

## SoundShare:アドホックな情報共有のためのグルーピングの手法

依田みなみ<sup>†</sup> 羽田久一<sup>†</sup>

**概要:** 本研究では、接点の少ない人同士で一時的に形成されるグループ内で情報共有を行うために、音を利用したグルーピング手法を提案する。高周波数の矩形波を流し、それらを受信した Android デバイス間で共通のグループ ID を生成し、同じグループ ID を所有している物同士で情報共有を行う。また、本研究では、応用例として写真共有を目的としたアプリケーションを構築した。

## SoundShare: Sharing Information Method for Ad-hoc Community

MINAMI YODA<sup>†</sup> HADA HISAKAZU<sup>†</sup>

**Abstract:** We propose a method to form an ad-hoc community among a number of people by sound signal from Android device. In order to share information among a group, the leader-device streams a sound of high-frequency wave, then, other-devices receive the sound and analyze it as a group-ID. We also develop photo-sharing application based on this method.

### 1. はじめに

近年、携帯端末—特にスマートフォンのカメラ性能が高まったことにより、人々は手軽に写真を撮影できるようになった。これにより、日常風景や友達と遊んだ時の写真等を共有する機会も増加した。

一般的に、複数人間で写真を共有する手段として、Facebook や LINE のグループ内に写真をアップロードしたり、Dropbox や Picasa などの WEB サービスを利用して共有をする。それらのサービスを利用するには、SNS を利用する場合であればアカウントを教え合う必要があるし、ファイル共有サービスを利用する場合はメールアドレスを教える必要がある。

友人など頻りに連絡をとりあう関係間で写真を行う場合は、既にアドレスや SNS アカウントを知っていることが多い。しかしながら、懇親会などの初対面同士の会合や、大人数で写真を共有したい場合等、普通の生活では接点が少ない人々とデータの共有を行いたい場合は、アカウントやメールアドレスを教え合う必要があり、写真共有までに手間がかかる。また、共有する人数が増えれば増えるほど連絡先の交換に時間がかかってしまい、非効率的である。

上記の問題を解決するために、本研究では、SNS 上の個人情報を使わず、複数端末を簡潔にグルーピングし、写真共有をおこなえる SoundShare を提案する。

本稿は以下のように構成される。まず、次章で本研究が想定する利用例に基づき、スマートフォン端末やタブレット間における、簡便な写真共有の必要性を述べる。そして、本研究に対する関連研究についても述べる。3章では、本研究が提案する音によるグルーピング手法の概要と、本手法を実現するための音分析のアルゴリズムについて述べる。

4章ではアプリケーションも含め、実装したシステムについてについて説明する。5章ではテストについて述べ、6章では今後の課題を中心に議論する。最後に本稿の結論を述べる。

### 2. 携帯端末グループ化と情報共有

#### 2.1 音を利用した情報送信インタラクション

上記で述べた問題を解決するために、SoundShare では、アドホックなグループ化(以降、グルーピングという)と写真共有を実現するために音信号を利用する。端末同士をグルーピングするリーダーの役目を担う端末(以降、リーダー端末)がグループ ID を生成する音を鳴らし、他の端末(以降、メンバー端末)はそれを録音する。その音を分析するとグループ ID が生成され、この ID が一致すれば同じグループであるとみなす。そして、そのグループ ID にもとづいて各々の端末が写真をアップロード・ダウンロードするで写真を行う。

#### 2.2 局所的なグルーピングにおける音信号の有用性

音を用いて、その場限りのグループを生成する場合には以下のような利点が考えられる。

- 音を利用する場合はスピーカーを通すことでボリュームを調整することができる。
- 人数が少なければ小さい音に、逆に多ければ大きな音に変えることで対応でき、グループの作成範囲を動的に変化させることが可能である。

表 1 他通信技術との比較

	通信距離	対人数	ペアリング方法
①NFC	10cm程度	基本的に1対1	端末を機器にかざす
②Bluetooth	半径10m~100mの距離	基本的に1対1	探索、認証、暗号化
③QRコード	カメラで認識できる範囲	カメラで認識できる範囲にいる人間	カメラで読み取り
④赤外線通信	0.3m~1m	基本的に1対1	赤外線送信部を機器の受信部に向ける
⑤SoundShare	音が届く範囲	音が届く範囲にいる人間	音を録音する

<sup>†</sup> 東京工科大学 メディア学部

- 音は電波や赤外線等とは異なり、人間が認知することができる。これにより、どの端末がグルーピングの対象となっているのかを「音が聞こえた」ことで理解できるので、ユーザビリティの面から考えても適当である。
- 音によるグルーピングはネットワーク環境に左右されることがないので、電波状況の不安定な場所や地下であっても、グルーピングを行うことができる。グルーピング以降の、写真のアップロードやダウンロードの動作は、各々の端末が同じグループIDを持っていただいても操作することができる設計をした。本研究ではグルーピングをローカルで完結させることを重視した。

### 2.3 他通信技術との比較

まず通信技術では、NFC[3], Bluetooth[4], QRコード[5], 赤外線通信の4つに対して、通信距離、一度に何人と情報共有できるのか、加えてペアリングの方法を表1にまとめた。

筆者らはまず、グルーピングを行うにあたって、端末同士がある程度の距離にあっても、グループの一員として検知されることを重要視した。これは、人数が増えたときに対応できるようにするためである。例えば、表1-1の1NFCの場合には通信距離が10cm以下のため、端末同士をかなり近づけなければいけない。人数が多くなればなるほど、現実的なインタラクションではなくなる。表1-2Bluetoothの場合には、通信距離自体は大きく、大人数になっても対応できると考えるが、通信する相手は基本的に1対1である。また、Bluetoothはペアリング手順が複雑であり、一方の端末を検出可能状態にし、もう一方の端末は周辺でBluetooth機能を音にしている機器を探索する。そして、見つかった場合は接続相手を指定し、双方のパスキーの確認をするという4ステップを踏まなければならない。表1-3QRコードの場合には、同じQRコードを見せることが出来れば、人数に制限はないがQRコードを複数の人に読み取らせる場合には大画面に表示するか、大きな紙に印刷するといった対策が必要となり、その場でのグルーピングに対応することは困難である。また表14の赤外線の場合には、基本的に一人しか対応できない上に、通信距離も1m以下である。本研究では、限定的なグループを煩わしきなしで生成することが最も重要だと考えているが、以上のように他の通信手段によるペアリング方法では達成できないと考える。



図1 SoundShare 全体構成図と動作の概要

### 3. 関連研究

端末間における情報通信は、研究はもちろんのこと、既にサービスとして一般に普及されているものも多い。

Bump[1]は2つのスマートフォンをぶつけることによって、プロフィール情報を交換するシステムである。スマートフォンの動きを加速度センサーを用いて検出し、GPSを利用することによって近傍にいる端末を検出する。GPSは建物内に入ってしまうと精度が劣化してしまうため、屋内では近傍にいる端末の検出に失敗しやすいという問題がある。また、加速度センサーの値が変化するという単純なインタラクションを採用しているため、意図せずして近くにいる人間とグルーピングされてしまう可能性がある。また、端末をぶつけるという操作を多くの人と行うことも困難であり、個人間の情報の交換のためのインタラクションとしては適しているが、グループを生成する手法としては向いていないと考えられる。

寺本ら[2]は環境音を利用としたアドホックグルーピングの手法を提案した。グルーピングをする際に周囲の環境音を利用し、音の類似度をサーバーサイドで処理することでグルーピングを行っている。この研究では、同じ環境音が得られる範囲のみでグルーピングをすることを目的としており、例えば大会議で会場全員の端末を一斉にグルーピングを行うときなどには向いていない。

### 4. SoundShare の設計と実装

#### 4.1 システムの設計

アプリケーション全体の構成と動作の概要を図1に示す。本システムではグルーピングを行うために、7ビットのビット列を音信号に変換して送信する。7ビットのうち4ビットはグループIDとなり、残りの3ビットはエラーチェックと訂正に用いる。エラーチェックと訂正については後述する。

まず、リーダー端末が自身の端末から7ビットを生成するグループID生成音(以降、グループ音という)を発生させ、グループ端末は音を受信し解析することでグループIDを入手する。リーダー端末は、グループの名前、開始時間、

表 2 音の意味と周波数

音の意味	周波数
開始の区別音	2500Hz
文字列 1 を生成	3000Hz
文字列 0 を生成	2000Hz
中間音	3500Hz

終了時間を設定する。その設定をサーバーのグループ専用フォルダにアップロードする。グループ端末はインターネット接続環境にいる時に、そのグループ情報ファイルを取得し、グループ情報ファイルにしたがって写真のアップロードを行う。写真のアップロードは、グループ情報ファイルに設定された開始時間と終了時間内に撮影された写真を自動でアップロードするようにする。グループ専用フォルダにアップロードされた写真は、アプリ内の写真ギャラリーで閲覧することができ、保存したい写真を選択することで端末に保存することが可能である。

#### 4.2 アプリケーションの実装

まず、音情報の解析の詳細な手順を以下に示す。リーダー端末からグループ ID をエンコーディングした音を出力し、メンバー端末でその音を録音してフーリエ変換にかける。この手法を利用するために、Android のライブラリである AudioRecord, MediaPlayer, Visualizer の 3 つを利用した。まず、録音の機能は AudioRecord [7] を利用して wave ファイルを作る。次に、MediaPlayer[8]を利用して、AudioRecord で作られたファイルを読み込む。そして、android でフーリエ変換や波形を表示する Visualizer[9]を利用してフーリエ変換を行う。

本研究では 4 種類の音を使用した。まず、音によるビット送信の開始音として始めに 2500Hz を 1 秒流す。開始音以降に流された 3000Hz の音をビット 1, 2000Hz の音ビット 0 として Android の Visualizer クラスで解析する。ビット音は 0.125 秒の長さである。より短い転送時間を実現するためにそれ以上の短い音で解析実験をしたところ、音が早すぎて解析にミスが出るが多かったため、0.125 秒で実装をした。

また、Visualizer クラスの音の解析タイミングによってはビットが重複して解析されてしまう場合があるため、3000Hz と 2000Hz の間に 3500Hz の音を挟み、3500Hz の音が間に挟まっている場合のみビット生成をするようにした。以上の説明を表 3 にまとめる。

音を高音にした理由は、周囲が騒がしくなっても解析しやすくするためである。我々の実験では 2000Hz 以下の音は環境音に吸収され、解析しにくくなってしまったため、2000Hz 以上の音を利用した。

表 3 配列に格納される値

インデックス	0	1	2	3	4	5...	N-2	N-1	
データ	RF0	高周波(n/2)個	RF1	IF1	RF2	IF2	...	高周波(n-1)/2	(n-1)/2個の場合

Visualizer クラスのインスタンスでは、サンプリング周波数は 44, 100Hz, キャプチャサイズは 1, 024 と設定した。ファイルが渡されると同時に、8 ビットごとに、キャプチャサイズの大きさと同じ 1, 024 個の配列が生成される。その後、フーリエ変換を行う。前述した 1, 024 個の配列の中に周波数の大きさ順に音の強さを格納する。

開始音である 2000Hz 音が格納されている配列を検出するための例を挙げる。これはサンプリング周波数が 44, 100Hz の場合、分解能は 21. 53Hz(サンプリング周波数/2 /1, 024)である。表 3 にあるように、分解能 1 に対して二つの配列が生成され、フーリエ変換をした結果の実部(RF)と虚部(IF)がそれぞれの配列に入っている。(表 3 参照)また、本稿では虚部の値を使っている。つまり分解能が 21. 53Hz の場合、2000Hz の音の強さの結果は、計算上 86 番目の配列に入っていることになる。単位時間内に 86 番目の配列の中身に格納されている数値が一番大きければ、2000Hz の音が強く出ているということである。本研究では、単位時間内に一番音が大きく出ている音の周波数で解析を行った。

以上の手順を踏むことによりユーザーの端末同士が同じグループであるかどうかを判断するグループ ID を生成する。そして、このグループ ID が一致するか否かでグループ ID を行う。

#### 4.3 音受信時のエラーチェックとエラー訂正

音からビットを生成するとき、周波数フィルタでおおよそのビットを生成することができるが、雑音などが入った場合の対応として、1 ビットエラーチェックと 1 ビットエラー訂正を実装した。グループ音から解析して生成された 7 ビットの 0 と 1 の数字が、もし誤っていた場合、1 ビットのエラーを検出し、該当ビットを訂正する。これにより、グループ ID の解析をより確実にすることができる。

7 ビットの内訳は、グループ ID4 ビット(G1~G4)とエラーチェックとグループ ID の誤りをチェックするための 3 ビット(E1~E3)である。

検出手法であるが、まずグループ ID の 4 ビット生成後、下記の組み合わせで XQR をとり、その結果が 0 となるように E1~3 のビット列を生成する。

- E1, G1, G2, G4
- E2, G1, G3, G4
- E3, G2, G3, G4

表 4 評価実験の環境

実験環境	昼食時間帯の学内マクドナルド
使用端末	Nexus5, GalaxyS2
端末間の距離	30cm

7ビットすべての生成が終わったら、それを音に変換し、受信側、つまりメンバー端末で解析を行う。解析方法は、音を解析して生成されたビット列に対し、再度上記の組み合わせで XOR をとり、結果が 0 になればエラーなしとみなす。3 通りの組み合わせのうちどこかが間違っていた場合は、エラー検出を行う。G1~G4 のどれかが誤っていた場合、エラー検出ビットが 2 つ間違ふことになる。例えば、G1 は E1 と E2 にまたがっているため、E1 と E2 が異常だった場合 G1 が誤っているとみなし、ビット反転をおこなうといったものである。エラー検出ビットが誤りだった場合は、エラー検出ビットは 1 つのみ誤りになるので、該当のエラー検出ビットをビット反転する。

#### 4.4 写真アップロード機能の実装

リーダー端末は、グループの情報設定画面でグループ名と、アップロードしたい写真を撮影した時間帯(想定では飲み会等の開始時間から終了時間)を指定する。これらの設定はリーダー端末が行い、サーバにアップロードされる。

各々の端末はサーバにアップロードされたグループ情報ファイルを取得し、情報内の時間にしたがって写真をアップロードする。アップロードは時間にしたがって自動でアップロードされる。

#### 4.5 サーバサイドのシステムの実装

サーバサイドでは主に PHP で実装を行った。リーダー端末からグループ情報設定ファイルがアップロードされる時に、グループ ID 専用フォルダを生成し、その中にグループ情報設定ファイルを格納。グループメンバーはグループ ID でサーバに問い合わせをし、グループ情報ファイルを取得したり、写真をアップロード・ダウンロードをおこなう。写真がアップロードした際、アプリ側でギャラリー表示の際にサムネイルを利用するために、サムネイルも一緒に生成するようにした。

### 5. 実験と評価

#### 5.1 音によるグループ ID の生成と伝送

騒がしい飲食店でグルーピングの評価実験をおこなった。実験環境を表 4 にまとめる。

#### 5.2 実験結果

端末間の距離で、同じグループ ID が生成されることを確認した。よって、理論上、正常にアプリケーションが動く端末が 30cm 以内であればすべての端末をグルーピングすることができる。

### 6. おわりに

本稿では、アドホックにグループ化を行うために、音を解析することでグループ ID を生成するという手法を提案した。これを実現するために、環境音に含まれない高周波数の音をグループ音とし、ビット送信時のエラーチェックとその訂正を行うことで、送信の精度を向上させた。

今後の課題として、実験時に近くで食事をしていた人の多くがグループ音に反応したため、日常にない異常な音として周囲の人に怪訝がられた点があげられる。この点については、異常を感じさせないように、更に高周波数の音をグループ音として使用することで、改善できる可能性がある。しかし、我々はグルーピングの際に音を利用したメリットとして「どこの範囲にいればグルーピングされるかがわかりやすい」という点も重視しているため、グループ音を非可聴域で行うかどうかは考える必要がある。

#### 参考文献

- 1) BUMP TECHNOLOGIES, INC., Bump, <http://bu.mp/company/>
- 2) 寺本 幸生, 野田 潤, 2013, 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 -
- 3) NFC Forum, NFC, <http://www.nfc-forum.org/home/>
- 4) Bluetooth SPECIAL INTEREST GROUP, Bluetooth, <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>
- 5) デンソーウェーブ QR コードドットコム, QR コード, <http://www.qrcode.com/request.html>
- 6) Android で音声解析! Visualizer による高速フーリエ変換 (FFT), <http://geekn-nerd.blogspot.jp/2013/05/androidvisualizerfft.html>
- 7) AudioRecord - Android developer official site <http://developer.android.com/reference/android/media/AudioRecord.html>
- 8) MediaPlayer - Android developer official site <http://developer.android.com/reference/android/media/MediaPlayer.html>
- 9) Visualizer - Android developer official site <http://developer.android.com/reference/android/media/audiofx/Visualizer.html>
- 10) ビット誤りについて <http://z.apps.atjp.jp/memo/teisei.html>