Vol.2013-MBL-65 No.16
Vol.2013-UBI-37 No.16
2013/3/14

O-MUSUBI: 環境音を利用するアドホックグルーピング – 音場の情報理論的素性に基づく類似度 –

寺本 幸生[1,a)]　野田 潤[1,b)]

概要：本論文は，アドホックグルーピングシステムを実現するために，環境音を利用する方式を提案する．アドホックグルーピングの要素技術は，ユーザの携帯端末でセンシングした環境音（音場）の類似性を基準に，近い音場を持つユーザを高精度に検索することである．一般的に良く知られている従来の類似度（コサイン類似度など）を音場に対して適用する場合，偽陰性を排除し，かつ同時に偽陽性を低減できないため，新たな類似測度を適切に設計する必要があった．また，様々な携帯端末を対象とする場合，マイク性能の差による検索精度の劣化への対策も必要である．大規模なシステムを想定する場合，システム全体の高負荷を緩和するために，音場は必要十分な情報を含みつつ，できる限り小さく表現する必要がある．我々は，同時発生確率を情報量で測定する新しい音場のための類似度を提案する．また，評価実験により，提案する類似度が各要件を十分に満たすことを示す．

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

Sachio Teramoto[1,a)]　Jun Noda[1,b)]

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.

1. はじめに

Social networking services (SNS) の普及に伴い，情報共有するユーザの指定は多様化している．例えば， ある期間，同じ場所を共有しているユーザを一時的にグルーピングし情報共有するような，アドホックなコミュニケーションサービスが考えられはじめている [1].

[1] NEC クラウドシステム研究所
Kawasaki, Kanagawa, 211–8666, Japan.
[a)] s-teramoto@bx.jp.nec.com
[b)] j-noda@cw.jp.nec.com

アドホックなグルーピング対象の指定は，同じような状況下にある端末を高精度で検索する技術が肝要である．つまり，グループを形成する際，少なくとも実際にメンバにしたいユーザを含み（偽陰性排除），それ以外のユーザはできる限り含まない（偽陽性発生率の制限）よう，正確に検索する必要がある．また，同時に，大規模なコミュニケーションサービスを想定する場合，様々な場面での利用を想定し，広範囲の空間を網羅する必要がある．ユーザの状況を推定するアドホックグルーピングシステム [2] において，センサを利用したユーザ間の状況の類似性を推定すること

は，最もプリミティブな課題の一つである．基本的なアプローチは，各端末の絶対的な位置情報を取得し，利用する手法である．GPS により，緯度・経度のような位置情報を取得し，端末間の距離を測定する手段である．一方で，GPS で測位できない場所（特に，屋内や地下）では，様々な位置測位の代替技術が提案されている．例えば，超音波を用いた Cricket [3] や ActiveBat [4]，無線 LAN を用いた Ekahau [5]，室内光を用いた LuxTrace [6] など様々な研究がなされている．

しかし，これらの手法は，広域な空間を網羅することを要件として持つ場合，適用が困難と言える．つまり，大量のセンサを広範囲に配置しなければならない．従って，設備コストおよび保守コストが高くなる問題を持つ．また，何らかの観点でセンサを配備することが困難な場所でも，端末間の近さを特定したい場面でも利用可能な方式が望まれる．

絶対的な位置を測位するのではなく，互いに同じような状況にあることを間接的に推定する手法に関しても，いくつか検討されている．例えば，加速度センサを利用し，同時に一定時間振動させたり [7]，同時に同じボタンを押すことを近くにいることの証拠として利用する手法である．

しかし，この手の手法は，端末数が増加すると検索性能は著しく悪くなる．なぜなら，異なる場所にいるにもかかわらず，同じだと認識してしまうような衝突する確率，すなわち偽陽性発生率が増加する．実際，大規模なコミュニケーションサービスで実現することを考慮すると，ユーザ数が数万人程度で衝突確率が限りなく 1 に近づくことは，誕生日のパラドックスのアナロジーで容易に推測することができる．

上記の課題を踏まえ，音場の情報理論的素性に基づく類似性によりユーザ間の状況の近さを推定する手法を提案する．ユーザの周囲の環境音は，マイクがあれば収集することができる．ここ数年広く普及し始めたスマートフォンをはじめ多くの携帯端末に必ず備わっているマイクを利用することで，広域な空間の網羅性と設備コストの両面を同時に解決できる．また，検索精度に関する課題に対して，偽陽性発生率を制限するよう活用できると考える．なぜなら，センシングした音場に含まれる情報量は，加速度やボタンなどをトリガとする手法よりも，多くの情報量を含んでいるからである．

本論文は，以下に示す構成に従う．第 2 節で，音場に基づく状況類似性推定の関連研究について言及し，音場の類似性に求められる要件を記述する．第 3 節は，音場に則したアドホックグルーピングシステムの全体構成，および，携帯端末とクラウドサーバそれぞれの処理について述べる．そして，第 4 節で，提案した音場の類似度に対する実証実験の結果を報告し，第 2 節で記述した要件に則した有効性について考察する．最後に，第 5 節で，得られた成果をまとめ，今後の課題を与える．

2. 音場に基づく状況類似性推定

2.1 関連技術とその課題

音場の解析に基づく位置測位の技術として，Sturm らは，マイクロフォンアレイを利用する手法 [8] を提案している．この手法は，音源のいくつかの移動モデルから導出したカルマンフィルタにより空間の移動を認識するものである．この手法は多数のマイクを配備する必要があるため，従来のセンサを利用する手法と同様に，広範囲を網羅することが困難である．

設備コストのかからない技術として，Tarzia ら [10] は，環境音の fingerprint 認証に基づき部屋単位で位置を特定する手法を提案している．この手法は，比較的簡潔な手段で異なる部屋どうしを識別することができるが，測位の解像度に関して高い精度を保証していない．この精度に関して提案手法が優れていることを第 4 節で考察する．

Lu ら [9] は，各ユーザのコンテキスト推定のために，音場を利用する手法を提案している．この手法は，音場の様々な素性に基づき，機械学習の下で，ユーザが会話しているのか，音楽を鑑賞しているのか，など環境音から推測できる状況を高い精度で推定することができる．しかし，識別できる音場に対しては限定的であり，異なる会話場に対する区別についてまでは言及していない．

特定の音場に関して，Nakamura ら [11] は，会話場に注目し様々な観点で優れた知見を得ている．特に，会話場の類似性をコサイン類似度により評価し，識別するような基本的なアーキテクチャを設計している．

一方で，多様な音場の類似性を評価する際に，最も慎重に設計しなければならない観点の一つが適合率と偽陽性発生率である．適合率は，正しくグルーピングしたいユーザ数を R，検索結果得られたユーザ数を N とした時，$\frac{R}{N}$ で表される．同様に，偽陽性発生率は $\frac{N-R}{N}$ で表され，適合率 $= 1-$ 偽陽性発生率の関係を有する．コサイン類似度は，音源との距離差や各端末のマイク性能差などにより正しく近さを測定することが困難な場面がある．偽陽性が問題となる例について以下に述べる．至る所で観測し得る定常的な背景音（**環境雑音 ambient-noise**）に対し，突発的に発生する特徴的な音（**イベント音**）は，多くの場合短い時間である．コサイン類似度では，騒音とイベント音を平等に扱うため，音場間の類似性に対して騒音が強く影響し，ユーザの周囲の音場を特徴づけるイベント音の貢献が低く抑えられる．そのため，至る所で観測し得る環境雑音を共有するすべてのユーザを検索結果として得ることになり，偽陽性発生率は高くなる．従って，音場に含まれる騒音の貢献を抑え，イベント音に高い効用を与える類似度を設計する必要がある．また，検索精度は，マイク性能差によって，ひどく劣化することがある．異なる精度のマイク

によってセンシングした音場の音圧値の時系列情報には，特に音圧値の大小に差が顕れる．従って，定常的な背景音に対して，音圧値に差があるデバイス間では，実際に同じ音場を録音していても，異なる環境下にあると誤った評価をしてしまう．これは，音場の類似性を適切に評価することができず，コサイン類似度は偽陰性を排除しきれないことを意味する．

また，上記検索精度の課題を踏まえたうえで，大規模なサービスをリアルタイムで提供することを想定すると，別の問題も浮上する．実際，スケールアウトするアーキテクチャを設計するとともに，システム全体の処理負荷を軽くするため，各ユーザの携帯端末から送信されるデータサイズは，できるだけ小さく制限する必要がある．実際，音場のセンシングデータは，必要以上に多くの情報量を含み，システム全体の負荷や通信コストの観点で非効率的である．特に，必要以上に大きなサイズのデータの受信は，通信 I/O がシステムの可用性に強い制限をかける．つまり，通信コストがボトルネックとなり，正しいタイミングでグルーピングできないことが問題である．従って，センシングしたデータを検索精度に影響を与えない範囲で，縮約する手法が必要である．

2.2 音場の相対的な近さを推定する類似度に対する要件

以下に，音場の類似度に関する設計指標を示す．

(i) **高い検索精度**． 提案する類似度は，対象音場に内在する特異値 (イベント音) 間の一致性を情報理論的な素性に基づき評価する．具体的には，相互情報量の概念を導入する．つまり，音場に含まれるイベント音，発生確率の低い情報，が一致したときに，より高い類似性を加味する．これにより，環境雑音の一致には類似度への貢献を制限され，偽陽性発生率を抑えることができる．従って，高い検索精度を保証し，お互いに類似する状況下にある端末群を関連付けることができる．

(ii) **マイク性能や音源距離に関する差を解消**．マイク性能の差や音源との距離差は，センシングした音圧値の大小に顕れる．このデータ値の差を解消するために，周波数スペクトルから特徴量（特徴ベクトル）を生成する際，冗長性を持たせ複数の特徴ベクトルからなる特徴ベクトルの族を抽出する．そして，この特徴ベクトル族どうしのマッチングにより情報量を算出する．これにより，音圧値の差を吸収し，性能差や距離差を柔軟に扱うことができる．

(iii) **通信量の低減**．上記要件を満たし，大規模なシステムを想定しても，リアルタイムな類似推定処理を実現するには，システム全体の処理負荷の低減しなければならない．また，システムの可用性を考慮し，通信コストがボトルネックとなることを避けなければならない．この際，携帯端末から送信されるデータ量を削減するために，ローパスフィルタを適用し，データを縮約する．縮約後のデータは，拍の時系列情報に丸められるが，それでも，依然，状況の類似性を推定するための情報を保持する．従って，この縮約手法に基づき類似度を算出することは，検索精度を保持しつつ，システムへの負荷を軽減することができる．

3. O-MUSUBI: 音場に則したアドホックグルーピングシステム

図 1 に，環境音を利用する類似性推定方式を利用したアドホックグルーピングシステムの概要を示す．以降，我々は，この類似性推定方式のことを O-MUSUBI[*1] と呼ぶ．アドホックグルーピングシステムは， O-MUSUBI 方式を用いて，複数の携帯端末からセンシングした音圧値の時系列データのそれぞれのペアの類似度を算出し，それらの類似度に応じたグループを形成する．

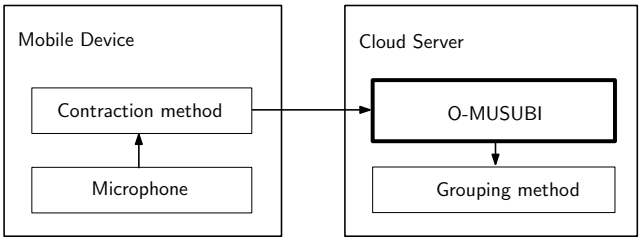

図 1 O-MUSUBI を利用するアドホックグルーピングシステム

携帯端末における処理は，以下の手順に従う．マイクで録音した音圧値の時系列データから縮約方式 Contraction method が不要な情報を排除し，データを縮約する．そして，縮約したデータをクラウドサーバへ送信する．ここで，録音時間はシステム全体で共有している単位時間で区切り，上記の処理を単位時間ごとに繰り返すことを想定する．

クラウドサーバの処理は，受信した音圧値の時系列データの集合に対し，すべてのペアごとに類似度を求める．二つの音圧値の時系列データ間の類似度は，音場近接性推定方式 O-MUSUBI が算出する．グループ管理機能 (Grouping method) は，類似推定エンジンの判定結果に応じて，グループ情報を更新し，最終的に生成されたグループの情報に応じて，各携帯端末に通知する．

3.1 携帯端末の処理

携帯端末での処理は，大きく 3 つの工程からなる．音をセンシングする処理，センシングしたデータを縮約する処理，および縮約したデータを送信する処理である．

ここで，フーリエ変換による周波数スペクトルの導出，および特徴量抽出などの処理は，各端末側で処理するのではなく，サーバ側に負わせる．これは，各音圧値の時系列

[*1] “**音**” に基づいてヒトやモノを “**結ぶ**” 方式.

データ間に発生する微細な時系列方向のズレを補正する必要があるからである．具体的には，第 3.2 節で記述される Algorithm 2 内の手続き TimeSynchronous が与えられた二つの音安置の時系列データを基に，時間軸方向のずれを補正する．また，安価に実装できる多種多様な携帯端末を対象とし，計算資源が乏しい機器でも動作させる側面も有する．

本節では，携帯端末におけるセンシングしたデータを縮約する処理について記述する．センシングしたデータを縮約するために，有限インパルス応答 (FIR) に準ずるローパスフィルタを適用する．高周波成分をカットしても，加工後のデータは拍の情報を保持する．この拍の情報には，イベント音のタイミングやテンポ（ペース）に基づく特異点が内在し，所望の検索精度を保ったまま，データサイズを大きく削減することができる．

Algorithm 1 は，縮約処理を記述している．入力は，パラメータ `samplingFreqency` で指定されたサンプリング周波数で，`sensingTime` で指定した時間だけ録音した音圧値の時系列データを格納した配列 `buf` である．出力として，時間窓ごとに最大値を取り出し，それらを要素として持つ配列を生成する．

類似推定エンジンとすべての携帯端末のあいだで共有すべき取り決めは，録音時間と送信するデータのサイズのみである．送信するデータサイズをパラメータ `sendDataSize` で指定する．また，連続する 2 つの窓の重複量を `overwrapRate` ($0 \leq$ `overwrapRate` < 1) で指定する．

Algorithm 1: 携帯端末の縮約処理

input : 音圧値の時系列データ配列 `buf`.
output: 縮約したデータの配列 `contractData`.

$w = \left\lfloor \frac{\texttt{buf}.length}{(1-\texttt{overwrapRate})*\texttt{sendDataSize}+\texttt{overwrapRate}} \right\rfloor$;
$w' = w \times$ `overwrapRate`;
for $i = 1, k = 0$; $i <$ `buf`$.length$; i++, k++ **do**
  if `buf[0]` $<$ `buf[`i`]` **then**
    `buf[0]` $=$ `buf[`i`]`;
  if $k == w$ **then**
    `contractData.push_back(buf[0])`;
    $k = 0$; $i -= w'$;
    `buf[0]` $=$ `buf[`i`]`;
return `contractData`;

送信端末をスマートフォンとする場合，マイクの仕様上，最低でもサンプリング周波数 8kH となる．3 秒間センシングした時，秒間 32 点に縮約したデータをサーバに送信する場合，重複率 `overwrapRate` 50% とすると，窓サイズ `windowSize` $= 506$ ごとに最大値をとるよう動作する．

3.2 クラウドサーバの処理

本節では，第 2.2 の方針を踏まえて設計したクラウドサーバ上の処理について述べる．

クラウドサーバは，類似推定方式 (O-MUSUBI) により音場の類似性を判定し，その結果に応じてグルーピング方式 (Grouping method) が関連付けるべき端末群の情報を更新する．

O-MUSUBI の処理を Algorithm 2 に示す．手順の概要は，はじめに波形間の時系列方向の同期を実施し，次に，各時間窓ごとに，特徴ベクトル生成，情報量計算を実行する．

Algorithm 2: O-MUSUBI: 音場間の情報量計算

input : 2 つの音圧値の時系列データ s_0, s_1.
output: 2 つの時系列データに対する類似性.

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

3.2.1 時系列方向の同期

手続き TimeSynchronous は，音圧値の時系列データ間の細微なズレの補正を行う．ここでは，時系列上のズレを補正するアルゴリズムを述べる．

同期処理は，以下の手順に従う．2 つの音圧値の時系列データに対し，極値のリストを抽出する．ここで，極値は，時系列上で直前・直後を含む 3 つ中で最大値をとり，さらにある一定値以上の差を有する値とする．従って，時系列データは複数の極値をとり得る．次に，一方の極値リストの各極値 x_i に対して，もう一方のリストから最も時刻差が小さい極値 y_i を見つける．そして，各ペア (x_i, y_i) の時刻差の平均値を用いて，データをシフトすることにより同期をとる．

3.2.2 音場の類似性推定

（補正後の）時系列データ s_0 と s_1 に対し，サイズ `FFTwinSize` の時間窓をスライドしながら，時間窓ごとに情報量を算出する．ただし，`FFTwinSize` は，FFT (Fast Fourier Transform) へのデータサイズに対応するため，2 の累乗を仮定する．ここで，時間窓のスライドは，前後が重複することが望ましい．これは，フーリエ変換の特性の関係もあるが，時間窓の境界値付近でイベント音が発生した場合における特徴量抽出のとりこぼしを防ぐ目的がある．

連続する 2 つの時間までが重複する割合を FFToverwrap ($0 \leq$ FFToverwrap < 1) で指定することとする．例えば，実際の重複サイズは，時間窓のサイズ FFTwindowSize を 64 とし，連続する時間窓が互いに $\frac{1}{8}w$ 重複する場合，$w = (1 - 0.125) * 64 = 56$ となる．

Algorithm 2 の 2 行目の for 文は，情報量を算出する．

まず，2 つの入力 s_0, s_1 ごとに，FFT を実施し周波数スペクトラム S_i を取得する．

次に，SpectrumQuantization が，冗長性を考慮した複数個の特徴ベクトル族 V_i を生成する．特徴ベクトルの定義域は，二つのパラメータ（カットオフ周波数 cutOffFreq と量子化レベル数 quantLever ）によって定義される．例えば，cutOffFreq $= 11$ で，quantLevel $= 4$ のとき，任意の特徴ベクトル v は，空間 $[1, 10] \times [1, 4]$ 上で定義される．図 2 に，例を示す．

ここで，補足として，携帯端末のパラメータ sendDataSize と類似性推定方式のパラメータ cutOffFreq の関係について言及する．つまり，cutOffFreq は，sendDataSize の下限値を与える．FFT により cutOffFreq より低い周波数帯の周波数スペクトルを得るためには，少なくとも sendDataSize $\geq$ 2cutOffFreq を満たさなければならない．また，送信するデータサイズと情報量の間にはトレードオフの関係がある．送信データサイズ sendDataSize が大きければ，時間窓の個数は，それだけ多く取れるからである．

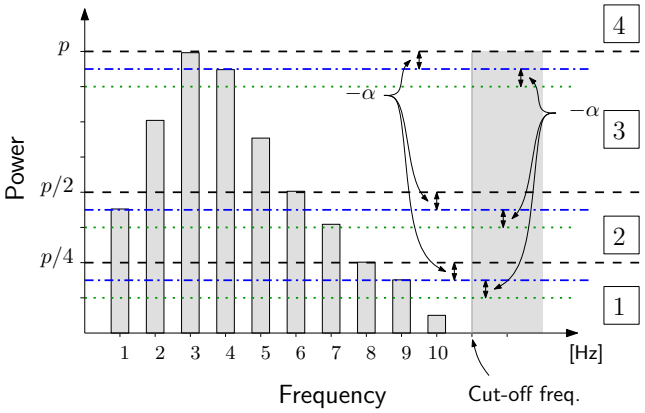

図 2 周波数スペクトルの量子化による特徴ベクトル生成

具体的な周波数スペクトルの各周波数成分の量子化は，以下の手順に従う．量子化レベルが k のとき，スペクトルパワーに関して $k-1$ 個の量子レベルをとる．基準となる一つ目の量子レベルを (カットオフ周波数よりも小さい周波数成分の中で) 最も高いパワー p とする．以降，2 番目以降の量子レベルは $\frac{p}{2^{i-1}}$ のように再帰的に設定する．ここで，$k-1$ 個の量子レベルにより，周波数スペクトルのパワーの値域は，k 個の区間 $[0, \frac{p}{2^{k-2}}), [\frac{p}{2^{k-2}}, \frac{p}{2^{k-3}}), \ldots, [\frac{p}{2}, p), [p, \infty)$ に分割される．そして，各区間は，始点に昇順に量子値を関連付ける．最終的に，各周波数成分に関して，そのパワーがどの区間に属すかに応じて量子化される．例えば，図 2 では，量子化レベル数を 4 で，各量子レベルをそれぞれ dash 線で示している．このとき，特徴ベクトル v は，$v = (2, 3, 4, 3, 3, 3, 2, 2, 1, 1)$ として得られる．

次に，量子化に冗長性を持たせた特徴ベクトルの多重化について記述する．マイク性能の差や音源との距離などの要因から派生する周波数スペクトルの誤差を解消する為に，量子レベルのとり方を変化させ，複数の特徴ベクトルを生成することを考える．具体的には，2 つのパラメータ (特徴ベクトルの候補数 numCand と量子レベルのズレの許容値 jitter) を導入する．候補数 numCand - 1 個の量子レベル設定は，初期設定の各量子レベルから jitter だけ減らすことで，再帰的に定義できる．図 2 は，numCand $= 3$, jitter $= \alpha$ を用い，上記とは別の 2 つの量子レベル設定例をそれぞれ dash-dot および dotted の線で示している．各量子レベル設定に応じて，新たに，特徴ベクトル $v' = (3, 3, 4, 4, 3, 3, 2, 2, 2, 1)$ と $v'' = (3, 3, 4, 4, 3, 3, 3, 2, 2, 1)$ を得る．

CommonVectorAggregation は，2 つの特徴ベクトル族間の一致性に応じた，同時発生確率のテーブル H を管理する．具体的には，どの特徴ベクトルが何回出現したかを記録する．また，特徴ベクトル族どうしの一致性判定は，図 3 のように，複数の異なる特徴ベクトルが一致する場合，できるだけ jitter の効果が少ないものをその時間窓を代表する特徴ベクトルとする．図 3 の場合，v'_A と v''_B の一致性を捉え，$(3, 3, 4, 4, 3, 3, 2, 2, 2, 1)$ が出現したこととしている．また，$v_A \neq v'_A$ で，$v_A = v'_B$ かつ $v'_A = v_B$ の場合，特徴ベクトル v_A か v_B のいずれかを時間窓を代表するベクトルとする．

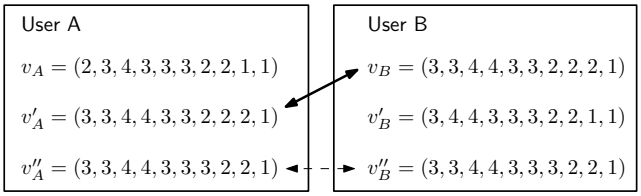

図 3 特徴ベクトル族どうしのマッチング処理

最後に，Algorithm 2 は，総一致回数 $H.count$ と，各特徴ベクトル v の発生回数 $H[v]$ を用いて情報量を計算する．Algorithm 2 の 8 行目から 10 行目までの for ループで，平均情報量を算出する．最終的に，(選択) 情報量を出力する．

4. 検索精度に関する評価実験

本節は，提案する類似度の検索精度に関する評価実験の結果を示し考察する．特に，イベント音を共有するユーザどうしは高い類似性を示し，イベント音を共有しないユーザとは低い類似性に留まることを示す．具体的には，それ

ぞれの状況下で，類似性の測度である情報量の増加推移を観測する．使用した端末として，マイク性能の異なる 2 台のスマートフォンを用いた．

評価実験は，図 4 に示す環境で実施した．ここで，ユーザ A と B は，カフェテリア内で同じテーブルを囲んでおり，新たにアドホックなグループを形成して，コミュニケーションを図ろうとしているシチュエーションを仮定する．環境雑音として，Ambient-noise source の位置にスピーカを置き，事前にカフェで録音した群衆ノイズ (crowd-noise) を発生させた．また，2 人の発話者の会話をイベント音として想定しいる．使用する携帯端末 A, B, and C の配置は，AB 間を約 1 メートルで発話者近くとし，C を発話者から約 10 メートル離している．また，Ambient-noise source から，各端末への距離は等しく，5m の間隔を置いている．

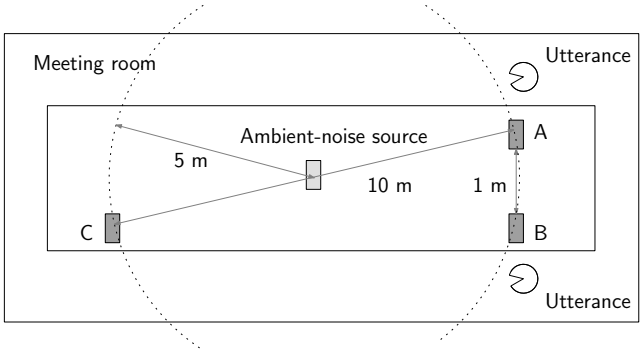

図 4 実験環境の概要

また，実験で設定した端末と類似推定方式のパラメータ値をそれぞれ 表 1 および表 2 に示す．この設定の下で，FFT と情報量算出の手続きは，3 秒ごとに 37 回実行され，各時間窓内では $[0..9]^{15}$ の空間が定義され，音圧値の時系列データごとに，それぞれ 6 個の特徴ベクトルを生成する．

表 1 実験で設定した端末上のパラメータ値

センシング時間	`sensingTime`	3 秒
サンプリング周波数	`samplingFreqency`	8192 Hz
縮約後のデータサイズ	`sendDataSize`	300

表 2 実験で設定した O-MUSUBI のパラメータ値

(FFT の) 時間窓のサイズ	`FFTwinSize`	32
連続する時間窓窓の重複率	`FFToverwrap`	50%
カットオフ周波数	`cutOffFreq`	16 Hz
量子化レベル	`quantLevel`	10
量子化境界線のズレ値	`jitter`	0.5
特徴ベクトルの候補数	`numCand`	6

図 5 は，図 4 に記載した携帯端末 AB 間の情報量と携帯端末 AC 間の情報量に関して，時間経過（横軸）に伴い，それぞれの類似測度の情報量（縦軸）が，どう推移するかを示している．実線が AB 間の情報量で，破線が AC 間の情報量に関する結果である．各時点での類似度は，それぞれ 5 回の試行の平均値をプロットしている．

4.1 検索精度に関する考察

図 5 の結果に基づいて，検索精度に関して考察する．AB 間の情報量の推移から，会話の含まれるイベント音を適切に評価していることが分かる．一方で，AC 間の情報量の推移から，環境雑音の類似度への貢献を低く抑えられていることが分かる．図 5 の AB 間の類似度と AC 間の類似度の増加傾向を評価すると，適切な閾値とタイムアウト時間を定めることができる．実際，線形回帰分析により増加傾向を一次式で表現すると，時間経過に伴い，約 4〜5 倍の差が出ることが分かる．ここで，最も平均値が近接している 10 秒付近のデータについて考察する．AB 間の平均 μ_{AB} と標準偏差 σ_{AB} それぞれ $\mu_{AB} = 19.75$ と $\sigma_{AB} = 9.42$ となった．また，AC 間については，$\mu_{AC} = 3.0$ と $\sigma_{AC} = 6.0$ となった．実際，$\mu_{AB} - \sigma_{AB} > \mu_{AC} + \sigma_{AC}$ を得，正しく，異なる状況を識別することができている．例えば，10 秒以内で検索を完了させる場合，情報量 $10(> \mu_{AC} + \sigma_{AC})$ を超えたユーザの情報を検索結果として出力すれば良い．この条件の下では，A の状況と近しいユーザの検索を実行した場合に，C を出力するような偽陽性は強く抑えられる．同時に，B を検索結果に含まないという偽陰性も発生しない．さらに，この結果から，提案類似度の精度は，[10] で示されている実験結果より，少なくとも本実験環境においては優れていると言える．具体的には，Tarzia らは，30 秒間録音した音圧値の時系列情報を用いて，ユーザが訪れた異なる部屋を約 69%の精度で識別する結果を得ている．一方で，上記の設定に基づく提案類似度では，AB 間の情報量 (93.0) と AC 間の情報量 (19.75) の間に十分な開きがあるので，より強い検索精度，および雑音耐性があると言える．

最後に，情報量の閾値を 10 に設定することの本質的な意義は，ランダムに発生させた人工的な音場情報を送信してくる攻撃に対して，偶然あるグループに属す確率が $\frac{1}{2^{10}}$ に抑えられることである．

4.2 マイク性能差の吸収

図 5 で，情報量の推移に明らかな差が確認できることから，マイク性能の差に依存することなく，適切に類似度を評価していることが分かる．このように異なるマイク性能を備えた携帯端末間でも，検索精度を下げることなく類似度として使用することができるといえる．

4.3 通信量の低減

評価実験では，Algorithm 1 により，表 1 の設定の下で，データサイズをおよそ $\frac{1}{8}$ に縮約した．これだけ縮約して

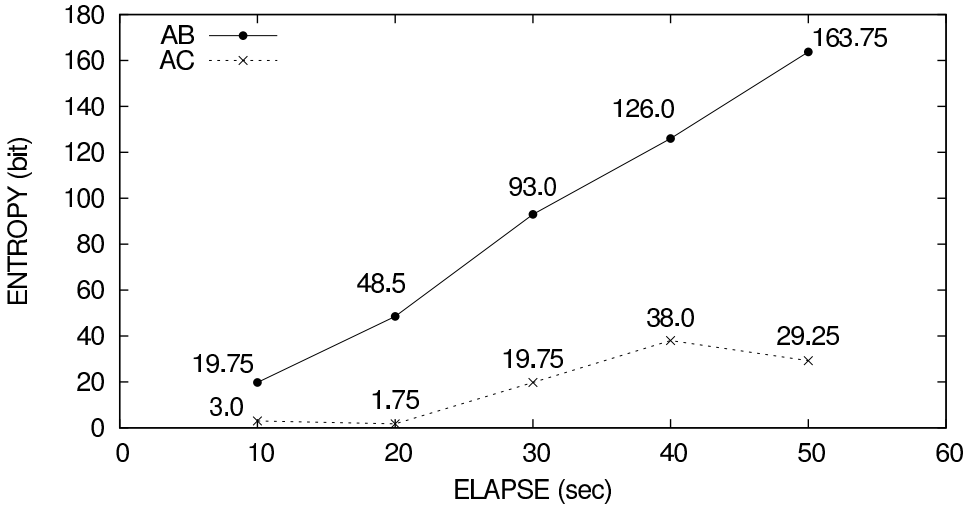

図 5 異なる状況下での類似度の増加推移

も，図 5 の結果から，検索精度を十分保証できることが分かる．従って，音圧値の時系列データから不要なデータを削除することで，検索精度をある程度維持したまま，単位時間当たりの受信データ通信量を低減できているといえる．

5. まとめと今後の課題

音場の情報理論的素性に基づく新しい類似度を提案し，アドホックコミュニケーションサービスに適用するに十分な検索精度を有することを実証実験により示した．また，本類似度は，マイク性能の差や音源からの距離差などを吸収するために，特徴量を冗長化し微細なズレを解消するよう設計されている．さらに，提案した類似度は，FIR に準ずる縮約処理を適用することができ，データサイズを低く抑えられる側面を有する．

今後の課題として，リアルタイムに状況推定するために，サーバをスケールアウトする方式が必要である．実運用を想定すると，単位時間ごとに多数の音場が入力される．このとき，入力された音場の全ペアに対し類似度を計算しなければならない．そのため，偽陽性を抑えながら比較すべきでないペアを特定する軽量なフィルタ機能が必要となる．

参考文献

[1] RingReef, `http://ringreef.com/`, 26/11/2012.
[2] B. Wellman, J. Boase, 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/`, 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] 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,* 2009.
[10] 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.
[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.