非同期分散マイクロホンによる ブラインド音源分離を用いた複数話者同時音声認識*

◎越智景子 (NII), 小野順貴 (NII/総研大), 宮部滋樹, 牧野昭二 (筑波大)

1 はじめに

音声認識技術を実環境で応用する際，会議やその他の対話が行われる場面では複数話者が同時に話して発話が重なることがあるため問題となる。しかも話者同士が近くに位置しているとは限らないため，集中型のマイクホンアレイではいずれかの話者に近づけると他の話者から遠くなってしまう。その一方で近年は，スマートホンといった録音機能を持つ携帯端末が普及し，1 台以上所持することも珍しくなくなってきた。本稿ではこれらを非同期マイクロホンアレイ [1] として用いた複数話者同時音声認識のプロトタイプシステムを提案する。2 台の端末により 2 話者の混合音声を分離し，音声認識性能の改善について検証した結果を報告する。

2 分散マイクロホンアレイによる音声認識

2.1 システムの概要

Fig. 1 に提案するシステムの構成を示す。本システムは，各話者が 1 台ずつ傍に置いたスマートホンによりそれぞれ録音を行うことを想定している。従来の集中型のマイクロホンアレイとは異なり，端末の設置場所を柔軟に決めることができるため，話者に近接した位置で録音をして高い SN 比を得ることが期待できる。録音データは 3G または Wi-Fi によりクラウドストレージに転送され，パーソナルコンピュータ (PC) により同期および音源分離と音声認識が実行される。分散型マイクロホンアレイでは，個々の端末で信号処理を分散計算する方法も提案されているが，ここではより簡便な実現方法として，ファイル共有システムに収録音を集め，一括して信号処理する方法を用いる。

2.2 iPhone と Dropbox を用いた分散マイクロホンアレイの実現

本研究では，スマートフォンとファイル共有システムを用いた分散型収音システムを，iPhone と Dropbox を用いて実現した [1]。録音およびデータの転送は iPhone アプリとして実装されている。Dropbox への転送は録音後自動で行われ，録音データとともに加速度センサの情報，リフレッシュレート 10-60Hz の GPS 情報が転送されるが，本稿では音声認識に必要な音声データのみを用いた。

2.3 ブラインド同期

各端末で独立に録音された音声信号は同期がとれておらず，録音開始時刻のずれ（オフセット）に加え，端末間のサンプリング周波数のミスマッチを含んでいる [2]。これらを補償するために，我々はブラインド同期の手法を開発してきた [3]–[4]。この手法では，オフセットを補償するために相互相関関数を最大にするよう全体をシフトし，サンプリング周波数のミスマッチは，短時間フーリエ変換 (Short time Fourier transform; STFT) の各時間フレーム内でミスマッチが一定であるとする近似に基づき，時間フレームのインデックスに比例した線形位相シフトにって補償する。サンプリング周波数ミスマッチの大きさは，音源が動かず定常であるというモデルを用いた最尤推定により算出する。

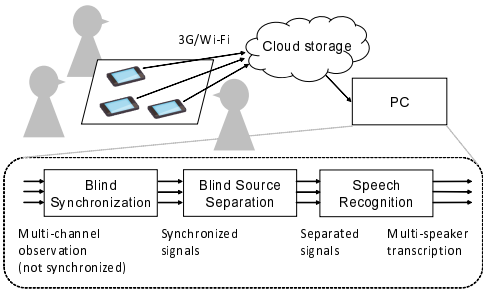

Fig. 1 System overview

2.4 ブラインド音源分離

想定した会議場面では，各録音端末の位置は未知であり，加えて，前節で述べたブラインド同期処理を行っても，一定のわずかな時間オフセットは残りうるので，話者位置情報を利用した信号処理は適用できない。そのため音源分離には，観測信号のみから音源信号を分離するブラインド音源分離 (Blind source separation; BSS) が適している。ここでは各話者が 1 台録音機器をもっていることを仮定しているので，音源数=マイク数の優決定問題となる。

BSS のアルゴリズムには，高速で安定な補助関数型独立ベクトル分析 (Auxiliary-function-based independent vector analysis; AuxIVA) [6] を用い，スケール不定性の解決のために，projection back [7] を適用した。

2.5 音声認識

音源分離後の音声は，音声認識により文字列に変換される。音声認識には，Julius version 4.3.1 [8] を用い，音響モデルおよび言語モデルは dictation kit [9] で公開されているものを用いた。

3 音声認識実験

3.1 実験方法

評価実験では 2 名の話者が会議を行っている場面を想定して，2 台のスピーカ (BOSE Comuter Mu-

*Automatic speech recognition for multiple speakers using blind source separation with distributed microphones by Keiko OCHI (NII), Nobutaka ONO(NII/SOKENDAI), Shigeki Miyabe, and Shoji Makino (Univ. of Tsukuba)

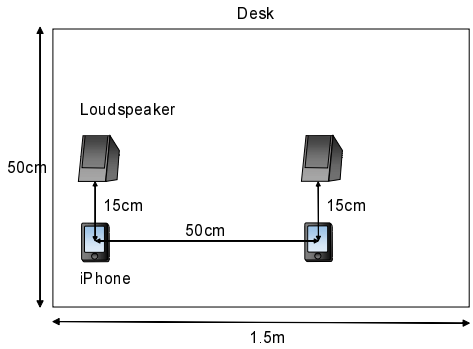

Fig. 2　Experimental setup

sicMonitor) から音声を再生し，15cm 離した 2 台の iPhone (Model A1453) によって録音した。スピーカー間の距離は 50cm である。Fig. 2 に使用機器の配置を示す。2 台のスピーカからは，新聞記事読み上げコーパス (JNAS) に含まれる ATR 音素バランス文 503 の読み上げ音声のうち，異なる 50 文 (6 分程度) をそれぞれ再生した。

分離前後の信号を比較するため，各 iPhone に対応する 2 つのチャンネルに対して以下の条件について，誤認識率を比較した。

- Mix: 2 音声が混合した状態の未処理の音声
- Sep w/o sync: 時間オフセットおよび周波数ミスマッチの補償なしで BSS により分離した音声
- Sep w shift: 時間オフセットのみ補償して BSS により分離した音声
- Sep w sync: 時間オフセット，周波数ミスマッチ両方を補償したうえで BSS により分離した音声
- Ideal: 1 音源のみを再生して録音した音声 (reference)

推定された時間オフセットは 21602 サンプルでサンプリング周波数のミスマッチは 1.63 ppm であった。

音声認識の音響特徴量はフレームシフト 10ms フレーム長 25 ms の 12 次元のメルケプストラム係数 (MFCC) とデルタ MFCC，デルタパワー計 25 次元である。音響モデルは JNAS コーパス約 86 時間により学習されたトライフォン HMM である。

3.2　結果と考察

Fig. 3 に，音声認識結果のモーラ誤認識率と単語誤認識率を示す。Mix は脱落，置換誤りが大きく，音声の重なりにより音声認識の結果が大きく影響を受けたと考えられる。また，同期をとらないまま BSS を適用した Sep w/o sync も脱落，置換誤りが大きい。この場合，BSS がうまく動作していないことがわかる。また，サンプリング周波数ミスマッチの補償をせずに分離を行った Sep w/o shift も大きな脱落誤りを持つ結果となった。BSS により精度よく音源分離を行うためには，時間オフセットの補償のみでは不十分であることがわかる。Sep w sync は上記 2 つに比較して誤認識率が小さく Ideal に近い結果となっているため，サンプリング周波数のミスマッチと時間オフセットを両方補償することによって精度よく BSS を行えたためと考えられる。

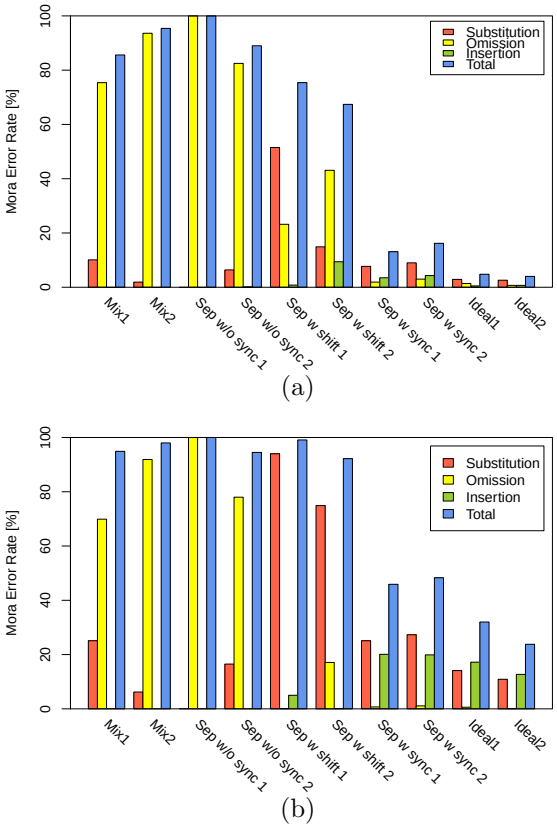

Fig. 3　(a) Word error rate and (b) mora error rate

4　終わりに

本稿では，複数話者の対話認識への応用を想定し，スマートホンによる非同期マイクロホンアレイを用いた複数話者同時音声認識のプロトタイプシステムを提案した。我々が開発してきたブラインド同期とブラインド音源分離の組み合わせにより，別々のスマートホンで録音した信号からでも高い分離性能が得られ，各話者の単独発話に近い音声認識精度を得ることができた。今後は 3 名以上の多数の話者による発話場面を想定した評価実験を行う予定である。

謝辞　本研究は文部科学省科研費基盤研究（B）(25280069) の助成を受けた。

参考文献

[1] 小野, 音講論, pp. 561–562, 2015.
[2] 小野, 音学誌, vol. 70, no. 7, pp. 391–396, 2014.
[3] S. Miyabe *et al.*, *Proc. ICASSP*, pp. 674–678, 2013,
[4] S. Miyabe *et al.*, *Proc. WASPAA*, Oct. 2013.
[5] S. Miyabe *et al.*, *Signal Process.*, vol. 107, pp. 185–196, 2015.
[6] N. Ono, *Proc. WASPAA*, pp. 189–192, 2011.
[7] N. Murata *et al.*, *Neurocomputing*, vol. 41, no. 1, pp. 1–24, 2001.
[8] A. Lee and T. Kawahara, *Proc. APSIPA*, pp. 131–137, 2009
[9] https://github.com/julius-speech/dictation-kit Accessed on Jan. 1, 2016.