

# アドホックなコミュニティのための 写真共有の手法

依田みなみ† 高松有紀† 羽田久一†

接点の少ない人々による一時的に形成されるグループでの情報共有のために音を利用したグループ生成の手法を提案する。近傍の端末を直接認識しグループ化することにより SNS 等のネットワーク上の個人情報と切り離されたグループの生成を可能とする。ユーザーの携帯端末から識別番号を含ませた音を出力し、その音を検知した周囲の端末をグループとして登録することで、その場限りのグループを生成し写真共有を実現するシステムを構築した。

## Photo sharing method for ad-hoc community

MINAMI YODA† YUKI TAKAMATSU†  
HISAKAZU HADA†

The authors propose a method to form an ad-hoc community among people using mobile device and sound. This method composes an ad-hoc community of mobile devices that are physically close, which is not made upon personal connections on social network services. This method is undertaken by letting mobile devices to output a sound containing identification number, then, group up the devices which recognized the sound. In this paper the authors will describe a photo sharing system made based on this method.

### 1. はじめに

近年では、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、写真が日常的に撮影されるようになった。スマートフォンは、普段から持ち歩くことが一般的であるため、旅行や記念日などの特別な日ではなくとも、写真を撮影する機会が増加している。また、自分だけではなく他の誰かと一緒に写真を撮影した場合、それを被写体同士の間で共有する機会が多いと考えられる。

スマートフォンの場合、デジタルカメラとは異なりネットワークに接続することができるので、写真をプリントしたものを配布するのではなく、画像データのみを交換することも頻繁に行われている。そのような画像データを相手に送る、または受け取る際には、メールサービスや SNS が利用される。これらのサービスを利用するためには、メールならメールアドレスを交換する必要があり、SNS を利用する場合であればアカウントを知り友人登録等を行う必要がある。

日常的に会う機会が頻繁にある人の場合は、既にアドレスやアカウントを知っていることの方が多い。しかしながら、大人数で行われる懇親会等に参加した場合等、普段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もあると考えられる。そのような日常的にあまり接点のない人と写真を共有するにはいくつかの不都合があると考えられる。

- (ア) 人数が多い場合に、各人のアドレスやアカウントを登録するのは時間もかかり、手間がかかる。
- (イ) 写真を共有するという限定的な目的のために、SNS のアカウントを教えてしまうことにより、その後 SNS 上での情報発信に煩わしさを感じる場合がある。

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、情報共有したい端末の数が多くなったとしても対応が可能な手法として、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 利用シナリオ

携帯電話の普及に伴い携帯端末のカメラを使用して誰かと写真を撮影することは日常的なものとなった。それらを利用して写真を一緒に撮影する場合、お互いに写真を共有

† 東京工科大学 メディア学部

Tokyo University of Technology, School of Media Science

したいと考えるのが一般的である。その際にはメールサービスまたは SNS が用いられることが多い。それらのサービスを使う場合にはメールアドレスまたは SNS アカウント（以下、ネットワーク上の個人情報という）を知っている必要がある。普段から親しくしている人同士の場合は、それらを知っている可能性は高く、またその場で教え合っても問題はない。しかし、飲み会の場で初めて出会ったり、日常的に接点が少なかったりする相手の場合、写真共有のためだけにネットワーク上の個人情報を教えることに、抵抗がある場合がある。日常的に接点があっても、例えば会社の上司等、自分のプライベートを知られたくない相手も存在する。

よって、ネットワーク上の個人情報を教えずして、その場限りのグループによる写真共有という限定的な目的を達成するためのシステムを提案する。

また、大人数で撮影した写真を共有する場合には、現在では SNS 上でそのためのグループを作ることが多いと考えられる。しかしながら、前提として、共有したい人全員が同じ SNS を利用していること必要である。現在広く利用されている SNS は Facebook・LINE・comm・google+ 等多様にあり、その人の属する国や地域、コミュニティによって主として使うサービスが異なるため、全員が同じサービスを利用していないことも少なくない。

またそれが解決したとしても、それまで SNS 上で友達同士ではなかった場合、①名前検索・QR コード等を利用して、アカウントを見つける②友達申請をする③全員をグループ化するという手順を踏まなければならない、人数が増えれば増えるほど、手間がかかる。

それらを解決するために、ネットワーク上の個人情報を利用せず、かつ大人数となった場合にも簡単にグルーピングができる必要があると考えた。

## 2.2 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。さらに、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。

森口ら[2]は端末同士をなぞることでグルーピングする手法を提案している。学校での授業や企業の研修で行われるグループ活動において、少人数からなるグループを複数形成し、それぞれのグループが同じ目的を共有して個別に共同作業を行うような場合に、特にタブレット端末を活用し、ユーザーが自由にグループ間を移動して共同作業を行えるような「ダイナミックグループコラボレーション環境」を構築した。この環境では、各ユーザーが持つタブレット端末をお互いに結合して机上に並べ、画面上のオブジェクトを端末間でまたがって移動させることができる。しかし、本研究では懇親会等の数十人規模になった際にも対応できることを理想としているので、なぞるというインタラクションを数十人に対して行うのはグルーピングの手間が改善されたとは言えないと考える。

## 3. 音を利用した情報送信インタラクション

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を利用した。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、リーダー端末という）がグループ ID を生成する音を鳴らし、他の端末（以降、メンバー端末という）はそれを録音する。その音を分析し、グループ ID が生成され、これが一致すれば同じグループだと見なすことにより、グルーピングを可能にする。

本章ではまず音を利用する優位性を示し、他の通信技術、そして情報共有のためのアプリケーションと比較した結果を表にまとめる。

### 3.1 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。まずスピーカーを通すことで音量を調整することが出来る。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。

また、音は電波や赤外線等とは異なり、人間が認知することのできるものである。そのため、グルーピングの際に、ユーザー自身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」という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だと考えた。

そして最後に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、グルーピングだけは行うことができるのが利点である。

### 3.2 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[6]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

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

表 2 他アプリケーションの比較

	ペアリング方法
①Bump	GPS+加速度センサー
②LINE	ふるふる(GPS+加速度センサー)
③Facebook	名前/メールアドレスの検索
④写真共有マスター	wifi
⑤SoundShare	音によるグループID生成

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の①NFCの場合は通信距離が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1②Bluetoothの場合は、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1③QRコードの場合は、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。

次に情報を交換する前段階としての友人登録に着目し、Bump, LINE[3], Facebook[4], 写真共有マスター[5]の4つのアプリケーションについて比較を行った。

表2①Bumpの場合、端末同士をぶつけるというインタラクションでペアリングを行う。しかし、前述したように、位置情報を把握するためにGPSを用いているため建物内や地下ではうまく動作しない場合がある。本研究で最も利用されると想定している場所は飲食店なので、屋内や地下で利用されることを考えると不適當である。また同じように表2②LINEも端末を識別する仕組みは同じなので同様の問題が生じると考えられる。そして、表2③Facebookの場合は、アカウントを見つける手段として、メールアドレスや名前を検索することが多い。メールアドレスで検索する場合は、メールサービスを利用する時の手間は変わらない。

また、名前を検索する場合でも、同姓同名が出てくることは少なくないので、アカウントを見つけるだけでも時間がかかることもある。加えて、この方法でアカウント検索するには必ずネットワーク環境が必要となり、ネットワーク環境がない場所でのペアリングは面倒である。SoundShareの場合は、ペアリングまたはグルーピングの段階ではネットワーク環境を必要としないので、この問題を解決できる。

最後に表2④写真共有マスターだが、これは、wifiを利用する。一人の端末をアクセスポイント化し、他の端末はテザリングによって同じネットワーク環境を利用する。この方法の場合、8台までという台数制限が設けられているので、通信技術の比較でも述べたように、大人数のグルーピングに対応できない。

#### 4. 音による端末グループ生成の実装

本研究では、多くのスマートフォン端末に搭載されているAndroidプラットフォームを対象とし、Javaで開発を行った。本章では、まずアルゴリズムを述べ、次節でアルゴリズムを用いたアプリケーションの実装について述べる。

##### 4.1 アルゴリズム

本システムではグルーピングを行うために、6ビットのビット列を音信号に変換して送信する。音によるビットの表現は440Hz(A4)の音が100ms続いた場合に1とし、392Hz(G4)の音が100ms続いた場合を0として判定する。音情報の解析の手順は以下のとおりである。リーダー端末からグループIDをコーディングした音を出力し、メンバー端末では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 [11]を利用して、このライブラリでは、waveファイルを作ることが出来る。次に、MediaPlayer[12]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[13]を利用してフーリエ変換を行った。今回、サンプル音はラ(A4)とソ(G4)の音を利用している。ラは基準周波数440Hz

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



図 1 アプリケーションの全体図



図 2 グループ ID 生成機能

であり、ソは 392Hz であるため、抽出する周波数領域を絞ることが出来る。今回の実装では、音の分析精度よりもデバックのしやすさを優先し、この二つの音を選んだ。録音したら次に、ラとソを区別するために、高速フーリエ変換を行う。これにより音の周波数を解析し、音階を特定することが出来る。このライブラリを利用する場合、まず音ファイルを Visualizer に渡し、そのファイルに対して 100ms 毎のデータを用いて周波数の解析を行う。サンプリング周波数は 44,100Hz、キャプチャサイズは 1,024 と設定している。

また、解析を開始するタイミングはデータをコーディングするのに利用したラとソの音のうちどちらかが、一定の閾値を超えた時刻である。また、終了のタイミングはユーザーが録音停止ボタンを押した時としている。

ファイルが渡されると同時に、100ms ごとにキャプチャサイズの大きさである 1,024 個の配列が生成される。次に、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。ラの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、ラの音 440Hz の音の強さの結果は、計算上 41 番目の配列に入っていることになる。41 番目の配列の中身に格納されている数値が大きければ、ラの音が強く出ているということである。ソの音は 39 番目の配列に格納されているはずである。本研究では、配列に 90 以上の数値が入っていれば強い音がでていると判断した。そして、ラの音を検出した場合には 1、そしてソの音だった場合は 0 とし、1 と 0 を組み合わせさせたビット列を生成する。

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リングを行なっている。

#### 4.2 アプリケーションの実装

アプリケーションの構成としては、4 画面分の機能があり、それぞれタブによってページ変移ができる。

アプリケーション全体の構成としては、以下の図 1 に示す。はじめに、グループ ID の生成部分について述べる。あらかじめラとソを交互に鳴らすサンプル音を用意した。今回は PC からサンプル音を鳴らしたが、実際にはユーザーが持つ端末のスピーカーから鳴らす想定である。音を鳴らす前に、図 2 の赤枠で囲ってある“音を聴くタブ”にある図中①の録音開始ボタンを押す。そしてサンプル音を鳴らして録音し、音が鳴り終わったら②の録音停止ボタンを押す。その後、録音した音をフーリエ変換にかけるため、③の“録音したファイルを読み込み”というボタンを押す。これは、ユーザーが本当に目的の音が録音されているのかを確かめさせるためにも有効である。最後に④の“show groupID”ボタンを押すとフーリエ変換後の数字が、グループ番号となり表示される。次に、“階の設定タブ”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(想定では飲み会等の開始時間から終了時間)を指定する。

“会の設定”タブに変えると、“日付”ボタンで DataPicker が表示され、日付を選べる。そして、開始・終了時間は TimePicker が表示されてそれぞれ時間を選ぶことが出来る。指定したグループ名と、開始時間または終了時間は“情報取得”タブのページでユーザー自身が確認することが出来るようになっていいる。時間を指定すると、端末の SD カードから jpeg ファイルを検索し、そのファイルから Exif 情報撮影日時部分の情報を出力する。その時間が開始時間～終了時間の間であれば、サーバーに自動でアップロード

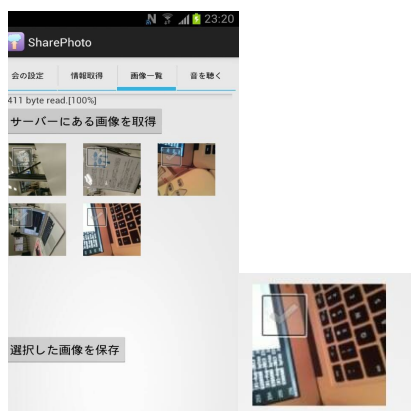


図 3 アップロードされた写真のギャラリー

される仕組みである。この機能により、会が終わった後にユーザーが写真を選択してアップロードする手間を省く狙いがある。最後に、アップロードされた写真だが、アプリ上でギャラリー view が用意されており一覧で見ることができる。また、サムネイル画像をクリックし、ダウンロードボタンを押すとローカルに写真のダウンロードができる。これらの機能は、タブで切り替えられるようになっており、ユーザーはページ遷移の際にストレスなく切り替えることができる。

## 5. 実験と評価

### 5.1 音によるグループ ID の生成と伝送

PC から出力したラとソの音を組み合わせさせたサンプル音を、開発したアプリケーションで録音、解析し、グループ番号を生成した。使用した端末は、GalaxyS2, Xperia acro, Xperia Z の 3 機種である。テーブル上においた PC の近傍に 3 つの端末を置き、データを取得する実験を行った。方法としては PC からサンプル音を出力し、それぞれの端末はアプリケーション上の機能によって音を録音、分析した。

### 5.2 実験結果

PC 上で「010101」というビット列から音信号を生成し、これをスマートフォン上で解析する実験を行った。今回の実験の場合、正しくグループ ID が取得できた場合「010101」という値が出るはずである。しかし結果としては、GalaxyS2 は「00101010000000」というグループ ID が出力された。次に Xperia acro は「010110100100」、そして XperiaZ は「0000010」という結果になった。GalaxyS2 では「0101010」また、acro では「0101101」と「010101」に近い数値が出現していることから、グループ ID 解析の手法を工夫することにより精度の向上が望めると考えている。以下にグループ ID 生成失敗の原因を述べ、この手法を活かした改善方法を次章で提案する。

原因として考えられるのは 2 点である。まず、端末に内蔵されているマイク位置の差異による影響である。録音し

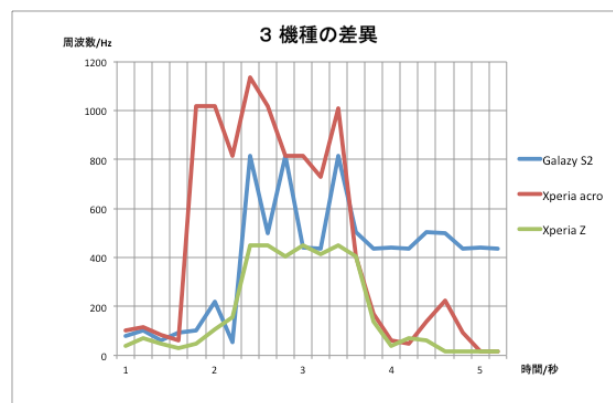


図 4 3 機種で録音した音の強度分布

た音の成分からもっとも音圧の高い周波数を調べ、コンソールに周波数の数値を出力した。それを図 4 のようにグラフ化した結果、XperiaZ のみ全体的に値が小さいことが分かった。これは、今回の実験ではすべての端末の背面を背にして机に置いていたことに加え、XperiaZ は他の 2 機種とは異なり、音を録音するマイクが端末の背面についていて、マイク穴を塞いでいたことが原因であると考えられる。そのために、録音された音が小さかったのだと考えられる。この結果から、ユーザーが必ずしも手で端末を持って録音するとは限らないので、アプリケーション上で、録音時の端末の保持方法を指示するアニメーション等の表示が必要であると考えられる。

次に、今回サンプル音として利用したラとソは単音である。これは、身の回りで聞こえる雑音等と比較しても、特徴のない音であり、解析の際には他の音に埋もれてしまう場合がある。実験で利用した端末全てのグループ ID に多く 0 が含まれているのは、雑音中に含まれるソの音を拾い、かつグループ ID の 0 と判断する条件式を通ってしまったからだと考える。環境音による影響を受けたとしても、グループ ID を生成出来るように ID 生成に利用する音の音域を選択し、雑音に対する耐性を十分に検討するべきだと考える。

今回の実験での問題としては、情報の符号化を行う場合に、始点を明確化するプロトコルが実装されていない点と、誤り訂正符号を取り入れていないため、ビットエラーが発生した場合の対策がとられていない点が挙げられる。また、端末ごとによるマイクの性能の差異を吸収するためには、解析時における動的な閾値の設定が必要となる。

## 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、音を分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ラ・ソの音でグループ ID の生成を行った。しかし、実験の結果から、特徴のない単音

であるラ・ソの音では、想定しているような飲食店のような騒がしい場所の場合、環境音に影響されてしまい、グループ ID の生成は難しいことが分かった。また、端末に搭載されているマイクの性能の差異が、そのままグループ ID 生成の際に差異を生んでしまうことがある。

今後の課題としては、どのような場所でも、かつ、どの端末を使用してもグルーピングが可能となるよう、グルーピングに利用する音の音程を変更すること、誤り検出符号や誤り訂正符号を用いることで、音信号を強固にすること、端末に備わったマイク間の差異をなくすような感度に対する補正をかけることなどが必要である。

## 参考文献

- 1) BUMP TECHNOLOGIES, INC. ,Bump ,  
<http://bump/company/>
- 2) 森口友也, 桑野元樹, 高田秀志, 2012, タブレット端末を利用したダイナミックグループコラボレーション環境の構築, 情報処理学会インタラクション, <http://www.interaction-ipsj.org/archives/paper2012/data/Interaction2012/interactive/data/pdf/3EXB-13.pdf>
- 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) Infrared Data Association, 赤外線通信, <http://www.irda.org/>
- 7) LINE 株式会社, LINE, <http://line.naver.jp/ja/>
- 8) Facebook, Inc. , Facebook, <https://www.facebook.com/facebook>.
- 9) キングソフト株式会社, KINGSOFT 写真共有マスター, <http://www.kingsoft.jp/allshare/>
- 10) 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>
- 11) AudioRecord - Android developer official site  
<http://developer.android.com/reference/android/media/AudioRecord.html>
- 12) MediaPlayer - Android developer official site  
<http://developer.android.com/reference/android/media/MediaPlayer.html>
- 13) Visualizer - Android developer official site <http://developer.android.com/reference/android/media/audiofx/Visualizer.html>