with quantLevel $= 4$, and each 4 quantization levels are represented by dashed lines. As an example, a feature vector $v := (2, 3, 4, 3, 3, 3, 2, 2, 1, 1)$ is generated.

We describe how multiplied feature vectors are made redundant. To tackle frequency spectrum errors derived from the differences in microphone performances, we change each quantization level to slightly below those defined previously and then generate each multiplied feature vectors. More precisely, we introduce two parameters:numCand and jitter. The parameter numCand specifies the number of multiplied feature vectors or candidates for matching. The parameter jitter specifies tolerance from each quantization levels referenced. We could define recursively numCand $- 1$ sets of quantization levels by shifting to below with jitter from each level referenced. Figure 2 shows two different sets of quantization levels represented by dash-dotted lines and dotted lines, respectively, where numCand $=$ 3 and jitter $= \alpha$. Then in accordance with each set of quantization levels, we newly generate feature vectors $v' = (3, 3, 4, 4, 3, 3, 2, 2, 2, 1)$ and $v'' = (3, 3, 4, 4, 3, 3, 3, 2, 2, 1)$.

CommonVectorAggregation manages a table H storing joint probabilities, which are temporal coincidences of two of each feature vectors occurring. Specifically, the table H stores information on which feature vectors that occur simultaneously and how many times as a whole given sound-pressure series. The temporal coincidences of each of two collections of feature vectors are evaluated as shown in Figure 3, If after comparing or matching, there exists a feature vector contained by both sets of feature vectors, then it is the representative in the time-window. On the other hand, if several vectors coincide, then we select one unaffected from the parameter jitter as far as possible. In the case of Figure 3, the coincidence between v'_A and v_B is preferentially selected as the representative feature vector $(3, 3, 4, 4, 3, 3, 2, 2, 2, 1)$, but v''_A and v''_B.

Finally, we compute the information entropy between given two sound-fields by using the number $H.count$ of all temporal coincidences and the number $H[v]$ of occurrences of each feature vector v. The procedure described in lines 8–10 of Algorithm 2 computes the average information entropy. Thus, we return the information entropy.

 ISBN: 978-1-61208-246-2

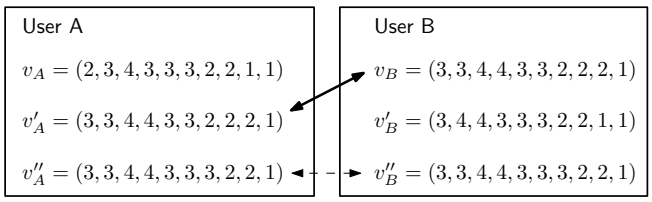

Figure 3. The matching process between characteristic vectors

IV. Empirical study for information retrieval accuracies

In this section, we present experimental results and discuss requirements described in Section II. Specifically, we observe transitions of the information entropies under each situation: sharing event-sound or not. We use two smartphones as test mobile devices, in which microphone performances differ.

Figure 4 shows an environment in which evaluation experiment we performed. We assume that there are users (A and B) who sit around a table in a cafeteria and wish to communicate in a new ad-hoc group. In this situation, we wish to observe whether proposed similarity has capabilities to distinguish conversation-fields in ambient-noise, and using microphones with different performances. For ambient-noise, we deploy a loudspeaker at the position ambient-noise source and produce crowd-noises recorded preliminarily at a cafeteria. For event-sounds, we had two people converse at positions A and B. We placed mobile device A, B, and C five meters away from ambient-noise source. Then, to make sure mobile devices A and B shared the same context, we placed them about one meter apart. On the other hand, to make sure A and C did not share the same context, we placed them about 10 meters apart.

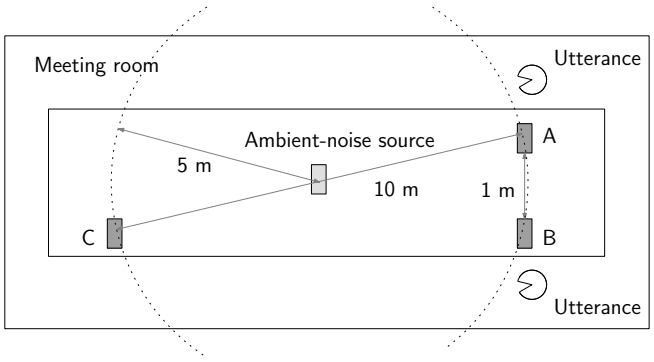

Figure 4. The overview of experimental environment

We show parameters on mobile devices and on the cloud server in Table I and Table II. Under this configuration, we performed the FFT procedures and computed information entropy 37 times for each 3-second cycle. Additionally, feature vectors are defined in the domain $[0..9]^{15}$. We generate six candidate feature vectors for each sliding time-window.

Table I
THE PARAMETER VALUES ON MOBILE DEVICES IN THE EXPERIMENT.

The sensing time	`duration`	3 sec
The frequency of sampling	`samplingFrequency`	8192 Hz
The data size after contraction	`sendDataSize`	300

Table II
THE PARAMETER VALUES IN THE SOUND-FIELD PROXIMITY INFERENCE METHOD IN THE EXPERIMENT.

The size of (FFT's) time widow	`FFTwinSize`	32
The overwrap ratio for continuous windows	`FFToverwrap`	50 %
The cut-off frequency	`cutOffFreq`	16
The number of quantization levels	`quantLevel`	10
The jitter value of each quantization level	`jitter`	0.5
The number of candidate feature vectors	`numCand`	6

Figure 5 shows increasing process for information entropy between A with B (solid line) and between A with C (dashed line). We set axis of the graph in Figure 5 such that the time progress corresponds to the horizontal axis and the entropy progress corresponds to the vertical axis. We plot each average of information entropy in a number of trials.

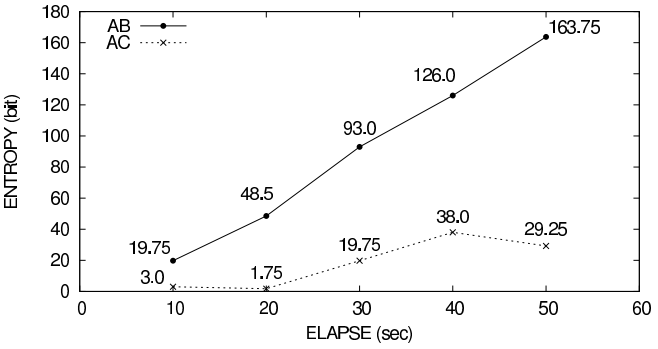

Figure 5. The increasing series on information entropy when all mobile devices are in separate positions

Discussions on accuracies of information retrievals: Here, we discuss the accuracies of information retrievals on the basis of the results shown in Figure 5. We observe from the progress of information entropies between AB(s) that the proposed similarity estimates appropriately event-sounds contained in conversation. Additionally, from results

of AC(s), the contributions of ambient-noise to the similarity can be more suppressed than AB(s). We note that by examining in full detail both increasing processes of entropies for each AB(s) and AC(s), we can decide suitable parameter values for the threshold used at grouping and an expired time (or terminated communications with the cloud server). In fact, breaking down with a linear regression analysis for each increasing process, the entropy of AB(s) increases at least four trials to ones of AC(s) for each cycle. Now we discuss about average values of entropies on 10 sec which are close to each other. Indeed, for AB, we have the average $\mu_{AB} = 19.75$ and the standard deviation $\sigma_{AB} = 9.42$. Similarly, for AC, we have $\mu_{AC} = 3$ and $\sigma_{AC} = 6.0$. Then, since we obtain $\mu_{AB} - \sigma_{AB} > \mu_{AC} + \sigma_{AC}$, proposed system might recognize appropriately difference contexts with high probabilities. With this knowledge from experimental results, for example, if we want searching to stop for up to 10 seconds, ad-hoc grouping systems output users who have the entropy exceeding 10 $(> \mu_{AC} + \sigma_{AC})$ as the search results. Under these conditions, we could restrict the false positive so that results AC(s) shows that we can distinguish the relative positions of A and C. Additionally, the accuracy is better than experimental results in [9]. Tarzia et al. showed localization accuracy, which measures correctness for recognizing different room where user visited, achieved 69 %, when sample time is 30 seconds. On the other hand, since there exist higher gaps between AB (93.0) and AC (19.75), proposed similarity may have higher noise-robustness than [9] at least in this case.

Finally, we discuss an essential insight for which we set the parameter information entropy thresholds with 10. We consider the probabilities of grouping with a user who attacks a system sending artificial sound-fields generated randomly. In this situation, we assume that the probabilities can be smaller than $\frac{1}{2^{10}}$.

Discussions on the differences in microphone performances: we could find out that proposed similarity evaluates appropriately on sound-fields without being dependent on the differences in microphone performance, since we can check clearly that transition of the information entropies of AB(s) increases appropriately, despite each A and B have mobile devices in different microphone performance. Therefore, it can be used as similarity between any mobile devices equipped with different microphone performances without lowering search accuracies.

Discussions on communication cost with accuracies: In the estimation experiments, under the configuration shown in Table II, each original sound-pressure series is contracted, its data size reduced to $\frac{1}{8}$, by Algorithm 1. Despite this fact, the results in Figure 5 show that search accuracies can be guaranteed sufficiently. Therefore, by deleting unnecessary features from the original sound-pressure series, we can reduce the network communication costs, or system-wide loads and network I/O loads by presented contraction algorithm.

V. Summary and future works

We proposed new similarity criteria for ambient sound based on information theoretical features of sound-fields. Experimental results verified that the similarity has sufficient search accuracy to be applied to ad-hoc grouping systems. Therefore, the proposed similarity has higher search accuracy and is more robust to differences in microphone performances. Furthermore, we proposed a contraction based on a FIR-like strategy in mobile devices. This contraction not only enables us to reduce network communication costs, but also ensures high search accuracy.

For future works for inferring context proximity in "real-time", architectures are required from the point of view of scale-out and scale-up. Supposing practical services, an ad-hoc grouping system will receive many sound-fields repeatedly for every unit time. Thus, the system has to compute in the unit time for all pairs of sound-fields received. This causes higher computation costs. Therefore, we need a technique for lightweight filtering that identifies pairs that do not need to be computed while restricting false positives.

References

[1] RingReef, http://ringreef.com/ (26.11.2012).

[2] B. Wellman, J. Boase, and W. Chen, "The networked nature of community: Online and offline," *IT & Society*, vol. 1, no. 1, pp. 151–165, 2002.

[3] N. Priyanha, A. Chakraborty, and H. Blakrishnman, "The cricket location-support system," *Proc. ACM MOBICOM 2000*, pp. 32–43, 2000.

[4] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *Proc. ACM MOBICOM 1999*, pp. 59–68, 1999.

[5] Ekahau, Inc., Ekahau Positioning Engine, http://www.Ekahau.com/ (26.11.2012).

[6] J. Randall, O. Amft, J. Bohn, and M. Burri, "LuxTrace: indoor positioning using building illumination," *Personal and Ubiquitous Computing*, vol. 11, No. 6, pp. 417–428, 2007.

[7] LINE, http://line.naver.jp/en/(26.11.2012).

[8] D. E. Sturm, M. S. Brandstein, and H. F. Silverman, "Tracking Multiple Talkers using Microphone-Array Measurements," *Proc ICASSP-97*, pp. 21–24, 1997.

[9] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," *Proceedings of ACM MobiSys '11*, 2011.

[10] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and T. Cambell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," *Proceedings of ACM MobiSys '09*, pp. 165–178, 2009.

[11] T. Nakamura, Y. Sumi, and T. Nishida, "Neary: Conversation Field Detection Based on Situated Sound Similarity," *IEICE Trans. INF. & SYST., Vol. E94-D*, pp. 1164–1172, 2011.

 ISBN: 978-1-61208-246-2