人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046

ＡＩチャレンジ研究会 (第４６回)

Proceedings of the 46th Meeting of Special Interest Group on AI Challenges

CONTENTS

日　時　2016 年 11 月 9 日　　場　所　慶應義塾大学 日吉キャンパス 来往舎　大会議室
Keio University, Kanagawa, Nov. 9, 2016

社団法人 人工知能学会
Japanese Society for Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-01 (11/09)

ノンパラメトリックベイズと深層学習に基づく音声データからの教師なし語彙獲得 ～記号創発ロボティクスによる知能と言語へのアプローチ～

Unsupervised word discovery from speech signals based on Bayesian nonparametrics and deep learning --Symbol emergence in robotics towards understanding cognition and language--

谷口忠大
Tadahiro TANIGUCHI
立命館大学
Ritsumeikan University

taniguchi@ci.ritsumei.ac.jp

記号創発ロボティクスは人間のコミュニケーションを成立させている諸要素をロボットを用いて構成論的にモデル化することで，人間の知能理解をすすめること，実世界の系において人間を支援し活動するロボットを構築することを目指す学術領域である．筆者は，記号創発ロボティクスとは記号創発システムへの構成論的アプローチであると定義している．記号創発ロボティクスは様々なチャレンジを抱えるが，その一つはロボットによる言語獲得である．

人間の幼児は生まれた時点で特定の言語に関わる語彙や音素に関する知識を持たないところから学習をはじめていく．その幼児が親や他の存在から得る音声情報にもとづいて，音素に関する知識を得て，また，語彙に関する知識を得ていく．このような発達過程を計算論的にモデル化し，ロボットに持たせることは，人工知能研究においても認知発達ロボティクスの研究においても大きなチャレンジであると言える．

従来，ロボットが用いる音声認識システムは，大量のラベル付き教師データを人手により準備し，教師あり学習を通じてモデルをトレーニングすることで構築することが一般的であった．しかし，幼児にとっても，人間と自然な環境でインタラクションしつづけるロボットにとっても，このラベル付き教師データを取得することは出来ない．幼児がそのような明示的にラベル付けされた音声データではなく，自らの感覚運動器から得られる感覚運動情報とラベル付けされない音声データに基づいて語彙獲得していることは明らかである．このような過程を計算論的に表現するためには，教師なし学習にもとづいて語彙獲得のプロセスを表現する必要がある．

ノンパラメトリックベイズはベイズ理論の一部であり，ディリクレ過程やベータ過程を活用することで，隠れ状態数に関して柔軟な機械学習手法を構築することが出来る．例えば，ディリクレ過程を混合率の事前分布としたディリクレ過程混合ガウス分布では理論的に混合するガウス分布の数を無限個とした上で，推論を行うことで，用いるガウス分布の混合数を事前に固定することなく学習させる事ができる．また，深層学習は多段のニューラルネットワークを用いることで高い特徴抽出能力を持つことが知られており，近年，画像認識や音声認識，自然言語処理で活用されている．

本発表では，筆者らが取り組んできたノンパラメトリックベイズに基づく教師なし語彙獲得の機械学習手法について紹介しながら，深層学習を用いた性能改善についても併せて紹介する．具体的には階層ディリクレ過程隠れ言語モデル(Hierarchical Dirichlet Process-Hidden Language Model)とそれを用いたノンパラメトリックベイズ二重分節解析器について説明し，また，マルチモーダル情報を用いた語彙獲得の手法に関しても概説する．また，記号創発ロボティクスによる知能と言語へのアプローチについて展望を述べる．

参考文献

[Taniguchi 16] Tadahiro Taniguchi, Ryo Nakashima, Hailong Liu and Shogo Nagasaka, Double Articulation Analyzer with Deep Sparse Autoencoder for Unsupervised Word Discovery from Speech Signals, Advanced Robotics, Vol.30, (11-12) pp. 770-783. (2016)

[Taniguchi 16] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, and Hideki Asoh, Symbol Emergence in Robotics: A Survey, Advanced Robotics, Vol.30, (11-12) pp. 706-728. (2016)

[Taniguchi 16] Tadahiro Taniguchi, Shogo Nagasaka, Ryo Nakashima, Nonparametric Bayesian Double Articulation Analyzer for Direct Language Acquisition from Continuous Speech Signals, IEEE Transactions on Cognitive and Developmental Systems, Vol.8 (3), pp. 171-185. (2016)

[中村 15] 中村友昭，長井隆行，船越孝太郎，谷口忠大，岩橋直人，金子正秀，マルチモーダル LDA と NPYLM を用いたロボットによる物体概念と言語モ デルの相互学習，人工知能学会論文誌, Vol.30 (3), pp. 498-509. (2015)

[谷口 14] 谷口忠大，記号創発ロボティクス ―知能のメカニズム入門，講談社. (2014)

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-02 (11/09)

量子化 Deep Neural Network のための 有界重みモデルに基づく音響モデル学習

Acoustic Model Training based on Weight Boundary Model for Discrete Deep Neural Networks

武田 龍 [*1], **中臺一博** [*2], **駒谷和範** [*1]

Ryu TAKEDA[*1], **Kazuhiro NAKADAI**[*2], **Kazunori KOMATANI**[*1]

大阪大学 産業科学研究所 [*1], (株) ホンダ・リサーチ・インスティチュート・ジャパン [*2]

The Institute of Scientific and Industrial Research, Osaka University[*1]

Honda Research Institute Japan Co., Ltd.[*2]

rtakeda@sanken.osaka-u.ac.jp, nakadai@jp.honda-ri.com, komatani@sanken.osaka-u.ac.jp

Abstract

本研究では，Deep Neural Network (DNN) に基づく音響モデルの省メモリ化と高速化のため，パラメータを数ビットに量子化した DNN の構築を目指す．それには，1) 量子化に伴う認識誤りの低減と，2) 高速演算が可能な実装方法の開発，が必要である．1) に対しては，DNN の重みパラメータの正規化を，層単位ではなくノード単位に行う有界重みモデルに基づく学習アルゴリズムを提案する．2) に対しては，量子化した重みパラメータを索引に用いるルックアップテーブルを用いた実装方法を提案する．これらにより，少ないビット数でのパラメータ表現が可能となり，また，複数変数の高速な積和演算が実現できる．評価実験により，単語正解精度の低下を抑えて重みパラメータを 2-bit まで量子化でき，DNN のフォワード計算を 40% 高速化できることを確認した．

1 はじめに

1.1 背景

Deep neural network (DNN) は，高い識別精度により Gaussian mixture model (GMM) に代わって音声認識の音響モデルに広く使われている [1, 2, 3, 4]．一方，その膨大なパラメータ数により，使用メモリ量と計算コストは高く，DNN を適用可能な計算機はまだ限られている．GPU や分散処理を用いた実装 [5, 6] は，DNN の処理速度向上に効果的だが，リソースが制限されたシステムには適用が難しい．例えば，通常の CPU を備えた小型計算機や組み込みシステムなどがある．それゆえ，計算コストとメモリ量の観点で**軽量な DNN** が必要である．

量子化 DNN は，パラメータを量子化することで，GPU や分散処理がなくとも合理的な計算コストとメモリ量で DNN 計算を可能とする．Fixed-point DNN は，重みやバイアスパラメータ，中間層の入力変数を n ビット固定小数点で表現しており，Very Large Scale Integration(VLSI) 実装や CPU 上の Supplemental Streaming SIMD Extensions 3 (SSSE3) 命令を用いた実装により，DNN の高速処理を実現した [7, 8]．Fixed-point DNN のパラメータは，n ビットへの量子化と誤差逆伝播法を繰り返すことで学習される [7, 9]．しかし，最適な量子化ビット数 n の選択は経験的であり，多くの実験を必要とする．また，上記の実装は CPU の特殊命令セットや特殊なデバイスを活用している．CPU のみを搭載した小型計算機上や CPU を伴った FPGA への実装では，特殊な命令セットを前提としない DNN の方が適用しやすい[1]．

ルックアップテーブル（LUT）は DNN の高速計算に有効な方法の 1 つであり，重みパラメータからなる値を LUT の索引として用いる．例えば，一般的な CPU 上でもキャッシュメモリを活用することで，LUT を用いた DNN は SSSE3 命令と同等の高速処理が可能である．もちろん，LUT 技術はハードウェア実装にも適用できる．我々は，これまで重み有界モデルと境界収縮に基づいた学習アルゴリズムを提案していた [10]．このモデルでは，各層の重みの値はある範囲内に制限されており（層単位の有界重みモデル），層単位で重みを正規化する機能を持つ．学習された重みは一様分布またはベルヌーイ分布に従っているため，学習の後に一度だけ量子化するだけでよい．また，大語彙音声認識実験により，単語正解精度を落とすことなく，パラメータを 4 ビットまで量子化可能なことを確認した．一方，4 ビット量子化における致命的な問題は，高速計算のために必要な巨大な LUT サイズにある．その LUT サイズは 32M バイトを超えており，CPU のキャッシュメモリを有効活用できず，高速計算のネックとなっていた．そのため，より少ないビット数，たとえば 2 ビットで量子化

[1] http://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

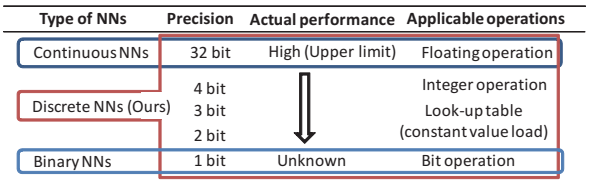


図 1: Properties of neural networks

した DNN が，高速計算に実現に必要である．

本稿では，2 ビット量子化 DNN を達成するため，新たにノード単位の有界重みモデルに基づく学習アルゴリズムと計算機上での実装手法を提案する．2 ビット量子化のキーは，従来研究では層単位で正規化をして量子化を行っていた点にある．実際，各ノードで重みの分布とダイナミックレンジが異なるので，層単位の正規化は量子化誤差を増加する．それゆえ，各ノードに適した重みの範囲で正規化することで，量子化誤差の低減が期待される．ノード単位の有界重みモデルに基づた学習アルゴリズムにより，上記の正規化に適した重みパラメータを学習できる．また，本稿では，LUT に適した重みのエンコード・デコード方法を 2 種類提案するが，そのうち 1 つは LUT サイズを大きく削減できる．提案した学習アルゴリズムと実装方法を，大語彙音声認識実験および DNN フォワード計算の real-time factor (RTF) で検証する．

量子化 DNN の利点としては，量子化ビット数を調整することで，必要な性能で多くのタスクに適応可能な点が挙げられる（図 1）．量子化 DNN は，バイナリ DNN と通常の（連続）DNN の中間的な位置付けである [11, 12]．バイナリ DNN では，すべてのパラメータがバイナリ $\{0, 1\}$ で表現されており，最小の量子化 DNN と言える．一方，著者の知る限り，音声認識でのパフォーマンスは評価されていない．Wang らは，DNN の重みパラメータをベクトル量子化し，軽量な音響モデルを構築している [13] ．音声認識精度を維持し，使用メモリサイズを 90% 削減しているが，実際の処理速度の改善は CPU キャッシュのために課題とされている．

1.2 関連研究

音声認識や画像処理分野では，軽量な DNN を達成可能な量子化以外の方法が提案されており，1) 重み行列の低ランク近似，2) ノード削減，3) 特殊なネットワーク構造，および，4) ノード削除と量子化の組み合わせ，に基づく手法がある．これらの手法は，使用メモリ量の高い削減率と高速な DNN 計算を実現している．主として，標準的な学習手順で構築した DNN に対してパラメータ圧縮処理を施し，圧縮後の DNN パラメータを再学習する流れとなっている．これらの DNN を比較する場合，同じタスクの条件下でメモリ量や速度の相対および絶対的な改善量を比較する必要がある．特に，圧縮前の DNN サイズには注意が必要で，元の DNN パラメータ（例えば，ノード数）が冗長だと，相対的な圧縮率は高くなるからである．

最初の手法は，特異値分解（Singular value decomposition; SVD）[14] や行列分解 [15] を用いて重みパラメータ数を削減する方法で，音声認識実験で評価されている．Xue らは，SVD を中間層の重み行列に適用することで，認識精度の低下なく，使用メモリ量を 80% 削減している [14]．削減率は，重み行列を近似する際に用いる特異値の数に依存し，256 や 196 などが使われている．次の方法は，ノード削減 [16, 17] や模倣学習 [18] であり，音声認識タスクで評価されている．He らは，重みの値に基づいてノードの重要度を定義し，不要なノードを削除することで，中間層のノード数を約 62% 削減している [16]．Li らは，元の DNN を模倣する小さな DNN を学習する方法を提案しているが，精度向上にはより多くの学習データが必要となる [18]．3 番目の方法は，特殊なネットワーク構造に基づくコンパクトな DNN[19] である．彼らは Toeplitz や Vandermonde 行列といった，特殊な行列を重み行列の表現として用いている．これらの行列構造は，行列・ベクトル積といった線形演算の高速化を可能にしている．最後の方法は，ノード削減と量子化の組み合わせにより，Convolutional NN (CNN) と DNN の高い圧縮率を達成しており，画像認識タスクで評価している [20]．彼らは，4 または 5 ビットで量子化しており，また，SVD による圧縮は量子化とノード削減と比べて効果的でないことを示している．

効率的な量子化方法の追求は，他の手法と組み合わせる場合でも，軽量な DNN を達成する上で重要であり，Han らによる実験でも示唆されている [20]．もちろん，ノード削除や DNN の構造も重要であるが，ビットとノードに関する冗長性の性質はやや異なる．例えば，ノード数を最適化した後でも，パラメータの量子化ビット数はある程度の最適化は可能である．本稿では，提案する DNN 量子化の効果を明らかにするため，同じ音声認識タスクにおいて SVD およびノード削減手法との性能も比較する．

2 層単位の有界重みモデルに基づく量子化 DNN

本章では最初に，層単位の正規化をほどこした量子化 DNN のフォワードモデルについて説明する．次に，有界重みモデルと収縮写像に基づくパラメータ学習を説明する．最後に，より小さいビット数へ量子化する際の問題点を説明する．図 2 に，量子化 DNN の学習手順の概要を示す．

2.1 フォワードモデル

DNN のフォワード計算は層 l に関して再帰的に定義される．入力ベクトル $\mathbf{x}_l = [x_{l,1}, ..., x_{l,N_l}]^T \in \mathbb{R}^{N_l}$ はアフィン変換され，活性化関数 $\mathbf{h}_l : \mathbb{R}^{M_l} \to \mathbb{R}^{N_{l+1}}$ が適用される．

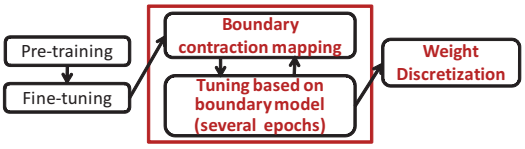


図 2: Training process of discrete NNs

ここで，$\cdot^T$ は転置を，N_l と M_l は l 番目の層への入力ベクトル $\mathbf{x}_l$ と中間ベクトル $\mathbf{z}_l = [z_{l,1}, ..., z_{l,M_l}]^T$ の次元を表す．最終層 L の出力は，初期入力ベクトル $\mathbf{x}_0$ が与えられた元で，層 $l = 0, ..., L-1$ に関する再帰的な計算で得られる．

$$\mathbf{z}_l = \mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l \quad (1)$$

$$\mathbf{x}_{l+1} = \mathbf{h}_l(\mathbf{z}_l) \quad (2)$$

ここで，行列 $\mathbf{W}_l = (w_{l,ij}) \in \mathbb{R}^{M_l \times N_l}$ とベクトル $\mathbf{b}_l = [b_{l,1}, ..., b_{l,M_l}]^T \in \mathbb{R}^{M_l}$ は，第 l 層目の重みとバイアスパラメータである．シグモイド関数とソフトマックス関数が活性化関数 $\mathbf{h}_l$ として用いられる．音響モデルにおいては，入力ベクトル $\mathbf{x}_0$ が音声特徴量，出力ベクトル $\mathbf{x}_L$ が Hidden Markov Model (HMM) の状態に関する音響尤度に対応する [3].

量子化 DNN のフォワード計算は，DNN の重みと中間入力ベクトルが数ビットに量子化された状態で定義される．**この式は，学習時ではなく，認識時のフォーワード計算に用いられる．**第 l 層の中間ベクトル $\mathbf{z}_l$ の i 番目の要素は，層単位の正規化に基づき，次のように計算される．

$$z_{l,i} = \alpha_l \sum_{j=0}^{N_l - 1} \mathcal{F}(\mathcal{Q}_y[y_{l,ij}], \mathcal{Q}_x[x_{l,j}]) + b_{l,i} \quad (3)$$

ここで，$\mathcal{Q}_y$ と $\mathcal{Q}_x$ は，正規化された重み $\mathbf{Y}_l = (y_{l,ij}) \in \mathbb{R}^{M_l \times N_l}$ (i.e. $y_{l,ij} = w_{l,ij} / \max_{i,j} |w_{l,ij}|$) と第 l 層の入力ベクトル $x_{l,j}$ をバイナリや整数にエンコードする関数である．$\mathcal{F}(a, b)$ は，2 つのバイナリ値 a と b のデコードと積演算を行う関数である．α_l は第 l 層の重みの正規化パラメータであり，$\mathbf{Y}_l$, $\mathcal{Q}_y$, $\mathcal{Q}_x$ 及び $\mathcal{F}$ の定義に依存する．

式 (3) における複数の $\mathcal{F}$ と加算演算は，目的の計算機の構造に応じていくつか実現方法がある．例えば，$\mathcal{Q}_y, \mathcal{Q}_x$ が n ビット整数に線形量子化する演算，$\mathcal{F}$ が単純な整数積演算であるとする．もし，SSSE3 のような特殊命令を用いると，8 変数の積和は 3 命令で実現できる．他にも LUT を用いれば，重みと入力ベクトルを結合した索引によって，予め計算された複数変数の積和結果を読み出すだけでよい．

2.2 パラメータ学習

量子化 DNN の学習は，連続領域で次の 2 ステップを何回か繰り返して行われる．1) 逆誤差伝播法 [21]，2) 重み境界の収縮．これ以降，登場するパラメータの初期値は，すでにファインチューニングされた DNN のものだと仮定する．

量子化に適したパラメータを学習するために，式 (3) に類似した重みの制約を導入する．潜在的なパラメータ行列 $\mathbf{V}_l = (v_{l,ij}) \in \mathbb{R}^{M_l \times N_l}$ を用いて，フォワード計算を次のように表現する．

$$\mathbf{z}_l = \alpha_l \mathbf{g}_l(\mathbf{V}_l)\mathbf{x}_l + \mathbf{b}_l, \quad (4)$$

ここで，$\mathbf{g}_l(\mathbf{x}) = (g_l(x_{ij}))$ は，$\tanh(x)$ のような値域が $[-1, 1]$ であるような，要素毎の有界な関数行列である．また，$\tanh(x)$ の導関数は $1 - \tanh^2(x)$ であり，$y_{l,ij} = g_l(v_{l,ij})$ を満たす．

パラメータは通常の NN の学習と同様に，コスト関数 E と教師信号 $\mathbf{r} = [r_1, ..., r_{N_L}]^T \in \mathbb{R}^{N_L}$ に基づく逆誤差伝播法に基づいて最適化される．本稿では，E としてクロスエントロピーを使う．初期誤差ベクトル $\boldsymbol{\epsilon}_L = \left(\frac{\partial E}{\partial \mathbf{x}}(\mathbf{r}, \mathbf{x}_L)\right)$ を計算した後，次のように $l = L-1, ..., 0$ に対する各パラメータを更新する:

$$\boldsymbol{\delta}_l = \left(\frac{\partial \mathbf{h}_l^T}{\partial \mathbf{z}}(\mathbf{z}_l)\right) \boldsymbol{\epsilon}_{l+1}, \quad (5)$$

$$\boldsymbol{\epsilon}_l = \alpha_l \mathbf{g}(\mathbf{V}_l)^T \boldsymbol{\delta}_l, \quad (6)$$

$$\alpha_l \leftarrow \alpha_l - \eta \boldsymbol{\delta}_l^T \mathbf{g}_l(\mathbf{V}_l)\mathbf{x}_l, \quad (7)$$

$$\mathbf{V}_l \leftarrow \mathbf{V}_l - \eta \frac{\partial \mathbf{g}_l}{\partial \mathbf{V}}(\mathbf{V}_l) \circ \left(\alpha_l \boldsymbol{\delta}_l \mathbf{x}_l^T\right), \quad (8)$$

$$\mathbf{b}_l \leftarrow \mathbf{b}_l - \eta \boldsymbol{\delta}_l, \quad (9)$$

ここで，$\circ$ は行列の要素毎の積を表す演算子であり，η は学習の速度と精度を制御する学習係数である．$\boldsymbol{\epsilon}_l$ は逆誤差伝播のための第 l 層における誤差ベクトルである．なお，我々の有界重みモデル [10] は，Kim らが議論したバイナリ NN の学習モデルの一部と類似している [12].

正規化パラメータ α_l を小さくするため，適当なエポック毎に次の境界収縮写像を行う．これは過去の実験から得られた次の事実によって導入している．1) 小さな α_l は式 (3) の量子化誤差を減少させる，2) 学習後の重み $w_{l,ij}$ の分布が非ガウス分布，特に線形量子化 $\mathcal{Q}_y$ に有効に働くベルヌーイ分布に近くなる．

$$\alpha_l \leftarrow \max_{i,j} |w_{l,ij}|, \quad (10)$$

$$v_{l,ij} \leftarrow w_{l,ij} / \max_{i,j} |w_{l,ij}|, \quad (11)$$

ここで，$\mathbf{W}_l = \alpha_l \mathbf{g}_l(\mathbf{V}_l)$ である．g に tanh を用いるので，この操作で α_l は単調に減少する．

2.3 問題点

層単位の有界重みモデルに基づく学習は，音声認識精度を下げることなく 4 ビットの量子化 DNN まで実現できた．しかし，LUT を用いたフォワード計算の高速化は非現実的であった．なぜなら，4 ビット量子化では LUT に

必要なメモリ量が 32M バイトとなり，一般的な CPU のキャッシュメモリサイズを大きく超えているからである．キャッシュメモリを活用するには，より小さなビット数での量子化が不可欠である．

実際の重みのダイナミックレンジと分布はノード毎に異なっている．そのため，層単位で重みを正規化すると情報損失が大きくなり，小さなビット数では量子化誤差が増大する．その改善には，学習と認識において，ノード毎に異なる重みのダイナミックレンジを考慮する必要がある．

3 ノード単位の有界重みモデルに基づく量子化 DNN

本研究では，重みのダイナミックレンジと分布を均一化可能な，ノード単位の有界重みモデルに基づく学習アルゴリズムと実装方法を提案する．はじめに，フォワード計算におけるノード単位での重みの正規化と，学習のためのノード単位での有界重みモデルを説明する．次に，任意の計算機構造で適用可能な LUT を用いた実装方法に関しても議論する．最後に，量子化対象の層に関して議論を行う．

3.1 フォワードモデルとパラメータ学習

本研究では，層単位ではなく，ノード単位で重みのダイナミックレンジをより適切に設定する．そのため，α_l の代わりに，ノード毎に異なる正規化パラメータ $\boldsymbol{\lambda}_l = [\lambda_{l,1}, ..., \lambda_{l,M_l}]^T$ を導入する．ノード単位の正規化は次式で表現される．

$$z_{l,i} = \lambda_{l,i} \sum_{j=0}^{N_l-1} \mathcal{F}(\mathcal{Q}_y[y_{l,ij}], \mathcal{Q}_x[x_{l,j}]) + b_{l,i}. \quad (12)$$

正規化パラメータの対角行列表記 $\boldsymbol{\Lambda}_l = \mathrm{diag}(\lambda_{l,1}, ..., \lambda_{l,M_l})$ を用いることで，ノード単位の有界モデルは次のように表現できる．

$$\mathbf{z}_l = \boldsymbol{\Lambda}_l \mathbf{g}_l(\mathbf{V}_l)\mathbf{x}_l + \mathbf{b}_l. \quad (13)$$

各パラメータは確率勾配に基づき逆誤差伝播法で学習される．新たに導入したパラメータに関する更新則は次のように導出される．

$$\boldsymbol{\delta}_l = \left(\frac{\partial \mathbf{h}_l^T}{\partial \mathbf{z}}(\mathbf{z}_l)\right) \boldsymbol{\epsilon}_{l+1}, \quad (14)$$

$$\boldsymbol{\epsilon}_l = \boldsymbol{\Lambda}\mathbf{g}(\mathbf{V}_l)^T \boldsymbol{\delta}_l, \quad (15)$$

$$\boldsymbol{\lambda}_l \leftarrow \boldsymbol{\lambda}_l - \eta \boldsymbol{\delta}_l \circ (\mathbf{g}_l(\mathbf{V}_l)\mathbf{x}_l), \quad (16)$$

$$\mathbf{V}_l \leftarrow \mathbf{V}_l - \eta \frac{\partial \mathbf{g}_l}{\partial \mathbf{V}}(\mathbf{V}_l) \circ \left(\boldsymbol{\Lambda}_l \boldsymbol{\delta}_l \mathbf{x}_l^T\right). \quad (17)$$

以上の式より，式 (7) では 1 つの α_l に集約していた伝播誤差ベクトル δ_l が，それぞれの正規化パラメータに影響していることがわかる．$\mathbf{W}_l = \boldsymbol{\Lambda}_l \mathbf{g}_l(\mathbf{V}_l)$ における境界収縮は同様に修正され，

$$\lambda_{l,i} \leftarrow \max_j |w_{l,ij}|, \quad (18)$$

$$v_{l,ij} \leftarrow w_{l,ij} / \max_j |w_{l,ij}|. \quad (19)$$

となる．

3.2 ルックアップテーブルを用いた実装方法

LUT の実装方法は，エンコード関数 $\mathcal{Q}_x$ がバイナリ値 $\{0, 1\}$ を出力するか否かで 2 種類に分けられる．初めに，一般的なモデルで LUT のエンコード・デコード方法を説明する．そのあと，特別なバイナリモデルの場合について議論する．

3.2.1 一般モデル

説明のため，まず，重みの行ベクトルと入力ベクトルをいくつかのグループに分け，各グループで D 個の要素を持っていると仮定する．それらを用いたバイナリコードをそれぞれ次のように記述する．

$$\bar{\mathbf{y}}_{(l,i,k)} = \{\mathcal{Q}_y[y_{l,ij}]\}_{j=Dk}^{D(k+1)-1}, \quad (20)$$

$$\bar{\mathbf{x}}_{(l,k)} = \{\mathcal{Q}_x[x_{l,i}]\}_{i=Dk}^{D(k+1)-1}. \quad (21)$$

LUT，T，は，以上のバイナリコードの組み合わせで参照される．例えば，$\bar{\mathbf{y}}_{(l,i,k)} = [110, 101]$ と $\bar{\mathbf{x}}_{(l,k)} = [010, 101]$ の場合，LUT のインデックスはバイナリ表現で [110101010101] となる．式 (12) から，量子化後のフォワード計算は次のように表現できる．

$$z_{l,i} = \lambda_{l,i} \sum_{k=0}^{N_l/D} T[\bar{\mathbf{y}}_{(l,i,k)}\bar{\mathbf{x}}_{(l,k)}] + b_{l,i}. \quad (22)$$

加算回数は明らかに $1/D$ に削減されている．

LUT は事前にすべてのバイナリコードのパターンを予め計算することで構築する．$\bar{\mathbf{a}}, \bar{\mathbf{b}}$ のすべてのパターン, 例えば, 0 から $2^n - 1$ を具体的に列挙して構築する．

$$T[\bar{\mathbf{a}}\bar{\mathbf{b}}] = \sum_{j=0}^{D-1} \mathcal{F}(a_j, b_j) = \sum_{j=0}^{D-1} \mathcal{Q}_y^{-1}[a_j]\mathcal{Q}_x^{-1}[b_j] \quad (23)$$

ここで，a_j と b_j は $\bar{\mathbf{a}}$ と $\bar{\mathbf{b}}$ に対応する第 j 要素を表現する．$\mathcal{Q}_y^{-1}$ と $\mathcal{Q}_x^{-1}$ は，$\mathcal{Q}_y$ と $\mathcal{Q}_x$ に対応するデコード関数であり，バイナリパターンの連続値表現を返す.

$g_l(x) = \tanh(x)$ で，$h_l(x) = 1/(1 + \exp(-x))$ である場合には, n ビットの各デコード・エンコード演算は次のように定義される．

$$\mathcal{Q}_x[x] = \mathrm{floor}[(2^n - 1)x + 0.5], \quad (24)$$

$$\mathcal{Q}_y[y] = \mathrm{floor}[(2^n - 1)(y + 1)/2 + 0.5], \quad (25)$$

$$\mathcal{Q}_x^{-1}[x] = x/(2^n - 1), \quad (26)$$

$$\mathcal{Q}_y^{-1}[y] = 2y/(2^n - 1) - 1. \quad (27)$$

なお，本稿では LUT 自体の最適化は扱わない．

表 1: Memory requirement per layer in bytes

	for weights	for LUT
32-bit float	$4N_lM_l$	–
8-bit SSSE	N_lM_l	–
4-bit general LUT	$N_lM_l/2$	2^{8D}
3-bit general LUT	$N_lM_l3/8$	2^{6D}
3-bit-bin binary LUT	$N_lM_l3/8$	2^{3D}
2-bit general LUT	$N_lM_l/4$	2^{4D}
1-bit general LUT	$N_lM_l/8$	2^{2D}

3.2.2 バイナリモデル

もし，$\mathbf{x}_l$ にバイナリ量子化を採用するなら，テーブルサイズはビットマスク演算により半分にできる．例えば，$\bar{\mathbf{y}}_{(l,i,k)} = [110, 101]$ と $\bar{\mathbf{x}}_{(l,k)} = [0, 1]$ の場合，T のインデックスは通常は [11010101] となる．デコード関数が $Q_y^{-1}[000] = 0$, $Q_x^{-1}[0] = 0$ かつ $Q_x^{-1}[1] = 1$ を満たす限り，$\bar{\mathbf{y}}_{(l,i,k)}$ のビットを $\bar{\mathbf{x}}_{(l,k)}$ でマスクすることで，バイナリモデルの LUT, T_b, のインデックスは [000101] とできる．これは次式により理解できる．

$$\begin{aligned} T[11010101] &= Q_y^{-1}[110]Q_x^{-1}[0] + Q_y^{-1}[101]Q_x^{-1}[1] && (28)\\ &= 0 + Q_y^{-1}[101] && (29)\\ &= Q_y^{-1}[000] + Q_y^{-1}[101] && (30)\\ &= T_b[000101]. && (31) \end{aligned}$$

なお，このバイナリモデルにおける $\mathcal{Q}_y$ と $\mathcal{Q}_y^{-1}$ の定義は，式 (25) と (27) から適切に変更する．使用メモリ量は削減できるが，量子化誤差の増加と音声認識率の低下を招く可能性があるため，その効果は実験的に確認する必要がある．

3.2.3 使用メモリ量の比較

表 1 に，各実装方法における DNN 重みと LUT のメモリ使用量を示す．*32-bit float* と *8-bit SSSE* は，通常の浮動小数点の積和演算と SSSE3 命令セットによる方法で，LUT 自体を必要としない．*3-bit-bin (binary LUT)* はバイナリモデルの LUT を意味し，その他の項目は一般モデルの LUT の結果を示している．*32-bit float* では $4N_lM_l$ のメモリが重みに必要であるが，他の方法ではそれの 1/8 以下のメモリ量で済む．なお，LUT のサイズはパラメータ D にも依存する．全体のメモリ使用量は，実際の CPU キャッシュサイズを考えると，1, 2 M バイト以下が理想的である．

3.3 量子化対象の層の選択

本研究では，量子化したフォワード計算はすべての層に適用しない．というのは，少なくとも入力層への入力は $[0, 1]$ の範囲に収まっていないからである [8, 10]．また，重みを量子化する層を選択することで，音声認識性能を改善できる可能性はある．以前の報告では，入力層を除いたすべての層の重みを量子化していた．最終層の出力は識別に用いられるので，本稿では最終層の重みも量子化を行わず浮動小数点で表現する．

表 2: Configuration

Item	Value
Audio data	16 bits, 16 kHz sampling
STFT analysis	hamming window: 25 ms, shift: 10 ms
Features for GMM	MFCC 39 dim. [13+ Δ 13 + ΔΔ 13]
Features for DNN	FBANK 825 dim. [(25+ Δ 25 + ΔΔ 25) × 11 frames]
Language model	3-gram statistical 65000 words
GMM-HMM	3-state tri-phone 4000 tied-states 32 mixtures
# of DNN layer (L)	7
DNN network size	input layer: 1024 × 825 middle layer: 1024 × 1024 output layer: 4000 × 1024
Training set	clean speech 223 hours (799 males and 168 females)
Test set	clean speech 3.5 hours (15 males and 5 females)

4 評価実験

4.1 実験設定

日本語話し言葉コーパス（CSJ）[22] を用いた連続音声認識実験により，ノード単位の有界重みモデルに基づく音響モデルを評価した．まず，n ビット量子化におけるノード単位・層単位モデルの単語正解精度（WA）を比較する．この時，学習フェーズ (Eq.(4) or Eq.(13)) と，認識フェーズ (Eq.(3) or Eq.(12)) のそれぞれにおいて，ノード単位・層単位のモデルの各組み合わせにおける正解精度を調査する．正規化した重みの平均量子化誤差，$|y - \mathcal{Q}_y^{-1}[\mathcal{Q}_y[y]]|$ と，ノード単位の重み統計量として正規化尖度 $\mathbb{E}[(x - \mathbb{E}[x])^4]/\mathbb{E}[(x - \mathbb{E}[x])^2]^2 - 3$ に関しても議論する．$\mathbb{E}$ は期待値演算を表す．LUT の一般・バイナリモデルの各実装におけるフォワード計算のリアルタイムファクタ (RTF) も比較した．RTF は次式で定義される．

$$\text{RTF} = (\text{processing time})/(\text{data duration}). \qquad (32)$$

ノード削減 DNN [16] と SVD-DNN[14] の WA，RTF およびメモリ使用量も調査した．

DNN 音響モデルの学習用データとして，約 223 時間分の学術講演音声を用いた．評価用データは，CSJ テストセット 1 および 2 の約 3.5 時間分，男性 15 名・女性 5 名分の音声である．言語モデルの学習データは，評価用データを除いた CSJ に含まれるすべての書き起こしテキストを用いた．語彙サイズは 65000 であり，トライグラム言語モデルを用いた．音声認識器は Julius (ver. 4.3.1) [23] を用い，言語モデル重みと挿入ペナルティはデフォルト値である 8 と −2 に設定した．また，ビーム幅は 4000 にした．

表 3: Word accuracy (WA) vs. quantization bits for clean speech task

		Applied model		Discrete layer l	Word accuracy (%)							
		Training	Recog.		1-bit	2-bit -bin	2-bit	3-bit -bin	3-bit	4-bit	8-bit	32-bit float
Normal Training		–	layer-wise	1-6	–	–	-2.24	–	-1.16	-3.35	80.63	81.86
		–	layer-wise	1-5	1.10	–	2.17	–	**2.80**	**49.54**	81.74	
		–	node-wise	1-6	–	–	2.13	–	57.47	79.07	81.82	
		–	node-wise	1-5	1.07	0.69	2.17	1.33	**62.35**	**79.83**	81.82	
Baseline	P0	layer-wise	layer-wise	1-5	2.98	–	77.78	–	80.07	81.07	81.32	81.33
Proposed	P1	layer-wise	node-wise	1-5	2.73	41.17	77.55	59.45	80.82	81.47	81.36	
	P2	node-wise	layer-wise	1-5	2.06	–	44.35	–	66.13	80.14	81.52	81.53
	P3	node-wise	node-wise	1-5	1.45	58.45	**79.37**	**61.40**	**81.05**	81.10	81.52	
	P4	re-train. (2-bit quan.)		1-5	–	–	**80.15**	–	–	–	–	

DNN 音響モデルは，GMM-HMM を用いたラベル付与とフレーム単位での識別に基づき学習した．初めに，混合数 32・共有状態数 4000 のトライフォン GMM-HMM を HTK (Hidden Markov Model Toolkit) [2]を用いて学習した．音声特徴量は，13 次元の Mel-frequency cepstral coefficients (MFCCs)，その 1 次差分および 2 次差分の計 39 次元である．音声データのサンプリング周波数は 16kHz であり，窓長 25 ミリ秒・シフト長 10 ミリ秒で短時間フーリエ解析を行った．また，発話単位毎に平均・分散正規化を行っている．DNN-HMM は，GMM-HMM と同じ HMM パラメータを利用し，DNN の中間層のノード数は 1024 で，$L = 7$ とした．出力ノードの次元は，HMM の状態を識別するため 4000 に設定した．DNN への入力特徴量は，基本特徴量における中心フレームと前後 5 フレームを連結した計 11 フレームの 825 次元となっている．その基本特徴量は，25 次元の対数メルフィルタバンク係数と 1 次差分，2 次差分で構成される．平均・分散正規化を同様に適用した．これらの設定を表 2 にまとめる．最後に，GMM-HMM に基づくビタビアライメントにより得られた状態ラベルを用いて，DNN パラメータを学習した．プレトレーニングは段階的に識別学習を行う方法を用い [3]，学習係数の半自動調整を行うため AdaGrad [24] を用いた．ミニバッチサイズは 64 に設定した．ファインチューニングの後，有界モデルを用いた学習を行った．重みの量子化後，入力層などの量子化を行わなかった層のパラメータの再学習も検討した．ドロップアウト [25] も DNN の絶対的な性能を改善する可能性があるが，Kim らが議論しているように [7]，その有無で手法間の相対的な差には致命的な影響を及ぼさないと考えている．

ノード削減 DNN と SVD-DNN においても，初期値に用いる DNN のパラメータ値，構造，特徴量などは提案法の場合と同様にした．ノード削減 DNN は，出力重みノルム基準 [16] によって行い，中間層のノード削減率は 10, 20, 40, 60, 80% に設定した．SVD-DNN [14] も，SVD は中間層の重み $\mathbf{W}_l$ $(l = 1, ..., 5)$ に対して適用し，用いる特異値数は 32, 64, 128, 256, 512 および 768 に設定した．

[2]http://htk.eng.cam.ac.uk/

従来研究と同様，性能改善のため，ノード削除と SVD を適用した後に，DNN パラメータを再学習した．

4.2 実験結果

4.2.1 単語正解精度

表 3 に，異なる量子化ビット数に対する WA を示す．*Normal Training* の行は，有界重みモデルではなく，通常の DNN モデルで学習したパラメータでの認識結果を表す．*Discrete layer* の列は，量子化対象の層を表している．*2-bit-bin* と *3-bit-bin* は，重みは 2・3 ビットに量子化し，中間層への入力ベクトルを 1 ビットに量子化した場合の結果を示す．*2-bit* などの他の表記では，重みと中間層への入力ベクトルの量子化ビット数は同じである．*32-bit float* は量子化していない浮動小数点で表現された重みを意味する．なお，GMM-HMM の単語正解精度は 75.8 % であった．

まず，量子化ビット数と各手法の WA の関係に関して述べる．8 ビット量子化ではすべての手法はほぼ同程度の性能であり，より小さいビット数では性能差が明確になっている．表中の *Normal Training* に着目すると，通常モデルで学習し，認識時に層単位の正規化を行った場合（*Recog.*），4 ビット量子化以下では認識に失敗していることがわかる．認識時にノード単位の正規化を適用するだけで，WA は大きく改善しており，ノード単位の正規化の重要性が示唆される．量子化対象の層を制限すると，3 ビットのようなシビアなビット数においても WA を改善している．これ以降は，層 $l = 1, ..., 5$ における重みに量子化を行った結果を比較する．

次に，ノード単位での有界重みモデルと各 LUT の実装を用いた場合の WA に着目する．ノード単位モデルの WA は，P2 のミスマッチ状況を除き，2・3 ビットの量子化において，ベースラインの WA を最大 1.8 ポイント上回っている．特に，P3 では 2 ビット量子化においても十分に高い精度を保っており，通常の DNN と比較して 2 ポイントしか WA は低下していない．P3 から量子化を行っていない層のパラメータ再学習することで，少し性能は改善している (P4)．*2-bit-bin* および *3-bit-bin* の WA は，中

表 4: Quantization error (QE) vs. quantization bits. Discrete layers were 1-5 in all cases for clean speech task.

		Applied model		Quantization error				
		Training	Recognition	1-bit	2-bit	3-bit	4-bit	8-bit
Normal training		–	layer-wise	9.13E-01	2.50E-01	8.42E-02	3.43E-02	1.96E-03
		–	node-wise	8.31E-01	**1.98E-01**	**7.25E-02**	3.33E-02	1.96E-03
Baseline	P0	layer-wise	layer-wise	3.10E-01	1.89E-01	8.16E-02	3.29E-02	1.97E-03
Proposed	P1	layer-wise	node-wise	2.26E-01	**1.27E-01**	**7.25E-02**	3.58E-02	1.96E-03
	P2	node-wise	layer-wise	7.24E-01	1.38E-01	6.83E-02	3.35E-02	1.96E-03
	P3, P4	node-wise	node-wise	2.92E-01	1.52E-01	7.85E-02	3.44E-02	1.96E-03

表 5: Computer specifications for forward calculation

OS	Ubuntu 14.04.2 LTS
CPU	Intel Core i5-4690 3.50GHz
Memory	32GB
Cache	6M

間層への入力ベクトルを $\{0, 1\}$ に量子化したにも関わらず 50% 以上となった．*2-bit-bin* では，層単位モデルから約 16 ポイント WA が改善しており，ノード単位モデルに基づく学習の優位性を示している．しかし，1 ビットに重みを量子化する（*1-bit*）と全く認識されず，2 ビットと 1 ビット量子化の間に大きな隔たりがある．1 ビット量子化 DNN を実現するには，より精度を保てるパラメータ学習と量子化方法が必要となる．

4.2.2 量子化誤差および重み分布

量子化誤差 (QE) と WA の関係を理解するため，各ビット数における平均量子化誤差を表 4 に示す．normal training と baseline (P0) の QE は，認識時でノード単位の正規化を行うことで減少している．しかし，低い量子化誤差が必ずしも高い WA に繋がっているとは限らない．例えば，P1 における QE は P0 と比較して改善しているが，それらの WA は同じ程度である．

次に，通常学習，層単位およびノード単位の有界重みモデルに基づく学習の違いを分析するため，重みに対するノード単位の尖度の分布に注目 s する．ガウス分布，一様分布，ベルヌーイ分布の尖度はそれぞれ 0, -1 および -2 である．図 3 に，学習後における normal training，P0 (baseline, layer-wise) および P3 (node-wise) の尖度の分布を示す．normal training の尖度は 0 以上であるため，重みはガウス分布やよりスパースな分布をしていることがわかる．一方，baseline と提案モデルの重み分布はベルヌーイ分布に近い．というのは，ほとんどの尖度が -1 よりも小さいからである．また，baseline の尖度分布には提案モデルの方にはない複数のピークが見られる．この内，比較的尖度が大きいピークは，量子化 DNN の中間出力における誤差を増大させ，その誤差の範囲や分布も各ノードで異なると考えられる．このような誤差が途中で打ち消し合うこともなく蓄積していき，HMM や言語モデルで平滑化可能な範囲を超え，層単位での正規化方法の認識精度低下を招いたと推察される．ノード単位の有界重みモデルに基づく学習は，各ノードの重みの分布を均一にし，連続値と量子化した値の隔たりを埋め，そのような状況を避けられたと考えられる．

最後に，層単位およびノード単位モデルの実際の重み分布を図 4 に示す．通常学習 (normal training) の重み分布はガウス分布に似ているが，有界モデルを用いた学習された重みの分布はベルヌーイ分布に近くなっている．-1 と 1 付近にある 2 つのピークは，正規化パラメータ $\alpha_l, \lambda_{l,i}$ が小さくなるにつれて，外側へ移動する傾向にある．それゆえ，それらは分布の形状を制御していると考えられる．

4.2.3 LUT を用いたフォワード計算の RTF

LUT における一般的モデル（*general model*）とバイナリモデル（*binary model*）の 2 つの実装方法を用いて，量子化 DNN のフォワード計算の速度向上率を明らかにする．また，重みと LUT の使用メモリ量も同時に示す．計算機のスペックは表 5 に示す通りである．CPU のキャッシュサイズと周波数は LUT を用いた計算に十分な値である．

RTF と LUT・重みに必要なメモリ量の関係を表 6 に示す．表中では 2 つの RTF が示されているおり，1 つは中間層 ($l = 1, ..., 5$) に対する RTF(partial)，もう一つは DNN 処理全体 ($l = 0, ..., 6$) に対する RTF(all) である．これは，量子化が中間層の重みに対して適用されており，実質的な速度改善はこれらの層（$l = 1, ..., 5$）に限られるからである．*32-bit float* は，LUT や特殊命令セットを用いない通常の積和実装を表し，*8-bit SSSE* は [8] で提案された SSSE 命令セットを用いた実装である．*baseline* は 4 ビットの量子化 DNN の結果であり，層単位モデルの限界値である．表中の D は，式 (22) において同時に計算される変数の数を意味する．

$D = 3, 4$ において，一般・バイナリの両実装モデルに対する RTF(all) は，baseline と比較して約 30〜40% の処理速度向上を達成し，SSSE 命令を用いた実装に近い処理速度となっている．さらに，3-bit-bin で $D = 7$ の RTF は，WA は十分ではないものの，SSSE と同等の処理速度となっている．しかし，3 ビットの量子化では 4M バイトの LUT を必要とする．もし，2 ビット量子化かつバイナリモデルを適用し，$D = 8$ に設定すると，SSSE と同等

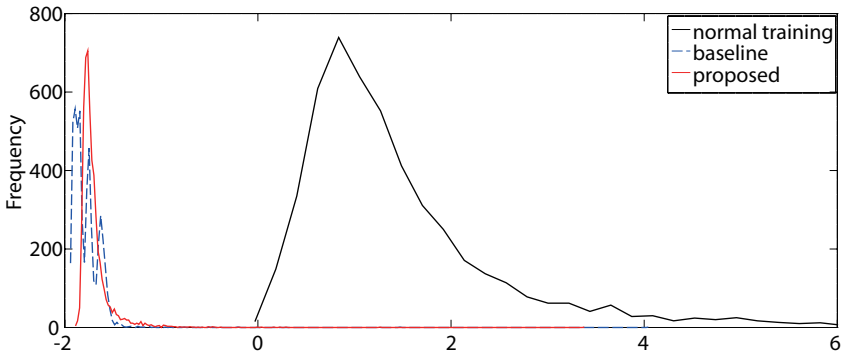


図 3: Kurtosis distributions

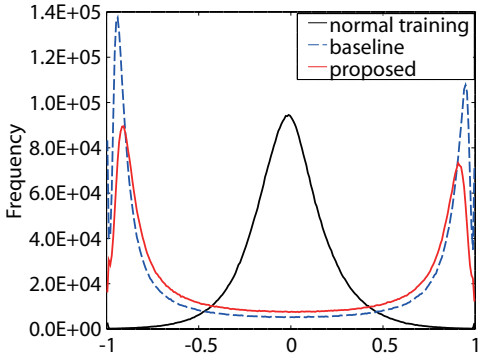


図 4: Weight distributions

表 6: Real-time factor of forward calculation and memory usage on single CPU

	Methods	RTF (partial) ($l = 1, ..., 5$)	RTF (all) ($l = 0, ..., 6$)	D	Memory usage (bytes) LUT	Memory usage (bytes) total weights ($l = 1, ..., 5$)
Standard	32-bit float	**0.435**	**0.841**	1	–	20 M
Baseline	**4-bit (general LUT)**	0.653	**1.060**	3	**32 M**	2.5 M
	4-bit (general LUT)	0.220	**0.641**	2	**128 K**	2.5 M
Proposed	3-bit (general LUT)	0.255	0.661	3	512 K	1.85 M
	3-bit-bin (binary LUT)	0.096	0.501	7	4 M	1.85 M
	2-bit (general LUT)	**0.111**	**0.517**	4	**128 K**	**1.25 M**
	1-bit (general LUT)	0.084	**0.489**	8	**128 K**	**640 K**
Special	8-bit SSSE	0.092	**0.498**	8	–	5 M

の処理速度を達成しつつ，LUT サイズは 512K バイトに抑えられる．1 ビット量子化 DNN は 8-bit SSSE の処理速度を上回っているので，その WA の改善が期待される．中間層だけの RTF(partial) に注目すると，DNN の量子化ビット数に比例して，RTF が改善していることがわかる．例えば，2 ビット実装の RTF(partial) は，32 ビット実装から約 75% 処理速度の向上がみられる．ハードウェア上での論理回路を用いた実装では，さらなる RTF の改善が期待される

4.2.4 ノード削減 DNN と SVD-DNN との比較

表 7 と表 8 に，ノード削減 DNN と SVD-DNN の WA，メモリ使用量および RTF を示す．これらの表でも 2 種類の RTF が示されており，1 つは中間層 ($l = 1, ..., 5$) に対する RTF(partial)，もう一つは DNN 処理全体 ($l = 0, ..., 6$) での RTF(all) である．再学習を行わない場合の WA は，すべての設定において元の DNN よりも低くなっている．SVD-DNN での性能低下はノード削減 DNN よりも深刻でない．これは，SVD は元の重み行列の値を保つような次元圧縮方法だからである．多くの場合，再学習の後で WA は 81% まで改善している．

まず，各手法の使用メモリ量と WA の関係について比較する．量子化 DNN では約 1.4M バイトの使用メモリ量で WA 80% を達成していたが，ノード削減 DNN および SVD-DNN では，3M のような少ない使用メモリ量の場合，WA は 80% 未満となっている．量子化 DNN と同等の使用メモリ量および RTF(partial) を達成するには，さらにノード削減率を高くする必要があり，使用する特異値数も極端に減らす必要がある．例えば，これまで報告されているような 80% の圧縮率を達成するためには，ノード削減率では 60% 程度必要であり，特異値数では 64 に設定する必要がある．しかし，それらの設定での WA は 80% を下回っている．この結果は，本研究での用いた中間層におけるノード数が他の報告で用いられている数，例えば 2048，よりも相対的に小さい点が一因している．以上より，使用メモリ量の削減において，量子化の効果は大きいと言える．

次に，各手法の RTF に注目する．ノード削減法は，通常多くのノードを含む出力層の重み行列の次元を大きく削減可能なので，RTF(all) は量子化 DNN と比較すると大きく改善している．量子化ビット数は，DNN のノード数を最適化した後でも，最適化するとは可能であるため，ノード削除と量子化の組み合わせは，DNN の構造を改善する効率的な戦略であるといえる [20]．そのような場合でも，量子化の方法は重みの圧縮率と認識性能に影響を与えるため，量子化アルゴリズムの改善は不可欠である．

5 議論

実験結果から，WA，量子化誤差および重みの統計量に関して，1) 量子化誤差と WA の関係と 2) 重みの正規化の効果に関する事実を得た．前者は，DNN のパラメータ学習の戦略に影響を与える．QE は必ずしも直接的に WA を改善するわけではないので，パラメータ学習のコストに

表 7: The Performance of Node-pruning: word accuracy, memory usage, and RTF

Pruning-ratio	Word Accuracy (%) w/o retrain	Word Accuracy (%) w/ retrain	Memory usage (bytes) ($l = 1, ..., 5$)	RTF (partial) ($l = 1, ..., 5$)	RTF (all) ($l = 0, ..., 6$)
80%	0	74.48	0.8 M	0.018	0.101
60%	0	78.29	3.2 M	0.070	0.233
40%	0.56	**80.28**	**7.2 M**	**0.155**	**0.397**
20%	1.35	81.29	12.8 M	0.274	0.592
10%	58.06	81.64	16.2 M	0.346	0.702

表 8: The Performance of SVD-DNN: word accuracy, memory usage, and RTF

SVD dim.	Word Accuracy (%) w/o retrain	Word Accuracy (%) w/ retrain	Memory usage (bytes) ($l = 1, ..., 5$)	RTF (partial) ($l = 1, ..., 5$)	RTF (all) ($l = 0, ..., 6$)
32	4.35	48.81	1.25 M	0.034	0.431
64	3.35	79.74	2.5 M	0.056	0.456
128	2.96	**80.52**	**5.0 M**	**0.111**	**0.508**
256	11.41	81.78	10.0 M	0.215	0.611
512	79.36	81.81	20.0 M	0.421	0.817
768	81.62	82.01	30.0 M	0.628	1.024

QE を最小化するような制約を加えるだけでは，効果的でないと考えられる．後者は，量子化 DNN に対する他の重み制約の可能性を示唆している．もし，WA を改善するためにこの事実を直接的に用いるのであれば，ノード単位の重みの尖度の平均を最小化するアプローチもある．LUT の最適化や，重みの非線形量子化なども，1 ビット量子化を含む低ビット量子化 DNN の WA 改善に効果的である．また， sequence training も WA の向上に効果があると推察される．なぜなら，HMM と言語モデルによるフィルタ効果により，量子化誤差の影響は小さくなるからである．量子化後のフレーム単位での状態識別率はとても低い，という事実もこの仮説を支持している．

バイナリモデル LUT に基づく実装も改善の余地があり，その実現可能性は残されている．提案したノード単位のモデルは，中間層への入力ベクトルの量子化を考慮していないため，その影響を学習に組み込むことで性能改善が可能だと考えられる．理想的には，そのベクトルの要素もベルヌーイ分布に従っていることである．そのような分布の形成に適した制約を，発見または開発することが必要である．これは，単に活性化関数で用いられるシグモイド関数などのスケールパラメータを調整するだけで十分な可能性もある．以上の課題を解決すれば，2 ビット重み量子化とバイナリモデル LUT に基づいた実装が実現できる．

最後に，DNN の本質的な「記憶」力も，より進んだ研究を行うために考慮すべきである．例えば，もし，巨大なネットワークを中間層に用いると，従来研究 [9] でも述べられているように，重みの量子化はより簡単になると予想される．彼らの研究では，10 種の数字画像の識別精度を，より多くの中間層のノード数を用いることで改善できたと報告している．そのため，ノード数と重み表現に必要なビット数，必要なメモリ量および演算効率などの関係をより詳細に調べる必要がある．また，雑音や残響音声を含むような，入力データのパターン数が増加する場合，それらを記憶または識別するためのより多くの「記憶容量」が必要である．そのため，ヘテロなデータを扱う場合に，クリーン音声用に設計された DNN 構造を流用すると，少ないビット数での量子化はより難しくなると考えられる．使用メモリ量と計算効率の観点でより一般的に最適化を行うには，入力データと演算の実装方法に合わせた DNN 構造の適応（ノード削減など）と量子化を同時に扱う枠組みが不可欠である．

6 結論

本研究の目的は，省メモリ・低計算コストな音響モデルのための量子化 DNN の開発である．4 ビットの量子化では LUT サイズが巨大となるので，使用メモリ量の観点で 2 ビットでの量子化が必要となった．2 ビット量子化のカギは，重みパラメータのバラつきの正規化方法にあった．本研究では，層単位ではなく，ノード単位の有界重みモデルに基づいてパラメータを学習するアルゴリズムを開発した．このアルゴリズムは，ノード単位で重み分布のバラつきを均一化し，量子化に有効な重みの学習を可能にした．また，量子化 DNN の実装として，メモリ使用量が異なる，一般的な LUT およびバイナリ LUT に基づくの 2 種類の実装方法を提案した．2 ビット量子化 DNN の評価実験では，高い単語正解精度を保ちつつ，DNN のフォワード計算の計算速度を約 40% 向上した．

残された主な課題は，メモリ量と処理速度を改善するために，ノード削減法との組み合わせや中間層への入力ベクトルの 1 ビット量子化である．そのカギとなるアプローチは sequence training，重みの非線形量子化などがあり，これらの技術を用いてより省メモリかつ低計算コストな DNN 構築を目指す予定である．

謝辞

本研究は JSPS 科研費 15K16051 の助成を受けたものです.

参考文献

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocaburary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 82–97, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Geroge, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and othres, "Deep neural networks for acuostic modelling in speech recognition," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] F. Seide, G. Li, , and X. Chen D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transaction," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 24–29.

[4] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using contex-dependent deep neural network," in *Proceedings of the Interspeech 2011*, 2011, pp. 437–440.

[5] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 873–880.

[6] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[7] J. Kim, K. Hwang, and W. Sung, "X1000 real-time phoneme recognition VLSI using feed-forward deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 7510–7514.

[8] V. Vanhouche, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Proceedings of the Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011, vol. 1.

[9] C. Z. Tang and H. K. Kwang, "Multilayer feedforward neural networks with single powers-of-two weights," *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2724–2727, 1993.

[10] R. Takeda, N. Kanda, and N. Nukaga, "Boundary contraction training for acoustic model based on discrete deep neural networks," in *Proceedings of the Interspeech*, 2014, pp. 1063–1067.

[11] D. Soudry, I. Hubara, and R. Meir, "Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights," in *Advances in Neural Information Processing Systems*, 2014.

[12] M. Kim and P. Smaragdis, "Bitwise neural networks," in *Proceedings of the ICML Workshop on Resource-Efficient Machine Learning*, 2015.

[13] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-vq," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4984–4988.

[14] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Proceedings of Interspeech*, 2013, pp. 2365–2369.

[15] T.N Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6655–6659.

[16] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, "Reshaping deep neural network for fast decoding by node-pruning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 245–249.

[17] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, 2015.

[18] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria.," in *Proceedings of Interspeech*, 2014, pp. 1910–1914.

[19] V. Sindhwani, T. Sainath, and S. Kumar, "Structured transforms for small-footprint deep learning," in *Advances in Neural Information Processing Systems*, 2015.

[20] S. Han, H. Mao, and W.J Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *ICLR*, 2015.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating erros," *Nature*, pp. 533–536, 1986.

[22] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Inproceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[23] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," in *Proceedings of the Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference*, 2009, pp. 131–137.

[24] J. Duchi, Elad. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-03 (11/09)

部分共有アーキテクチャを用いた深層学習ベースの音源同定の検討

Sound Source Identification based on Deep Learning with Partially-Shared Architecture

森戸隆之 *1，杉山治 *2，小島諒介 *1，中臺一博 *1,3
Takayuki MORITO*1, Osamu SUGIYAMA*2, Ryosuke KOJIMA*1, Kazuhiro NAKADAI*1,3
東京工業大学 *1，京都大学 *2，(株) ホンダ・リサーチ・インスティチュート・ジャパン *3
Tokyo Institute of Technology*1, Kyoto University*2, Honda Research Institute Japan Co., Ltd.*3
morito@cyb.mei.titech.ac.jp, sugiyama@kuhp.kyoto-u.ac.jp,
kojima@cyb.mei.titech.ac.jp, nakadai@jp.honda-ri.com

Abstract

災害地における要救助者の捜索を音源同定で実現するために，*Partially Shared Deep Neural Network* (PS-DNN) およびこの拡張版である *Partially Shared Convolutional Neural Network* (PS-CNN) を提案し，これで音源同定器を学習する手法を提案する．通常の深層学習には大量のデータにラベルを付与する作業が必要であるが，提案手法は音源同定器の学習にラベルが付与されていないデータを有効に利用することで，ラベルが付与されたデータのみで学習した場合と比べて高い同定精度が得られることを検証した．

1 序論

地震等の災害現場では，寸断された道路や散乱した瓦礫が要救助者の捜索活動の大きな妨げとなる．クアドロコプタを始めとする *Unmanned Aerial Vehicle* (UAV) で捜索すれば移動の問題は解消されるが，要救助者が瓦礫に埋もれている場合，カメラやレンジファインダ等の視覚的なセンサでの探知は困難である．このため，我々はセンサとしてマイクロホンアレイを用い，災害現場で発生する音の種類と発生位置を同定することで要救助者を探知する方法を研究している．

クアドロコプタにマイクロホンアレイを搭載する場合，風切り音やプロペラが発する雑音によって *Signal-to-Noise* (SN) 比が低下する．このような低 SN 比環境下で音源同定を行う手法として，我々はこれまでに多チャネル音響信号を元にした音源定位手法である *MUltiple SIgnal Classification based on incremental Generalized Singular Value Decomposition* (iGSVD-MUSIC) [Ohata 14] を用いてマイクロホンアレイの収録音から同定対象の音の定位と区間検出を行い，音源分離手法である *Geometric High-order Decorrelation-based Source Separation* (GHDSS) [Nakajima 10] を用いて SN 比の低い多チャネル音から信号成分のモノラル音を分離し，この分離音の種類を *Convolutional Neural Network* (CNN) [Lawrence 97] で識別する手法 [Uemura 15] を提案した．しかし，この手法では音源分離が識別器の最適化とは独立しているため，多チャネル音からの音源分離という大幅な低次元化の過程で識別に有用な情報までもが失われる可能性がある．明示的にノイズ抑圧等の処理を行わず，大規模な DNN を用いて原信号から直接識別する手法 [Hannun 14] も提案されているが，大規模な DNN の学習には大量の学習データが必要である．実環境音を収録して学習データセットを構築する場合，何の音がどの区間で鳴っているのかを示すラベルデータを付与する作業（アノテーション）を人力で行う必要があるり，データ量が多くなれば膨大な工数が発生する．

本稿では，全収録音の一部しかアノテーションされていないデータセットを用いて音源同定器を効率的に学習する手法を提案し，これを実際に DNN，および CNN に適用してその有効性を検証する．一般的な深層学習は学習に教師データ，つまり入力とそれに対応する望ましい出力の組み合わせが必要であるため，アノテーションされていないデータは学習に使用できない．提案手法はラベルデータに加え，信号処理的な音源分離手法で自動生成できる分離音を学習データとして用いることで，音源同定器の学習を効率的に行いつつ未アノテーションデータを有効に利用することができる．

2 部分共有型ニューラルネットワーク

本節では，*Multi-Task Learning* (MTL) [Caruana 97] の一種である *Partially Shared Deep Neural Network* (PS-DNN) およびこれを CNN に拡張した *Partially Shared Convolutional Neural Network* (PS-CNN) の構造につい

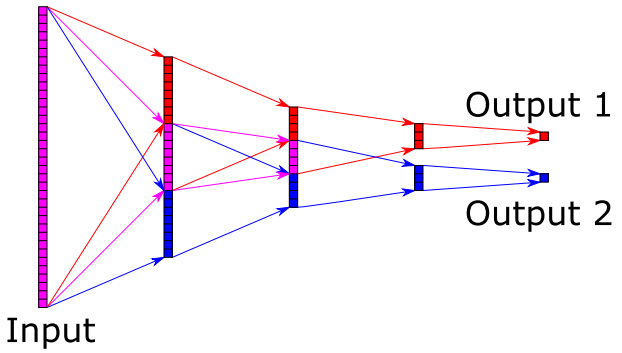


図 1: Partially Shared Deep Neural Network

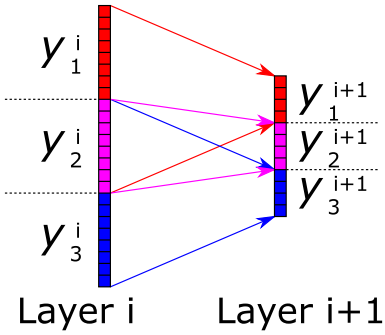


図 2: A hidden layer of PS-DNN

て述べ，多チャネル音響信号を元に音源同定を行う方法を説明する．

2.1 Partially Shared Deep Neural Network

PS-DNN の構造を図 1 に示す．PS-DNN は二つのサブネットワークから成るニューラルネットワークであり，本稿では片方はラベルデータを出力する音源同定器，もう片方は分離音を出力する音源分離器である．サブネットワーク間で入力層と隠れ層の一部が共有されており，共有された隠れ層は二種類の教師データを用いて学習される．

入力層と最初の隠れ層の間は全結合であるが，隠れ層間は全結合ではなく，$\boldsymbol{y}_1^i, \boldsymbol{y}_2^i, \boldsymbol{y}_3^i$ を図 2 における第 i 層の上側，中央の共有部分，下側の隠れ層の出力とすると，第 $i+1$ 層の出力 $\boldsymbol{y}_1^{i+1}, \boldsymbol{y}_2^{i+1}, \boldsymbol{y}_3^{i+1}$ は次の式 (1) で計算される．

$$\begin{pmatrix} \boldsymbol{y}_1^{i+1} \\ \boldsymbol{y}_2^{i+1} \\ \boldsymbol{y}_3^{i+1} \end{pmatrix} = \boldsymbol{\sigma}\left(\begin{pmatrix} \boldsymbol{W}_{11}^i & \boldsymbol{W}_{12}^i & 0 \\ 0 & \boldsymbol{W}_{22}^i & 0 \\ 0 & \boldsymbol{W}_{32}^i & \boldsymbol{W}_{33}^i \end{pmatrix}\begin{pmatrix} \boldsymbol{y}_1^i \\ \boldsymbol{y}_2^i \\ \boldsymbol{y}_3^i \end{pmatrix} + \begin{pmatrix} \boldsymbol{b}_1^i \\ \boldsymbol{b}_2^i \\ \boldsymbol{b}_3^i \end{pmatrix}\right) \quad (1)$$

ここで $\boldsymbol{W}_{jk}^i$ は $\boldsymbol{y}_k^i$ から $\boldsymbol{y}_j^{i+1}$ への重み行列，$\boldsymbol{b}_j^i$ はバイアスベクトル，$\boldsymbol{\sigma}(\cdot)$ は要素ごとの活性化関数である．

共有された隠れ層の出力は上層の全ネットワークに影響を与える．この構造は，音源同定と音源分離はある程度共通の処理で行えるという予想に基づいている．一方，非共有部分の出力はもう一方のサブネットワークの上層には影響を与えず，パラメータは一種類の教師データのみを用いて学習される．これは，音源同定と音源分離にはそれぞれに固有の処理も必要であるという予想に基づいている．この構造により，音源同定と音源分離に共通する処理を二種類の教師データを用いて効果的に学習しつつ，音源分離に固有の処理が音源同定の学習に悪影響を与えることを抑制することが期待される．

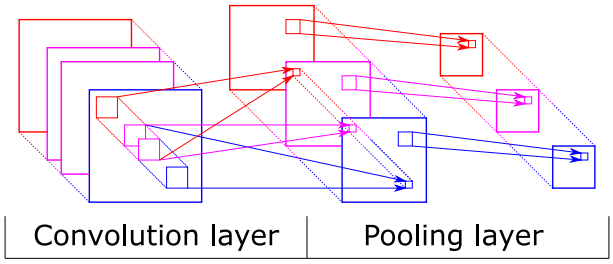


図 3: A convolution-pooling layer in PS-CNN

2.2 Partially Shared Convolutional Neural Network

PS-CNN は，CNN の畳み込み層に PS-DNN の構造を取り入れたものである．なお，本稿では CNN で音響信号を扱う際，1 チャネル分の音響特徴量ベクトルを時間方向に並べた 2 次元の配列を 1 枚の画像とみなす．つまり，一般的なカラー画像認識を行う CNN の入力は画素値を表す 2 次元配列を RGB の 3 チャネル分並べたものであるが，本稿で扱う CNN の入力は上述の 2 次元配列をマイク数分並べたものである．[1]

PS-CNN の畳み込み・プーリング層の構造を Fig. 3 に示す．PS-DNN では各層の出力ベクトルの要素を共有部分と非共有部分に分けたのに対し，PS-CNN ではチャネルを共有チャネルと非共有チャネルに分ける．つまり，一般的な CNN では一つのフィルタは前の層の全てのチャネルを入力とするのに対し，PS-CNN では各サブネットワークに固有のチャネルおよび共有チャネルのみを入力とする．この構造により，PS-DNN と同様に二種類の教師データを有効に利用しつつ，CNN の構造を取り入れることができる．プーリング層は一般的な CNN と同様にチャネルごとにプーリングを行う．出力層の前の全結合層の構造は PS-DNN と同様である．

第 i 層の出力の，一つ目のサブネットワークに固有のチャネルの数を $K_{i,1}$，共有チャネルの数を $K_{i,2}$，二つ目のサブネットワークに固有のチャネル数を $K_{i,3}$ とする．第 i 層の出力を $[\boldsymbol{X}_1^{(i,1)}, \cdots, \boldsymbol{X}_3^{(i,K_{i,3})}]$，$\boldsymbol{X}_1^{(i,1)} = [x_{1,1,1}^{(i,1)}, \cdots, x_{1,V,H}^{(i,1)}]$，第 $i+1$ 層の第 j チャネルの出力のサイズを $V \times H$，出力

[1] CNN で音響信号を扱う別の方法として，音響特徴量ベクトルの次元数をチャネル数とし，音響特徴量ベクトルの要素をマイク数，フレーム数分並べた 2 次元配列を 1 枚の画像とみなす方法も考えられる．この方法の検討は今後の課題とする．

を $\boldsymbol{C}^{(i+1,j)} = [c_{1,1}^{(i+1,j)}, \cdots, c_{v,h}^{(i+1,j)}, \cdots, c_{V,H}^{(i+1,j)}]$ とすると，$c_{v,h}^{(i+1,j)}$ は第 j チャネルが一つ目のサブネットワークに固有のチャネルである場合は式 2 で，共有チャネルである場合は式 3 で，二つ目のサブネットワークに固有のチャネルである場合は式 4 で求められる．

$$c_{v,h}^{(i+1,j)} = \sigma \quad (\sum_{k=1}^{K_{i,1}} \sum_{s=1}^{m} \sum_{t=1}^{n} w_{k,s,t} x_{1,v+s,h+t}^{(i,k)} + \sum_{k=1}^{K_{i,2}} \sum_{s=1}^{m} \sum_{t=1}^{n} w_{k,s,t} x_{2,v+s,h+t}^{(i,k)} + b^{(i,j)}) \tag{2}$$

$$c_{v,h}^{(i+1,j)} = \sigma \quad (\sum_{k=1}^{K_{i,2}} \sum_{s=1}^{m} \sum_{t=1}^{n} w_{k,s,t} x_{2,v+s,h+t}^{(i,k)} + b^{(i,j)}) \tag{3}$$

$$c_{v,h}^{(i+1,j)} = \sigma \quad (\sum_{k=1}^{K_{i,2}} \sum_{s=1}^{m} \sum_{t=1}^{n} w_{k,s,t} x_{2,v+s,h+t}^{(i,k)} + \sum_{k=1}^{K_{i,3}} \sum_{s=1}^{m} \sum_{t=1}^{n} w_{k,s,t} x_{3,v+s,h+t}^{(i,k)} + b^{(i,j)}) \tag{4}$$

ここで m, n はフィルタサイズ，$w_{k,s,t}$ は重み，$b^{(i,j)}$ はバイアス，$\sigma(\cdot)$ は活性化関数である．

3　評価実験

提案手法の有効性を示すため，各手法で音源同定器を構成し，同定精度を比較した．同定精度はフレームごとの正解率とした．各ネットワークは Python のライブラリである TensorFlow version 0.8.0 [Abadi 15] で実装した．

音源同定器は 4 つの手法で構成し，それぞれ DNN, PS-DNN, CNN, PS-CNN と表記する．DNN, CNN はそれぞれ典型的なフルコネクテッド，畳み込みニューラルネットワークで構成した音源同定器で，PS-DNN と PS-CNN は第 2 節で述べた学習手法で構成した音源同定器である．DNN, CNN では学習用データセットの内アノテーション済みのものしか学習に使用しないが，PS-DNN, PS-CNN では未アノテーションデータも音源分離器の学習に使用する．

学習・評価用の音源として，DCASE2016 [Mesaros 16] の Acoustic scene classification に収録されている 15 種類合計 35100 秒分の音データを用いた．5 分割交差検証を行うために合計 1170 個の Wave ファイルを 5 つのグループに分け，3.1-3.2 に示す手順で合計約 465 万個のデータベクトルを生成した．

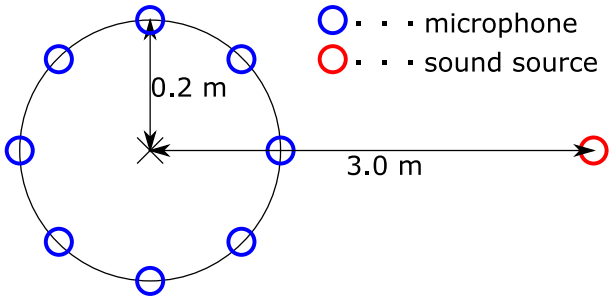


図 4: The layout of the microphones and the sound source

図 5: UAV (Parrot Bebop Drone)

3.1　音響信号の合成

実験に使用した多チャネル音響信号は数値シミュレーションで合成した．まず，コーパスの収録音を元にマイクロホンアレイと音源の位置関係が図に示す通りであるときの 8 チャネルの音を合成した．その後，図 5 に示すクアドロコプタで実際に収録したノイズを適当な重みを付けて足し合わせることで，一定の SN 比の多チャネル音響信号を生成した．なお，本実験で使用したデータセットは場面認識のベンチマーク用のものであるため，収録されている音は既に様々なノイズを含んでいるが，SN 比を計算する際は純信号として扱った．

本稿で使用した SN 比の計算式を式 5 に示す．SN 比は 0 dB に統一した．

$$SNR = 20 \log(S_p / N_p) \tag{5}$$

ここで S_p, N_p はそれぞれ信号成分の最大振幅，雑音成分の最大振幅である．SN 比は信号のエネルギの比率で計算されることもあるが，このような計算方法では有音区間の定め方によって求められる SN 比が大きく変化する場合があるため，本稿では最大振幅で定義した．

表 1: Dimensions for the DNN

Hidden layer	Units
1	2000
2	1000
3	400

表 2: Dimensions for the PS-DNN

Hidden layer	Units		
	Identify	Shared	Separate
1	1500	1500	1500
2	800	400	800
3	400	0	800

3.2 音響特徴量の算出

各学習器への入力としてメルフィルタバンク特徴量を使用した．各音のサンプリングレートは 16 kHz に統一し，フレーム幅 512 sample (32 ms)，フレームシフト 120 sample (7.5 ms) でフレーム化し，窓関数として複素窓を掛けて短時間フーリエ変換で複素スペクトルを求めた．これの絶対値から，下限周波数 63 Hz, 上限周波数 8 kHz, 次元数 20 のメルフィルタバンク特徴量を算出した．以上の処理は，ロボット聴覚ソフトウェア *Honda Research Institute Japan Audition for Robots with Kyoro University* (HARK) [Nakadai 10] で実装した．

各学習器への入力は，20 次元の音響特徴量を 8 チャネル各 20 フレーム分並べた，合計 3200 次元のベクトルである．また，PS-DNN, PS-CNN の音源分離側の出力は，多チャネル音の合成に使用したモノラル音から同様に算出した 400 次元のベクトルである．

3.3 学習器の条件

各学習器の層構成を表 1-4 に示す．全ての場合で入力は 3.2 で述べた 3200 次元のベクトルである．音源同定器の出力層は 15 次元のソフトマックス層であり，PS-DNN, PS-CNN の音源分離側の出力層は 400 次元の全結合層である．各パラメータは 0 に近い正の値で初期化し，pre-training を行わずに Adam で学習した．隠れ層に対しては Dropout を使用し，drop rate は畳み込み層で 0.2，プーリング層で 0，その他の層で 0.4 とした．畳み込み層のフィルタは

表 3: Dimensions for the CNN

Hidden layer	Type	Channels	Size
1	Conv	40	20×20
2	Pool	40	10×10
3	Conv	80	10×10
4	Pool	80	5×5
5	Full	400	1×1

表 4: Dimensions for the PS-CNN

Hidden layer	Type	Channels			Size
		Identify	Shared	Separate	
1	Conv	40	40	40	20×20
2	Pool	40	40	40	10×10
3	Conv	80	40	80	10×10
4	Pool	80	40	80	5×5
5	Full	400	0	800	1×1

表 5: Accuracy of Sound Source Identification

		DNN	PS-DNN	CNN	PS-CNN
100%	Avg.	55.88	**56.27**	55.79	**56.75**
	S.E.	0.6756	0.5742	0.5348	0.5275
75%	Avg.	54.07	**54.57**	54.36	**55.09**
	S.E.	0.6599	0.6584	0.5055	0.3268
50%	Avg.	51.63	**51.91**	51.95	**52.71**
	S.E.	0.5786	0.6525	0.5332	0.5357
25%	Avg.	47.84	**48.04**	48.35	**48.74**
	S.E.	0.6477	0.6526	0.5566	0.5898

全ての場合で 5 × 5 とし，zero padding を使用した．プーリング層では 2 × 2 の範囲で最大値プーリングを行った．全ての場合でバッチサイズは 100 とし，学習は 10 epoch 行った．

3.4 実験結果

実験結果を表 5 および図 6-9 に示す．識別精度はフレームごとの識別正解率とし，5 分割交差検証の平均値 (Avg.) と標本標準誤差 (S.E.) を求めた．5 分割されたデータセットの内 4 つを学習に用い，その 4 つの内の所定の数についてはラベルデータを使用しないことで，アノテーション率が 100%, 75%, 50%, 25%の場合の実験を行った．

表 5 より，識別精度はアノテーション率に依らず DNN < PS-DNN，また CNN < PS-CNN となった．いくつかの場合で両側 t 検定の検定の p 値が $p > 0.05$ で有意な差があった．

識別精度に大きな差が出なかったのは，実験で用いた音

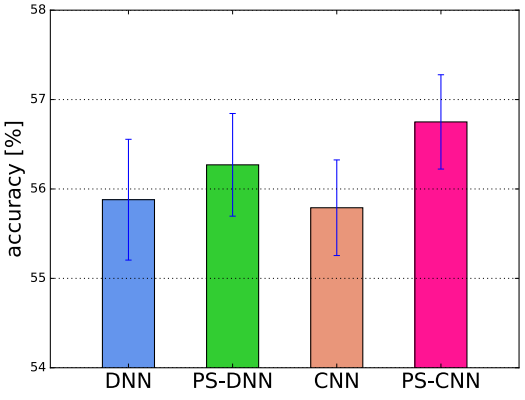


図 6: Trained with 100% annotated data

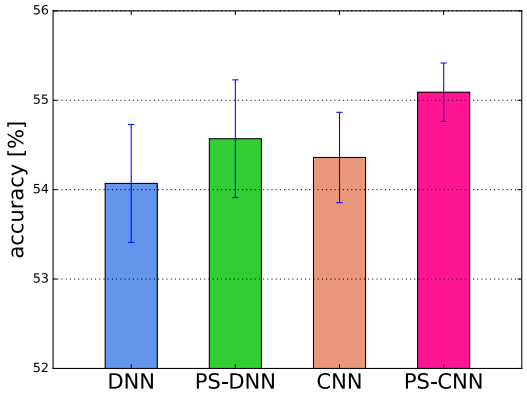


図 7: Trained with 75% annotated data

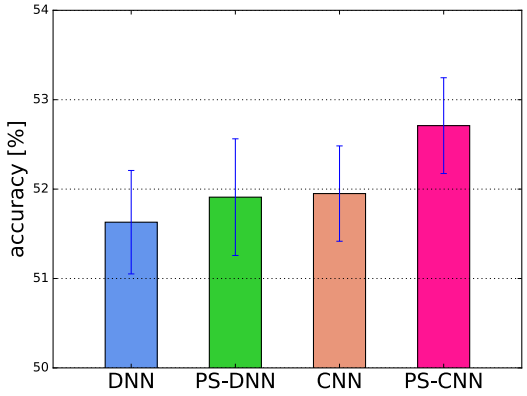


図 8: Trained with 50% annotated data

源分離の教師データに雑音が始めから含まれてしまっていたことが一因であろう．本稿で提案した手法は，低 SN 比環境下で音源同定を行うニューラルネットワークは雑音抑圧の処理を学習しているという推測の下，音源同定器に雑音抑圧の処理を効率的に学習させることを意図している．しかし，本実験で用いたコーパスである DCASE2016 の Acoustic scene classification 用のデータセットは，音を収録した場所（公園，レストラン，電車等）の識別を行うベンチマークデータセットであり，収録されている音は様々な雑音を元々含んでいる．実験ではこの収録音を教師データとして用いたため，学習された音源分離器はクアドロコプタ由来の音以外を除去せず，むしろその他の雑音を積極的に残していたと考えられる．残りの雑音の抑圧は識別器側の共有されていない部分のみを用いて学習することになるため，識別精度が大きく向上しなかったと考えている．

4 結論

本稿では，マイクロホンアレイを搭載したクアドロコプタによる災害地での要救助者の捜索を目的とした，低 SN

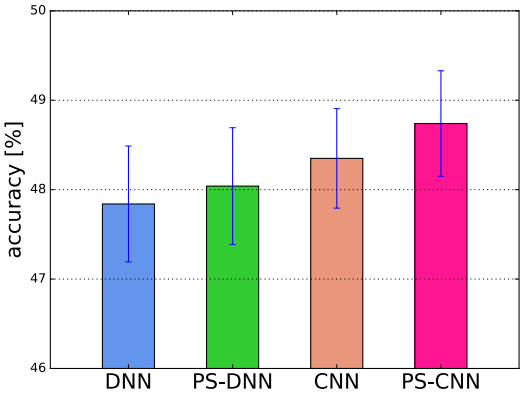


図 9: Trained with 25% annotated data

比環境下での音源同定器の学習手法について述べた．多チャネル音響信号を入力とする音源同定器に，音源分離の処理を積極的に学習させる手法を提案した．提案手法は一般的な DNN, CNN と比べて若干高い同定精度を実現した．今後は，別のデータセットを用いた提案手法の有効性の検証を行う予定である．

謝辞

本研究は JSPS 科研費 24220006, 16H02884, 16K00294 および，JST ImPACT タフロボティクスチャレンジの助成を受けた．

参考文献

[Abadi 15] Abadi, M., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, http://tensorflow.org/ (2015)

[Caruana 97] Caruana, R., et al.: Multitask learning, Machine Learning, vol.28, no. 1, pp. 41-75 (1997)

[Mesaros 16] Mesaros, A., et al.: TUT database for acoustic scene classification and sound event detection, 24th Acoustic Scene Classification Workshop 2016 European Signal Processing Conference (EUSIPCO) (2016)

[Hannun 14] Hannun, A., et al.: Deepspeech: Scaling up end-to-end speech recognition, arXiv preprint arXiv:1412.5567 (2014)

[Lawrence 97] Lawrence, S., et al.: Face recognition: A convolutional neural-network approach, IEEE Transactions on Neural Networks, vol. 8, no. 1, pp. 98-113 (1997)

[Nakadai 10] Nakadai, K., et al.: Design and Implementation of Robot Audition System "HARK", Advanced Robotics, vol. 24, pp. 739-761 (2010)

[Nakajima 10] Nakajima, H., et al.: Correlation matrix estimation by an optimally controlled recursive average method and its application to blind source separation, Acoustical Science and Technology, vol. 31, no. 3, pp. 205212 (2010)

[Ohata 14] Ohata, T., et al.: Inprovement in outdoor sound source detection using a quadrotor-embedded microphone array, IEEE/RSJ International Conference on Intelligent Robots and Systems (2014).

[Uemura 15] Uemura, S., et al.: Outdoor acoustic event identification using sound source separation and deep learning with a quadrotor-embedded microphone array, The 6th International Conference on Advanced Mechatronics (2015)

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-04 (11/09)

ウグイスに対するプレイバック実験における マイクロホンアレイを用いたさえずりの方向分布分析

An analysis on a directional distribution of songs of Japanese Bush-Warbler in playback experiments using a microphone array

炭谷晋司 *1 松林志保 *2 鈴木麗璽 *2

Shinji SUMITANI*1, Shiho MATSUBAYASHI*2, Reiji SUZUKI*2

名古屋大学 情報文化学部 *1, 名古屋大学 大学院情報科学研究科 *2

School of Informatics and Sciences, Nagoya University*1

Graduate School of Information Science, Nagoya University*2

Abstract

本稿では，マイクロホンアレイとロボット聴覚ソフトウェア HARK を用いた野鳥の歌行動観測システム HARKBird の更なる活用可能性の検討として，大学演習林におけるウグイス 1 個体の縄張り内に，同種の歌をスピーカで再生可能にしたシステムを用いてプレイバック実験を行った際の，対象個体のさえずりの種類とその方向を抽出した．また，対象個体のさえずりに対して鳴き返すインタラクティブ実験も行った．実験の結果，本システムは対象個体のさえずりをよく定位できることを確認した．また，再生音の有無が対象個体のさえずりと移動のパターンに大きく影響することなどが明らかになった．

1 はじめに

1.1 背景

野鳥の生態の理解においては，録音機材の進歩や低価格化に伴って録音に基づく行動観測データが容易に得られるようになった．しかし，単一のマイクロホンを用いた録音では，位置情報等を記録できないために個体識別が容易でないなど，個体間で生じる動的な相互作用の分析に問題が生じる場合がある．一方，近年では複数のマイクで構成されるマイクロホンアレイを用いて音源の方向や位置を定位したり，定位した音源を分離したりする技術がロボット工学分野等において発展している．

我々のグループでは，この音声処理技術を野鳥の生態観測に活用することを目的として，ロボット聴覚オープンソースソフトウェアである HARK（Honda Research Institute Japan Audition for Robots with Kyoto University）[1] と，市販の会議などに用いられるマイクロホンアレイを用いた簡易なシステムである HARKBird を構築し，野鳥の歌行動のタイミング，方向，および，音源の自動抽出を試行している [2, 3]．このシステムでは，従来の単一のマイクロホンを用いた録音では得られなかった空間的な情報が取得できるため，生態理解への大きな貢献が期待される．

これまで，我々のグループでは HARKBird システムの定位精度などの基本的な検討を行ってきた．本研究では，HARK の野鳥観測への更なる活用の可能性の検討として，HARK の拡張性と実時間処理を活用した野鳥の歌行動の詳細な方向的・時間的ダイナミクスの計測を試みた．

具体的には，大学演習林におけるウグイス 1 個体の縄張り内に，同種の歌をスピーカで再生可能にした HARKBird システムを用いてプレイバック実験を行った際の，対象個体のさえずりの種類とその方向を抽出した．また，HARK の特徴の一つである実時間処理を活用し，対象個体のさえずりに対して鳴き返すインタラクティブ実験も行った．また，自然環境と人工システムとの相互作用の理解の観点から，ボーカロイド初音ミクによるウグイスの真似歌を用いた実験も行った．これらの結果から，HARKBird を用いた野鳥の歌行動のより詳細な計測と理解への応用の可能性を検討する．

1.2 HARKBird

HARKBird は，フィールドにおいて安価で容易に利用可能なシステムを目指した，野鳥の歌行動観測のためのノート PC と USB マイクロホンアレイを用いた観測・分析システムである（Figure 1）．このシステムは，マイクロホンアレイでの録音の開始，終了をはじめ，HARK の機能を用いた音源定位と分離や，分析結果の可視化がノート PC 上の GUI で実行可能な Python をベースとしたスクリプト集である．詳細は，Web ページ[1]や [2, 3] を参照されたい．

本研究では，HARKBird 内で用いる音源定位・分離の

[1] http://www.alife.cs.is.nagoya-u.ac.jp/~reiji/HARKBird/

ためのネットワークを拡張し，録音中に一定の時間間隔で音声ファイルを再生したり，実時間での音源定位に応じて再生ができるようにした．また，実験後に得られた音声ファイルの分析にも HARKBird を用いた．

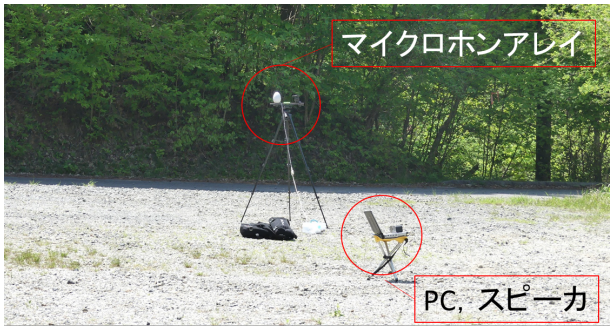


Figure 1: 実験システム

2　手法

2.1　対象種：ウグイス

多くの小鳥（鳴禽類）のオスは，春から夏にかけての繁殖期になると比較的長い音声を繰り返し発するさえずり（歌）を行う．さえずりには，主に他のオスに向けての縄張りの主張と自分の存在をメスにアピールする 2 つの役割があるといわれている．

百瀬は，本研究で対象とするウグイスのオスのさえずりには，ホーホケキョと聞こえる高いピッチで歌う H 型とホーホホホケキョなどとホーの部分が断続する低いピッチで歌う L 型があることなどを，スピーカから同種の鳴き声を再生するプレイバック実験を行って示した [4]．この実験では，スピーカから同種のさえずりの再生を一定期間行い，その間に行われた注目個体のさえずり回数とその種類の計測を行った．結果，スピーカからさえずりを再生している間は再生を行っていない間と比べ H 型の頻度が減少し，L 型の割合が増加することを示唆している．この結果から，L 型のさえずりには近隣個体への威嚇の意味があると考えられることを示した．また，同時に探索飛行といわれる，他個体を探索するために行う飛行回数も測定し，再生音がある場合は探索飛行の回数が増加する結果を示した．

本研究では，プレイバック実験中に音源定位を行い，再生音がウグイスの移動とさえずりの種類に与える影響を調べた．

2.2　実験方法

実験は，2016 年 5 月 21，22 日に名古屋大学大学院生命農学研究科付属フィールド科学教育研究センター稲武フィールド（愛知県豊田市稲武町）の森林で行われた．Figure 2 に示す，周囲を木々に囲まれた開けた場所に三脚に固定した USB 接続マイクロホンアレイ（8 チャネルマイクロホンアレイ TAMAGO（システムインフロンティア社製[2]））を配置し，スピーカ（サンワサプライ社製 MM-SPBTBK）および PC はマイクロホンアレイを原点として真東の方角に 5m 離れた地点に配置した．このとき，スピーカは指向性を考慮して上向きに設置した．この周辺は，ある 1 個体のウグイスの縄張りであり，このウグイスを今回のプレイバック実験の対象個体とした．実験中，対象個体がさえずるソングポストは，マイクロホンアレイ周辺からそれほど大きな差のない距離の木々にあり，方向を計測することで個体の移動パターンをおおよそ把握することができるので，ここを実験場所とした．

Figure 2: 実験フィールド

今回の実験では，再生音源として注目するウグイスについて事前に録音した H 型，L 型のさえずり，昨年の調査で周辺において録音した同種他個体の H 型，L 型のさえずり，ボーカロイド初音ミクを用いてウグイスのさえずりを模した L 型の音を使用した．各再生音源は，正規化することによって最大音量を統一した．

さえずりの再生方法として，数秒毎に再生音を繰り返して周期的に再生する方法と，対象個体のさえずりに対して応答して数秒後に再生する 2 つの方法を採用した．周期的な再生では，8 秒毎，12 秒毎にスピーカから音源を再生し，さえずりに応じて再生する場合には，HARK でリアルタイムに音源を定位した 3 秒後にスピーカからさえずりを再生するように設定を行った．

各実験中に同時に行った録音に対し，ウグイスの鳴き声をよく定位するように HARKBird のパラメータを適宜調整し，音源定位を行った．音源定位結果は，Figure 3 の可視化した音源定位結果が示すように，対象個体の歌行動の時刻と方向をうまく記録することができた．この出力結果を修正し，対象個体のさえずりの時刻，方向，およ

[2]http://www.sifi.co.jp/system/modules/pico/index.php?content_id=39

び，応答実験におけるシステムの再生時刻のデータを作成した．さえずりの種類に関しては，出力結果を元に手作業で集計した．応答実験における注目個体以外の音で反応した場合もスピーカの再生回数として集計した．

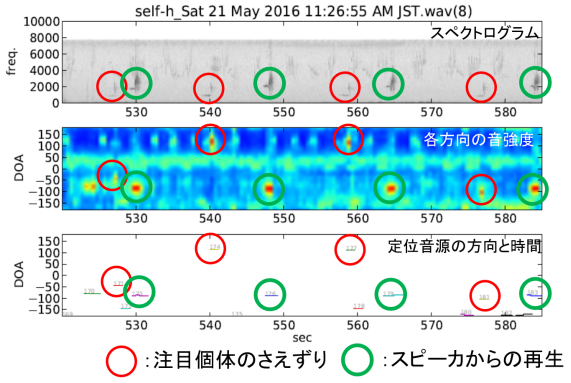


Figure 3: 音源測定結果（注目個体 H 型，応答実験）

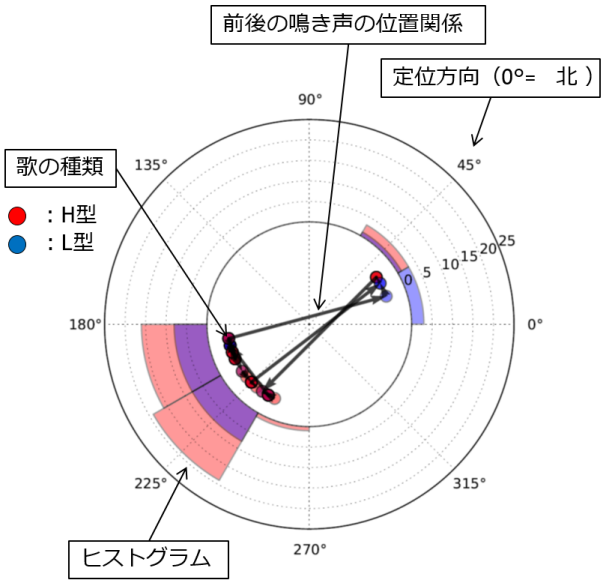


Figure 4: 方向分布（録音のみ）

3 実験結果

3.1 さえずりの種類と方向の分布

各条件での実験 500 秒間における注目個体が行ったさえずりの種類と方向の分布について報告する．ここでの 500 秒間は，注目個体がスピーカ周辺から離れた場所へ移動したなどの理由で長時間鳴いていなかった期間を省いた時間である．

まず，Figure 4 のプレイバックなし，録音のみの結果を例に図を説明する．赤色と青色の丸は H 型あるいは L 型のさえずりをマイクから見てその方向で歌ったことを意味し，方向は 0 度が北（磁北）を指す．さえずり同士を繋ぐ矢印は，定位したさえずりを順に繋いだものであり，さえずり前後での移動を簡易的に示したものである．ヒストグラムは，方向を 30 度ごとに区切った各範囲内での H 型，L 型の各さえずりの回数を示す．

Figure 5 は，各条件における結果を Figure 4 と同様な方法で示したものであり，Figure 1 はその結果に関する定量的な値を示したものである．これらの図と表を元に，全体的な傾向や個別の条件の影響について論ずる．

3.1.1 再生音の有無の影響

まず，プレイバックを行わず，録音のみ行った場合は，南東方向でほとんどのさえずりが定位されているように，ほぼ定位置で頻繁にさえずる傾向があった．さえずりの種類は，L 型のさえずりも用いる（58 回中 20 回）が，H 型のさえずりをより多く歌うこと（58 回中 38 回）が明らかとなった．また，移動に関しては，近隣の木々を少しずつ移動する様子がみられるが，定位方向が大きく変化するような移動はあまり行わない傾向があった．他個体のいない環境においては，定位置に留まって頻繁にさえずりを行う傾向があるようである．

これに対して，再生音のある場合においては，いずれの条件でも再生音のない場合と比較してさえずり回数が減少した．H 型でのさえずり回数の減少がみられる一方で（全再生音平均 11 回，27 回減），L 型のさえずりに関しては増加傾向はあったが，H 型ほど変化はみられなかった（全再生音平均 22 回，2 回増）．また，移動に関しては，定位方向の大きく変わる移動が増加する傾向がみられた．さえずり方向の分散をみても，ほぼ全ての条件において再生音のない対応する条件より高い値を示しており (録音のみ 0.464 全再生音平均 0.876，0.412 増)，個体は同じ場所に留まらず，あちこち動き回るような傾向があった．これらのことから，ウグイスは H 型のさえずりを減らし，L 型の比を高め，周囲を飛び回ることで強い警戒を示していることが考えられる．

3.1.2 プレイバック間隔・応答の影響

次に，プレイバックの間隔の影響について考える．プレイバック間隔の違いは，さえずりの頻度や動きに影響を与えることが明らかとなった．まず，8 秒間隔と 12 秒間隔それぞれの H 型のさえずり回数を比較すると，8 秒の場合平均 9 回，12 秒の場合平均 19 回であり，全ての再生音において H 型のさえずり回数が 8 秒間隔で再生を行った際に少なくなる傾向がみられた．また，移動に関しても，8 秒間隔では分散は平均 0.916 であるに対して，12 秒間隔では，平均 0.798 であるように，大きな移動の頻度が 8 秒間隔において高い傾向にあった．一方，L 型に関しては，さえずり頻度が 8 秒の場合に大きくなるものの影響は小さかった（12 秒平均 22 回 8 秒平均 23 回）．これより，頻繁な再生は注目個体に対して強い警戒を生じさせ，

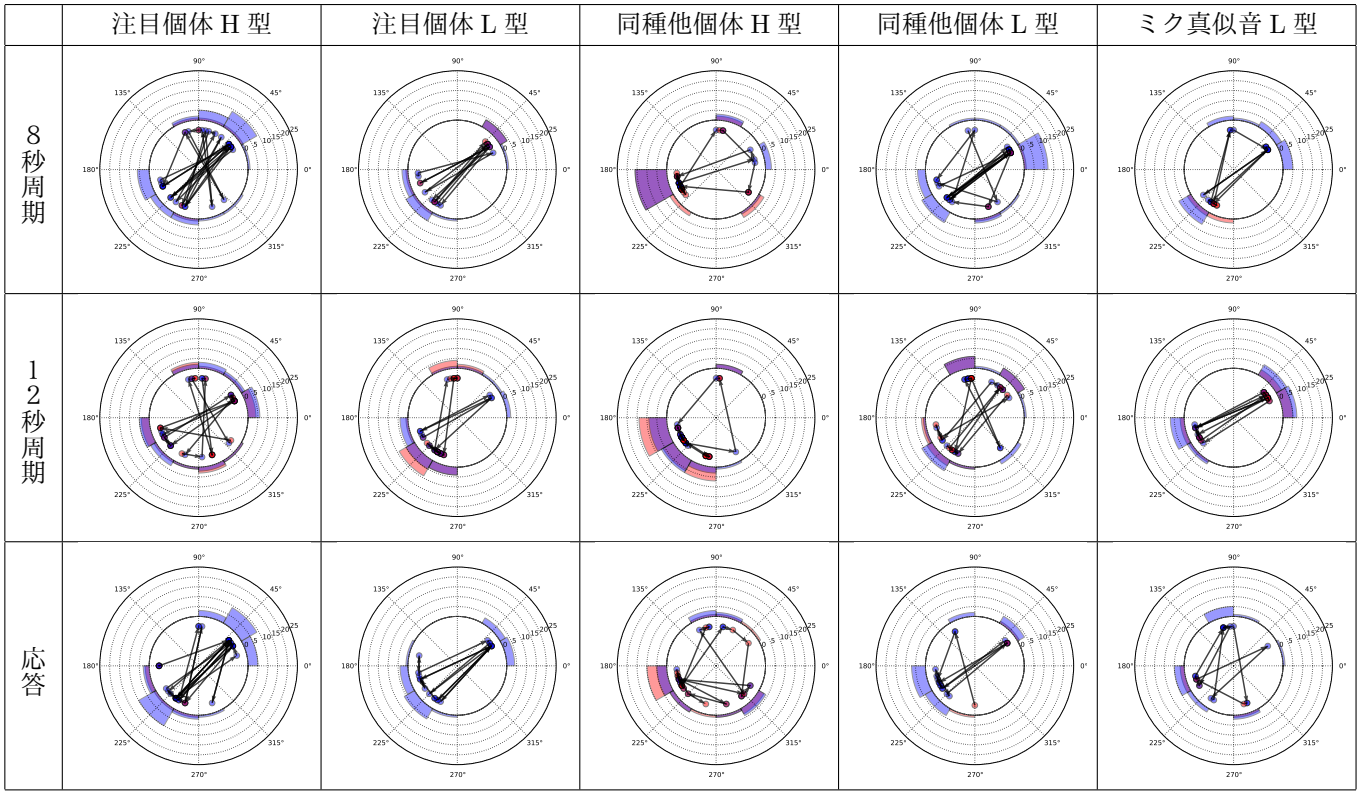


Figure 5: さえずりの種類と方向の分布と推移

Table 1: 各条件におけるさえずり回数と方向分散，スピーカからの再生数

再生音			対象個体のさえずり回数 H	L	H＋L	H：L		さえずり方向の分散	スピーカからの再生回数
録音のみ（プレイバックなし）			38	20	58	0.655	0.345	0.464	0
注目個体	H	8秒	4	31	35	0.114	0.886	0.960	60
		12秒	19	25	44	0.432	0.568	1.000	40
		応答	4	33	37	0.108	0.892	0.998	28
	L	8秒	6	14	20	0.300	0.700	0.995	60
		12秒	20	18	38	0.526	0.474	0.780	40
		応答	0	19	19	0.000	1.000	0.958	25
同種他個体	H	8秒	27	24	51	0.529	0.471	0.674	59
		12秒	29	22	51	0.569	0.431	0.370	40
		応答	20	17	37	0.541	0.459	0.854	29
	L	8秒	2	28	30	0.067	0.933	0.959	60
		12秒	16	22	38	0.421	0.579	0.941	40
		応答	2	17	19	0.105	0.895	0.885	21
初音ミク	L	8秒	4	18	22	0.182	0.818	0.992	59
		12秒	11	21	32	0.344	0.656	0.900	40
		応答	3	16	19	0.158	0.842	0.869	30
平均		8秒	9	23	32	0.238	0.762	0.916	60
		12秒	19	22	41	0.458	0.542	0.798	40
		応答	6	20	26	0.182	0.818	0.913	27
		全再生実験	11	22	33	0.293	0.707	0.876	42

H 型のさえずり頻度を減らし，移動頻度を増加させることが示唆される．

これらに対し，対象個体のさえずりの 3 秒後に再生する応答実験では，対象個体のさえずり回数はさらに少なくなることがわかった（H 型平均 6 回，L 型平均 20 回，合計 26 回）．特に H 型のさえずりは 8 秒周期と同程度、あるいはそれを超えるほどの抑制がみられた．このとき，全体としてスピーカによるシステムからの応答回数は平均 27 回であり，他の条件（8 秒周期平均 60 回，12 秒周期平均 40 回）と比べて少ない傾向があった．

以上から，周期的なプレイバック実験の結果ではスピーカからの再生回数が多いほど注目個体の H 型でのさえずり回数は抑制されていたが，応答実験ではその傾向とは異なり，少ない頻度でも大きな抑制の効果があったといえる．これは，注目個体が応答によるスピーカからの再生音を，単純なプレイバックと比べてより自分自身に向けられたもとのとして意識し，警戒を強めた可能性を示唆している．8 秒間隔の実験でのスピーカからの再生回数は，応答実験での再生回数の倍近くあることと，また 8 秒間隔と 12 秒間隔にある差を考慮すれば，注目個体のさえずりに与える影響の度合いとしては，再生間隔よりも，再生のタイミングによる影響が強いことが示唆される．観測とプレイバック実験において個体の鳴き声による反応が異なることは知られているが [5]，この結果は，単純なプレイバック実験とインタラクティブ実験においても異なる反応が得られうることを示唆している．

3.1.3　再生音の種類による影響

再生音の種類によっても，さえずりや動きに変化がみられた．再生音のある場合，定位方向が大きく変化する移動が増加したと述べたが，その移動は大まかに，スピーカ上近くを通過するように 2 つのソングポスト間を繰り返し行き来する移動と，スピーカ周りの移動を繰り返し，周りからスピーカの様子をうかがう移動の 2 種類が確認できた．

前者の移動は，注目個体の H 型，L 型，同種他個体の L 型，ミクの真似音 L 型を再生した場合に多くみられた．この移動は，Figure 5 に示す方向グラフの中央付近（270 度方向）にスピーカがあり，その周辺を矢印が横切っていることから確認できる．特に，注目個体の H 型を再生した場合，分散も高く（0.960-1.000），図からも様々な場所へ移動してあらゆる方向でさえずりを行っていることが確認できた．また，他の実験に比べてスピーカ上近くを通過する回数が多いことから (約 10 回程，他の実験では 5-7 回程度)，注目個体は注目個体 H 型の再生音に対して，より強い警戒を示したことが示唆される．

また，後者の移動は，同種他個体の H 型，L 型を再生した場合に多く確認できた．特に，同種他個体 L 型を再生した場合に多くみられ，またスピーカ上近くを通過するような移動がほとんどみらなかったり，他の再生音を用いた実験に比べて H 型のさえずり回数の減少が少なく，さえずり方向の分散も低い値を示すなど，異なった傾向がみられた．これは，自分（注目個体）の縄張り内に他個体が侵入し，縄張りの主張として H 型をさえずるという状況に，対象個体が呼応してさえずりの頻度を高めたのではないかと推測される．

さらに，初音ミクの真似音 L 型に関して，先ほど述べたように本物のウグイスの録音を用いた他の再生実験と同様な傾向がみられていることから，注目個体は初音ミクの真似音に関しても同種の個体であると認識していることが考えられる．

以上のように，注目個体は，再生音の違いによって異なった挙動をみせた．人間には判断の難しいわずかなさえずりの違いでも，ウグイスは認識し，その状況に応じて歌と移動のパターンを変化させているということが示唆される．

3.2　さえずりの種類と移動の関係

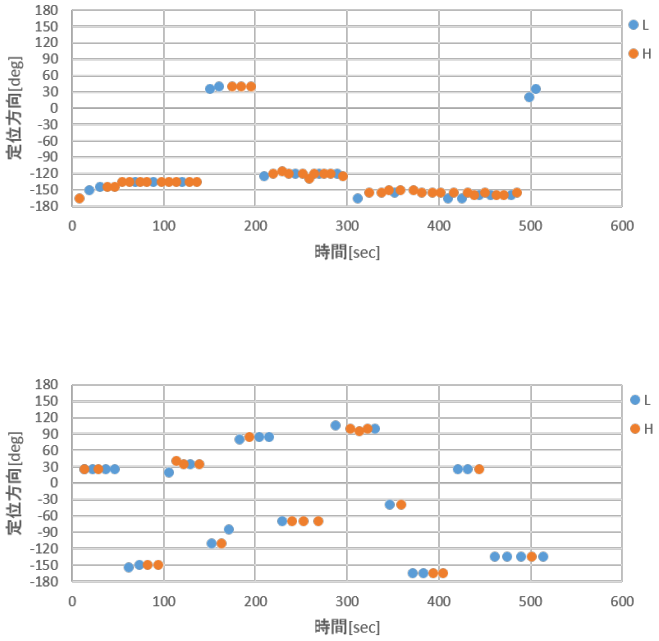


Figure 6: さえずりの方向と種類の推移．上：録音のみ，下：注目個体 H 型の録音を 12 秒周期で再生した場合．

次に，注目個体の移動と歌の種類の関係について，分析で得られた結果を報告する．Figure 6 は，録音のみと注目個体 H 型の録音を 12 秒周期で再生した場合のさえずりの方向と種類の推移を示したものである．このグラフから，注目個体は定位位置が大きく変化する移動を行った後，最初にさえずる歌の種類として L 型を用いる確率が非常に高いことが確認できる．Table 2 は，各条件における，定位位置の変化回数 と，その変化後の最初のさえず

Table 2: さえずりの種類と移動の関係

再生音			30度以上の定位位置変化回数	その後のさえずり	
				H型	L型
録音のみ（プレイバックなし）			5	0	5
注目個体	H	8秒	19	2	17
		12秒	11	1	10
		応答	17	0	17
	L	8秒	10	1	9
		12秒	6	0	6
		応答	12	0	12
同種他個体	H	8秒	6	1	5
		12秒	5	0	5
		応答	8	0	8
	L	8秒	15	0	15
		12秒	8	0	8
		応答	7	0	7
初音ミク	L	8秒	7	0	7
		12秒	7	0	7
		応答	7	1	6

りの種類を示した表である．定位位置の変化回数は，さえずり前後で 30 度以上の定位角度の変化があるときを移動と考え，その回数を集計した．表からわかるように，すべての条件において，定位位置の大きく変化するような移動を行った後の最初のさえずりは，L 型を用いる傾向があった．再生音の種類，有無に関わらず，この傾向がみられることから，注目個体の行動として移動後は L 型のさえずりを行う傾向があることが示唆される．

例えば，現在地点から遠方で移動し，最初に H 型でさえずりを行った場合，移動先に他個体がいたとすれば，他個体はすぐさま攻撃してくる可能性がある．しかし，L 型を最初にさえずることにより，移動先に他個体がいたとしても，他個体は自分の存在が気付かれていると認識し，攻撃を行ってこないかもしれない．そういったリスクを排除するために，移動直後に L 型のさえずりを行っていることは考えうる．いずれにしても，ウグイスにとって何らかの意味ある行動であることが強く示唆される．

4　議論

我々の実験で得られた結果において，周期的なプレイバックとさえずり頻度の関係については，先行研究 [5] の結果とよく一致した．従来のモノラルマイクによる実験方法では，多くの場合，歌の種類やタイミングは録音から手作業で抽出することが多い．また，探索行動などの個体の動きを観測する場合も，対象とする野鳥は小さく素早く動くため，その場で目視で確認するのが一般的であり，どちらの意味でも時間や作業のコストが掛かる．一方，我々の行ったマイクロホンアレイを用いた録音での観測は，どちらに関しても，容易に様々なデータを得ることができるだけでなく，より正確なデータを得られることが期待できる．

その結果，今回の実験では，従来研究と同様な結果に加えて，スピーカ上を通過する，スピーカ周りを移動するなどのように，探索飛行の詳しい挙動も確認できたり，更には移動後におけるさえずりの特徴など，これまで確認されていなかった挙動もみることができた．この結果は，移動情報を伴う録音データを分析することによって得られたものであり，実験と同時に調査を行うことなく分析が可能な点もマイクロホンアレイを用いた録音のメリットでもある．

また，応答実験では，周期的な再生による実験と異なる結果が得られた．これは，ウグイスが他個体のさえずりと相互作用しうることを意味しており，実時間での処理によるインタラクティブ実験の有用性を示唆している．今回の実験では，単に音源定位の 3 秒後というタイミングで再生音を流したが，様々に拡張しうる．例えば，注目個体のさえずりの間隔に合わせて再生したり，再生を行う，行わない，あるいは再生音にバリエーションを持たせるなどの変化を加えることで，従来研究では確認できなかった新たな挙動も確認できる可能性がある．

一方で，本実験の課題もいくつか存在した．今回の実験では，ノイズや近隣個体のさえずりによる応答実験でのプレイバックの誤動作などの不具合も確認された．音源の定位は，環境に大きく依存するため，場所ごとにパラメータを変更する必要性がある．応答実験では，そういった問題も考慮しつつ，注目した個体のさえずりにのみ反応するといった方法も考えていかなければならない．

再生音の選定に関しても課題がある．例えば，注目個体のさえずりは，注目個体自身がどのように捉えているか、同種他個体のさえずりは同じフィールドで得た録音であるが，近隣個体として捉えているのか，あるいは全く知らない個体として捉えているかなど，不明な点があった．この問題は，他の再生音を用いて実験を行うことで解決できると考えられる．

さらに，今回の実験では 1 個体のみを実験対象としたため，実験結果に個体特有の特徴が現れている可能性が示唆されること，実験の順番や実施時間，注目個体の再生音に対する慣れの影響など，考慮すべきことがいくつかあった．今回の結果を踏まえ，より信頼できるデータが得られる実験を行うことも今度の課題となる．

5　おわりに

本稿では，マイクロホンアレイを用いたロボット聴覚ソフトウェア HARK に基づく野鳥歌行動解析システム HARK-Bird を拡張し，ウグイスに対して行ったプレイバック実験におけるさえずりの時間的・方向的ダイナミクスの分析結果を報告した．本システムは対象個体のさえずりをよく定位した．録音のみ行った場合，定位置で縄張りを主張する H 型のさえずりを多く歌う一方，プレイバックを行った場合，H 型のさえずりの頻度が減少して敵を警戒する L 型のさえずりの割合が増加したり，ソングポスト間をより頻繁に移動するなど，注目個体はスピーカからの再生音を警戒する傾向がみられた．その傾向は，再生音によって異なることも確認された．また，ソングポストの

移動直後はL型の歌を歌う傾向があるという，歌行動と移動の複合的なパターンを抽出することができた．以上の分析はHARKBirdを用いることによって容易に可能であり，野鳥の生態観測への活用の可能性が示された．

謝辞

高部直紀氏（名古屋大学）の調査協力，クリプトン・フューチャー・メディア株式会社関係各氏のボーカロイド音声作成協力に謝意を表する．本研究の一部はJSPS科研費15K00335，16K00294，24220006の助成を受けたものである．

参考文献

[1] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa and H. Tsujino: Design and implementation of robot audition system 'HARK' - Opensource software for listening to three simultaneous speakers, *Advanced Robotics*, 24. 739-761 (2010).

[2] R. Suzuki, S. Matsubayashi, R. Hedley, K. Nakadai and H. G. Okuno: HARKBird: Exploring acoustic interactions in bird communities using a microphone array, *Journal of Robotics and Mechatronics* (accepted).

[3] R. Suzuki, S. Matsubayashi, K. Nakadai and H. G. Okuno: Localizing bird songs using an open source robot audition system with a microphone array, *Proc. of 2016 International Conference on Spoken Language Processing (Interspeech 2016)*, 2626-2630 (2016).

[4] 百瀬博 : 音声コミュニケーションによるなわばりの維持機能. 山岸哲（編）. 鳥類の繁殖戦略（下）,127-157, 東海大学出版会, 東京 (1986).

[5] D. F. Maynard, K. A. A. Ward, S. M. Doucet and D. J. Mennill: Calling in an acoustically competitive environment: duetting male long-tailed manakins avoid overlapping neighbours but not playback-simulated rivals, *Animal Behavior*, 84(3): 563-573 (2012).

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-05 (11/09)

空間情報を用いた鳥の歌分析

小島 諒介 [1], 杉山　治 [2], 干場　功太郎 [1], 鈴木　麗璽 [2], 中臺 一博 [1,3]

Ryosuke KOJIMA[1], Osamu SUGIYAMA[2], Kotaro HOSHIBA[1], Reiji SUZUKI[3], Kazuhiro NAKADAI[1,4]

1. 東京工業大学, 2. 京都大学, 3. 名古屋大学,
4. （株）ホンダ・リサーチ・インスティチュート・ジャパン

1. Graduate School of Information Science and Engineering, Tokyo Institute of Technology,
2. Graduate School of Information Science, Nagoya University,
3. Honda Research Institute Japan Co., Ltd.

kojima@cyb.mei.titech.ac.jp, sugiyama@kuhp.kyoto-u.ac.jp,
hoshiba@cyb.mei.titech.ac.jp, reiji@nagoya-u.jp, nakadai@jp.honda-ri.com

Abstract

本稿では鳥の歌の自動分析システムの構築を目的とし，そのための空間情報を考慮した音源同定モデル提案する．鳥の歌を分析し，いつどこで何の鳥が鳴いているかを自動的に発見するシステムは野鳥研究や動物行動学研究において，観測の効率化･大規模化といった点から期待されている．しかし，このようなシステムは，異なる地点で同時に複数の野鳥が鳴いているといった状況に対応する必要があるといった困難がある．我々は，音源検出･定位･分離･同定といった，ロボット聴覚の技術を利用することで，鳥の歌分析フレームワークの構築を目指している．本稿では，位置情報に注目することで，これらの統合を可能にする確率モデル　Spatial-Cue-Based Probabilistic Model (SCBPM) を提案する．さらに，提案モデルを用いたアノテーション補助システムを構築し，実データを用いた評価実験を行った．その結果，鳥の歌識別のタスクにおいて，従来法より最大で 5%の識別率向上が確認できた．また，提案モデルが有効でない状況においても，従来法と変わらない性能が達成できることが確認できた．

1　はじめに

音環境理解は，環境中の音に注目することで，画像などの情報からでは得ることが難しい情報や障害物が多くオクルージョンが問題となるような環境で大きな手助けとなることから，環境モニタリングやロボットなどの分野で，注目されている．音環境理解における主な困難の一つは，環境中にある混合音の中から有用な情報を抽出することである．我々は音環境から 5W1H 情報 (When, Where, What, Who, Why, How) を抽出することを音環境理解と定義し，研究を行っている．これら 5W1H のうち，はじめの 4W は音環境中でのイベントに関する重要な情報であるため，特に，音源検出（When），音源定位（Where），音源分離（What），話者認識・音源同定（Who）として取り組まれている．これまでの 4W 情報抽出はそれぞれの情報に関して個別に抽出するものであり， 4W すべての情報を抽出する単純な手法はこれらをカスケード的に実行することである．つまり，収録した音から音源を検出･定位し，複数音源がある場合にはそれぞれ分離し，同定を行う手法である．このアプローチには二つの欠点がある．一つは，処理が多段になるため，誤差が蓄積してしまうことである．もう一つは，音源の位置と種類の関係性など 4W 間の相互依存を考慮していない点である．こういった課題を扱うため，音源定位と音源分離を同時に行う手法として BNP-MAP 法 [Otsuka 14] が提案されている．また，環境理解以外でも相互に依存した情報を扱うモデルは研究されており，例えば，関係データを扱う Stochastic Block Model （SBM） はネットワーク分析や関係クラスタリングで利用されている [Holland 83]．関係クラスタリングは，二つ以上の対象の関係を用いてクラスタリングする手法であり，同時に複数の音が存在する場合の相互依存関係を扱う問題とも関係が深い．

また，定位と同定の相互依存性，特に，いつどこで誰が話しているかという問題は話者ダイアライゼーションとして広く研究されている．例えば，マイクロホンアレイを用いて到達時間差を特徴量として補助的に利用することで話者のセグメンテーション精度が向上できることが知られている [Pardo 06]．これらの話者ダイアライゼーションの手法は会議室などの室内で，人の声を対象にしている．

本稿では，定位と同定の相互依存性を取り扱うために，鳥の歌分析を対象にした．鳥の歌の音源同定は音源の位置情報と深く結びついたタスクの一つである．なぜならば，野鳥は自分の縄張りを持っており，近くの個体は同一の個体であるといったこれらの情報を考慮しつつ，障害物の

多い森林などの環境において，音源同定を行うタスクは環境モニタリングとしても挑戦的なタスクの一つである．

また，鳥研究においても，鳥の歌は縄張りの主張や求愛などの役割を持つことから興味深い対象の一つである [Catchpole 03]．鳥の歌研究には二つの方法があり，一つはよくコントロールされた環境での観測で，もう一つは実際のフィールドにおける観測である．前者は条件をコントロールした鳥の歌の性質の研究が可能であり，不要な条件を排除して実験を行うことができる．本稿では後者に主眼をおいており，この方法では，実際の環境下で鳥がどのように活動しているかを直接観測することができるため，注目されている方法である．

鳥の歌の同定は機械学習分野でも注目されており，いくつかのコンペティションで取り上げられており，様々な手法が提案されている [Briggs 13, Goëau 16]．これらのコンペティションで用いられているデータは 1 もしくは 2 チャンネルのマイクロフォンで収録された音である．3 つ以上のマイクロフォンで構成されたマイクロフォンアレイを用いた収録は，定位や分離の性能向上のために鳥の歌に利用できることが報告されている [Suzuki 16]．音源同定についても，定位や分離と深く結びついているため，マイクロフォンアレイを用いることが有効であると期待できるが，その性能評価については十分されていない．

先行研究　[小島 15]　として，音源位置を考慮した同定モデルが提案されている．この手法は，音源同定を行うための音響モデルを分離音ごとに独立なモデルとして学習した後，音源位置を考慮した関数を用いてそれらのモデルを結合するというアプローチを取っていた．しかし，このアプローチは，分離音ごとに独立に学習を行うため，音源位置を音響モデルに反映できなかった．これを解決するために，EM アルゴリズムによる相互依存の学習を可能にしたモデルも提案されている [小島 16]．しかし，この手法では一部のパラメータに関するパラメータ学習を行っておらず，また，[Kojima 16] では実験も一環境のみの限定的なものであった．本稿では，これらのパラメータについての学習を可能にし，異なる環境での実験を行い，モデルを評価した．

2　課題とアプローチ

我々は，マイクロフォンアレイを用いて収録した鳥の歌を自動的に分析し，鳥の研究を補助するシステムの構築を目指す．本稿では音源検出・定位・分離・同定をターゲットにし，音源同定においてこれらを統合するモデル Spatial-Cue-Based Probabilistic Model (SCBPM) を提案する．

モデルの構築にあり以下の 3 つの課題が挙げられる．

1. 音源定位情報を用いた音源同定のモデル化
2. 音源定位情報を考慮したモデルパラメータの学習法
3. 部分的なアノテーションを考慮した学習法

一つ目の課題は，野鳥観測を行うような森林や屋外では，木々などの障害物や地形による影響により，音源分離が十分な性能を発揮できないことに由来している．実際に分離音に他音源からの音（同時に鳴いている他の鳥の歌など）が漏れてしまい，識別がうまくいかない場合が多く存在した．一方で，MUSIC 法 [Schmidt 86] やその拡張手法は，屋外の騒音環境や野鳥が住処にするような森林であっても，ある程度の音源の到来方向を推定できることが報告されている [松林 15, Uemura 15]．そこで，分離音からの情報を補助するために，到来方向を含めた確率モデルを提案する．提案モデルは，分離音に関する確率分布と到来方向に関する確率分布の 2 つの分布から構成される．分離音に関する分布としては，音源同定のモデルとしてよく知られている 混合ガウス分布 (GMM: Gaussian Mixture Model) を用い，到来方向に関する分布には方向統計学でよく用いられる von Mises 分布を用いる．

二つ目は，構築したモデルのパラメータの学習をいかに行うかという課題である．提案モデルでは，同時刻の分離音は定位情報を通して相互に依存している．提案モデルは，隠れ変数を持つ確率モデルであるため，Expectation Maximization (EM) アルゴリズムで学習することが考えられるが，分離音間の相互依存性を考慮して学習する必要がある．そこで，本稿では新たに提案モデル上での EM アルゴリズムを導出し，この課題の解決を図る．

三つ目にデータのアノテーションに関する課題である．提案モデルの学習のためにはアノテーションが必要であるが，すべてのデータに対し人手でこれを行うは労力がかかる．そこで，一部のデータに人手でアノテーションをし，残りを推定するのが妥当である．すると，アノテーションの推定は大量のアノテーションされていないデータと少数のアノテーションデータから識別器を構成する問題となり，これは半教師あり学習問題として知られている．特に，上述の EM アルゴリズムでパラメータ学習をする場合には，アノテーションされていないデータを欠損値とみなすことにより，自然に半教師あり学習へと拡張することができる [Nigam 00]．

3　カスケード法

ここでは，収録した音から音源を検出・定位・分離・同定のための単純なカスケード手法を説明する．図 1 は，マイクロフォンアレイで収録した音を検出・定位・分離・同定の順に実行して分析するカスケード手法を示している．この手法ではそれぞりの処理は独立して行われる．この節の以降ではこれらの処理についての詳細を述べる．

音源検出・定位では，音源数，音源の位置，音源がアク

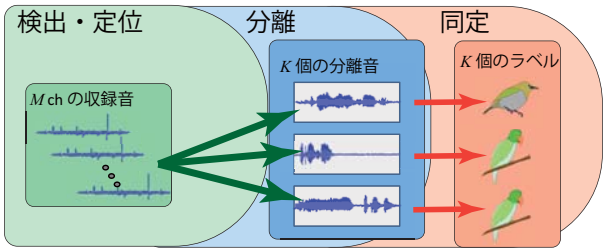


図 1: 音源検出・定位・分離・同定のカスケード法

ティブになっている区間の推定を行う．これらは MUSIC 法を用いて達成される．MUSIC 法ではそれぞれの方向 θ ごとに MUSIC パワー $P(\theta)$ と呼ばれる信号部分空間のパワーを計算する．

$$P(\theta) = \frac{||\mathbf{a}^H(\theta)\mathbf{a}(\theta)||}{\sum_{i=L+1}^{M} ||\mathbf{a}^H(\theta)\mathbf{e}_i||}$$

ただし，$\mathbf{a}(\theta)$ は予め計測しておくステアリングベクトル，$\mathbf{e}_i$ はマイクロフォンアレイのチャネル間の相関行列を固有値展開し，固有値の大きいもの順に並べた固有ベクトルとそのインデックス i である．L は MUSIC 法のパラメータであり，音源候補の数，M はマイクロフォンの数である．MUSIC 法では信号部分空間が雑音部分空間と直交することを利用しており，音源のある方向で $P(\theta)$ は高い値となる．この時，スレッショルドを超えるもののみを検出し，そのピーク値を見つけることにより，音源の方向を推定する．また，各時間フレームごとに方向を推定し，フレーム間での方向の差がある閾値以下であれば同一の音源とみなし，時間的に音源を追跡することで，音源がアクティブになっている区間を推定することができる．これらのスレッショルドの値も MUSIC 法のパラメータである．

音源分離では混合音から目的の音源の音を抽出する．音源分離の手法としてはビームフォーミング法などがよく知られている．GHDSS(geometric highorder decorrelation-based source separation) 法は分離音間の高次無相関性を考慮してビームフォーミングを拡張した手法であり，方向性ノイズに強い．我々は，屋外の様々な方向性ノイズから野鳥の歌を分離するため，GHDSS 法を用いた．

音源同定では GMM を用いた音響モデルを利用する．このモデルでは，一つの音源クラスは複数のサブクラスを持ち，各時刻でのある音源からの音は，それらの中から確率的に選択されると仮定する．より具体的には，周波数スペクトルから計算した音響特徴量が多変量ガウス分布に従うとし，一つの音源クラスであってもサブクラスの数だけ周波数スペクトルのパターンを表現できる．本研究では，周波数スペクトルを時間方向に主成分分析（PCA）により次元圧縮をしたベクトルを音響特徴量とした．

このようにモデル化すると，入力の音響特徴量 $\mathbf{x}$ は以

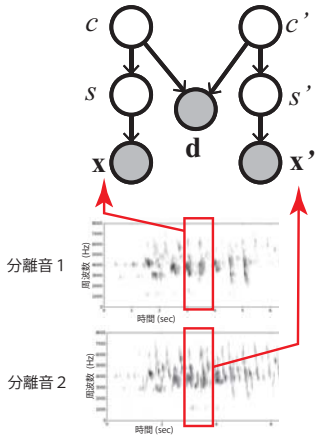


図 2: 提案モデルのベイジアンネットワーク表現（同時刻の音源が二つの場合）:観測変数 $\mathbf{x}$，$\mathbf{x}'$ は同時刻の別の分離音から計算される音響特徴量であり，観測変数 $\mathbf{d}$ はそれらの到来方向のベクトル．

下の混合ガウス分布に従うこととなる．

$$p(\mathbf{x}, s_{cj}, c) = \mathcal{N}_{cj}(\mathbf{x})p(s_{cj} \mid C = c)p(C = c)$$

ただし，s_{cj} は音源 c の j 番目のサブクラス ($\sum_j p(s_{cj} \mid C = c) = 1$)，$\mathcal{N}(\cdot)$ は多変量ガウス分布である．ここで，音源の種類 C を確率変数とおき，アノテーション済みデータの場合には固定値とすることで，EM アルゴリズムにより半教師あり学習を行う．また，モデルの構築後は MAP(Maximum A Postriori) 推定により，音源の同定を行うことができる．

4 提案法

前節の音源同定のモデルでは音源位置に関することは考慮されていなかった．ここでは音源位置を利用できるようにモデルを拡張し（4.1 節），そのモデルのパラメータ学習について述べる（4.2 節）．

4.1 提案モデル

前述の GMM による音響モデルでは分離音ごと独立にモデル化していた．したがって，時刻 t，分離音 k_t ごとに独立であった．提案モデルは各分離音間の依存性を導入し，これを拡張したものになっている．提案モデルのベイジアンネットワーク表現を図 2 に示す．ここで，時刻 t の音源 k_t の方向 $\mathbf{d}_t = d_{t,1}, d_{t,2}, ..., d_{t,k_t}, ..., d_{t,K_t}$ $(0 \leq d_{t,k_t} < 2\pi, 1 \leq k_t \leq K_t)$ は MUSIC 法を用いて計算できる．この時，3 節で述べたように時刻 t の音源数 K_t も決定する．また，各分離音の音響特徴量 $\mathbf{x}_{k_t}$ は GHDSS 法を用いて計算できる．そのため，図 2 ではこれらを観測として表現している．以下では，時刻 t は簡単のために

具体的には，以下のように記述される．

$$p(\mathbf{x},\mathbf{d},\mathbf{s},\mathbf{c}) = p(\mathbf{d}\mid\mathbf{c})\prod_{k=1}^{K}\mathcal{N}_{s_{c_k}}(\mathbf{x}_k)p(s_{c_k}\mid c_k)p(c_k) \tag{1}$$

$$p(\mathbf{d}\mid\mathbf{c}) = \prod_{c_i=c_j,i\neq j} p(d_i,d_j\mid c_i=c_j) \prod_{c_i\neq c_j,i\neq j} p(d_i,d_j\mid c_i\neq c_j) \tag{2}$$

$$p(d_i,d_j\mid c_i=c_j) = f(d_i-d_j;\kappa_1) \tag{3}$$

$$p(d_i,d_j\mid c_i\neq c_j) = f(d_i-d_j+\pi;\kappa_2) \tag{4}$$

$$f(d;\kappa) = \frac{\exp(\kappa\cos(d))}{2\pi I_0(\kappa)} \tag{5}$$

ただし， $f(d;\kappa)$ は von Mises 分布であり，$I_0(\kappa)$ は 0 次の変形ベッセル関数である．また，κ は分布の集中度を表すパラメータである ($\kappa \geq 0$)．式 (3) で定義される $p(d_i,d_j\mid c_i=c_j)$ に注目すると，この確率値は二つの音源の位置が近く，かつ，二つの音源が同じクラスに属している時に高い値をとる．一方，式 (4) で定義される $p(d_i,d_j\mid c_i\neq c_j)$ に注目すると，この確率値は二つの音源の位置が遠く，かつ，二つの音源が異なるクラスに属している時に高い値をとる．同時刻に二つ以上の音源がある場合 ($K_t>2$) を考慮するために，$p(\mathbf{d}\mid\mathbf{c})$ は式 (2) のようにすべての音源間の組み合わせによって定義されている．

このモデルを用いて音源のクラスを推定するときには，同時刻の他の音源のクラスを考慮する必要がある．したがって，音源クラスの MAP 推定は以下のようになる．

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmax}}\, p(\mathbf{x}\mid\mathbf{c})p(\mathbf{d}\mid\mathbf{c})p(\mathbf{c}) \tag{6}$$

4.2 提案モデルのパラメータ学習

本節では提案モデルにおける EM アルゴリズムについて説明する．まず，音響特徴量 $\mathbf{x}$ に対応するクラス c が与えられた場合，つまり教師あり学習の場合，図 2 より，ベイジアンネットワークの性質から，c は他の音源クラス c' と独立に計算することができ，従来の GMM による音響モデルのパラメータ学習と同様に学習を行うことができる．しかし，部分的なアノテーションの場合，つまり，半教師あり学習を行う場合には，c と c' が独立とはならず，$\mathbf{x}$ ごとに独立に学習することもできない．以下ではクラス c, c' がアノテーションされていない場合について説明する．

EM アルゴリズムにおいては，データセット中のサブクラス s の出現確率の期待値を計算する必要があり，以下のように表現される．

$$N_s = \sum_t\sum_{k_t} p(s_{t,k_t}=s,\mathbf{X},\mathbf{d}) \tag{7}$$

ただし，s_{t,k_t} は時刻 t の音源 k_t に関するサブクラスを表す確率変数とし，$\mathbf{X}$ は時刻 t の音響特徴量全ての集合とする．$p(s_{t,k_t}=s,\mathbf{X},\mathbf{d})$ は提案モデル上で計算することができる． ただし，図 2 より，ベイジアンネットワークの性質から，$p(s_{t,k_t}=s,\mathbf{X},\mathbf{d})$ は音源 k_t だけでなく，時刻 t におけるそのほかの音源と独立に決定することはできない．具体的に，$p(s_{t,k_t}=s,\mathbf{X},\mathbf{d})$ を計算する方法を示す．まず，ここでは簡単のため時刻 t に 2 つの音源のみがあるとして，それぞれ，音源 k_t，k'_t，音響特徴量 $\mathbf{x}$, と $\mathbf{x}'$（$\mathbf{X}=\{\mathbf{x},\mathbf{x}'\}$），音源方向 d と d' が与えられた場合を考える．すると，音源 k_t のサブクラス s に関する確率 $p(s,\mathbf{X},\mathbf{d})$ は以下のように表現される．

$$p(s,\mathbf{X},\mathbf{d}) = \sum_{c,c'} p(d,d'|c,c')p(\mathbf{x}|s)p(s|c)p(c)p(\mathbf{x}'|c')p(c') \tag{8}$$

ただし，$p(\mathbf{x}'|c')=\sum_{s'}p(\mathbf{x}'|s')p(s'|c')p(c')$ とする．二つ以上の音源がある場合，$p(\mathbf{x}|c)$ を何度も計算する必要があるため，予め依存しているフレーム全てに対して $p(\mathbf{x}|c)$ を計算し，テーブルを作っておくことで，高速に計算することができる．また， $p(s|\mathbf{x})$ は s に関する多変量ガウス分布となり，それ以外の確率は定義より与えられる．

von Mises 分布のパラメータ κ_1, κ_2 についても EM アルゴリズムで決定可能である．これらのパラメータの更新式は，混合 von Mises 分布のパラメータ推定 [Banerjee 05] と類似したものになる．κ_1 の更新値 $\kappa_1^{(new)}$ を決める式は以下のようになる．

$$U_{c=c'} = \sum_t\sum_{x,x'}\sum_{c=c'} cos(d-d')p(c\mid d,x_t)p(c'\mid d',x'_t)$$

$$V_{c=c'} = \sum_t\sum_{x,x'}\sum_{c=c'} p(c\mid d,x_t)p(c'\mid d',x'_t)$$

$$\kappa_1^{(new)} = A^{-1}\left(\frac{U_{c=c'}}{V_{c=c'}}\right)$$

ここで，$U_{c=c'}$ と $V_{c=c'}$ はモデル上ですべての同時刻の音イベントについて $c=c'$, $x=x'$ 可能な組み合わせを計算する．$A(x)$ は以下のように定義される．

$$A(x) = \frac{I_1(x)}{I_0(x)}$$

ただし，$I_0(x)$ と $I_1(x)$ はそれぞれ 0 次と 1 次の変形ベッセル関数とする．$A(x)$ の逆関数 $A^{-1}(x)$ は以下の近似式で計算可能である [Sra 12].

$$A^{-1}(x) \approx \frac{x(2-x^2)}{1-x^2}$$

κ_2 の更新式も $c=c'$ を $c\neq c'$ とすることにより，同様に計算可能である．

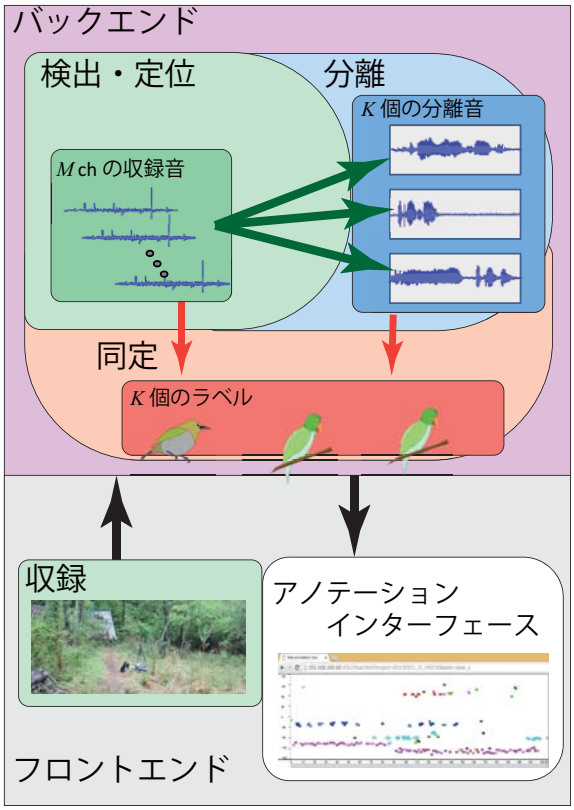


図 3: 提案モデルを用いたプロトタイプシステム

5 プロトタイプシステム

提案モデルを評価するためのプロトタイプシステムを構築した．このシステムは図 3 に示したようにバックエンドとフロントエンドからなる．バックエンドは音源検出・定位・分離・同定を行い．音源検出・定位（MUSIC 法）・分離（GHDSS 法）には HARK[1] [奥乃 10] をもちいた．フロントエンドは，収録とアノテーションからなり，マイクロフォンアレイには 7 つのマイクロフォンからなる Microcone [2] を用いた．収録は図 4 に示すように，三脚に固定し，ラップトップ PC に接続して収録を行った．アノテーションインターフェースでは，図 4　に示すように縦軸に音源方向，横軸に時間，色がアノテーションしたラベルを表すものとした．本システムでは一部にラベルをつけると残りのラベルに自動的にラベルの候補が半透明で着色されるようになっており，アノテータは予測されたラベルが正しいかどうかを確認・修正することができるようになっている．

6 評価実験

SCBPM を評価するために，データセット (A)(B) の二つの異なるデータセットを用意した．

[1]Honda Research Institute Japan Audition for Robots with Kyoto University

[2]http://www.dev-audio.com/

図 4: 収録システム

表 1: データセット (A): ラベルとイベントの数，色は図 6 と対応．ヒヨドリ（A）とヒヨドリ（B）は異なる個体のヒヨドリで，歌い方の特徴が異なるため，別ラベルとした．

ラベル	イベント数	色
キビタキ	5	red
メジロ	7	cyan
ヒヨドリ (A)	12	blue
ヒヨドリ (B)	13	yellow
その他	17	green

データセット (A) は 2013 年 5 月 5 日の朝（晴天）に愛知県都市部の公園で収録した [Suzuki 15]．この 1 分間のデータを対象として，3 節 で述べた MUSIC 法を適用することで， 54 のイベントが抽出された（図 6）．この時，MUSIC 法のパラメータは，一つのイベントが鳥の歌の一フレーズになるように選択した．これらのイベントに対し，分離音を手掛かりに，表 1 のラベルを用いて，人手でアノテーションを行ったものを正解として，データセットを作成した．データセット (B) は 2013 年 5 月 9 日の朝（晴天）にアメリカのカリフォルニア州の針葉樹とナラの混合樹林で収録した．収録した 4 分間に 140 のイベントを抽出した（図 7 ）．データセットは表 2 に示した 8 つのラベルを用いた．

図 5 と 図 8 はそれぞれデータセット (A)(B) に関して，カスケード法と SCBPM を用いた提案システムでの結果を比較したものである．これは，収録時間を 10 等分し，ラベルありの区間とラベルなしの区間に分けて，ラベルなしの区間のイベントを予測した際の正答率で評価を行った．これらのグラフの横軸（アノテーション率）は全体のうちのラベルありの区間の割合である．

ここで用いた，音響特徴量・モデルパラメータは以下のように設定した．まず，16-bit 16kHz サンプリングの分離音から窓幅 80（オーバーラップは 40 サンプル）として

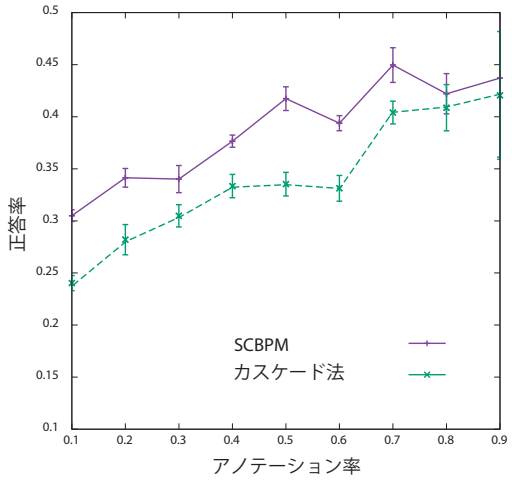


図 5: データセット (A) に関する正答率の比較

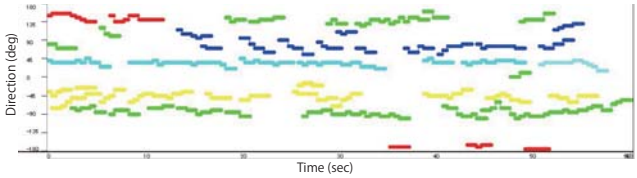


図 6: データセット (A) のイベント．ラベルは表 1 と対応．

STFT により周波数スペクトルを計算した[3]．さらに，10 フレームのステップ幅で 100 フレーム分を 1 ブロック 4100 次元のベクトルとみなし，PCA により 32 次元へと次元削減し，利用した．また，この 1 ブロックごとにクラスを推定し，最終的にイベント内の全てのブロックの多数決によってそのイベントのクラスを決定した．GMM の混合数は 30 とした．半教師あり学習ではラベルありデータの重みをとラベルなしデータの重みを設定した [Nigam 00]．ラベルありデータの重みを 1.0 とし，ラベルなしデータの重みはデータセット (A) に関しては 0.1，データセット (B) に関しては 0.001 とした．

図 5 と 図 8 からこの図から全てのアノテーション率において，SCBPM が良い正答率であることがわかる．また，データセット (B) に比べデータセット (A) の正答率がカスケード法，SCBPM ともに低いことがわかる．これはデータセット (A) のほうが同時に歌っている鳥の数が多く（図 6），完全な分離が困難であったためと考えられる．提案法は，特に，このような分離の性能が低い場合において，比較的良い性能であり，最大でおよそ 5%の差が確認できた．一方，データセット (B) では同時に歌っている鳥の数が少なく（図 7），いくつかのイベントについては他の鳥の方向と重複してしまっている．このような状況では提案法があまりよく働かないが，そのような場合

[3]これらのパラメータは人間が．鳥の歌を識別する際によく利用するパラメータセットの一つと同じものを用いた．

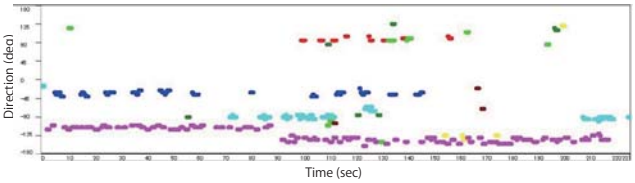


図 7: データセット (B) のイベント．ラベルは表 2 と対応．

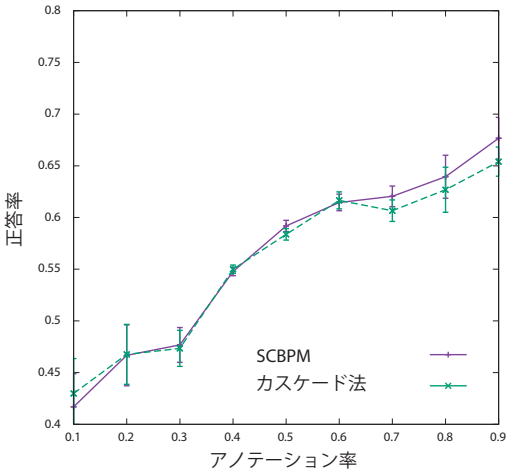


図 8: データセット (B) に関する正答率の比較

でも従来法と同等の性能が確認できた．

7 まとめ

定位情報を利用した音源同定モデルである SCBPM を提案し，モデルパラメータを分離音間の依存性を考慮しつつ学習する EM アルゴリズムを導入した．また，そのモデルを用いて音源検出・定位・分離・同定を行うアノテーションシステムのプロトタイプを構築し，実際のフィールドにおける鳥の歌のデータを用いて提案法の有効性を示した．今後は，より大規模なデータを用いたパラメータ学習を行い，音源同定の正答率を上昇させることや本システムのスケーラビリティを評価することが課題である．また，専門家に鳥の歌の分析システムを利用してもらい，評価を得ることで新たな課題の抽出を行いたいと考えている．

謝辞

本研究におけるフィールドでの収録手順や現状について助言を頂いた名古屋大学の松林志保氏に感謝する．
本研究は、JSPS 科研費 24220006, 16H02884, 16K00294 および，JST ImPACT タフロボティクスチャレンジの助成をうけた．

参考文献

[Banerjee 05] Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S.: Clustering on the unit hypersphere using von Mises-Fisher distributions, *Journal of Machine Learning Research*, Vol. 6, No. Sep, pp. 1345–1382 (2005)

表 2: データセット (B): ラベルとイベントの数．色は図 7 と対応．

ラベル	イベント数	色
Pacific-Slope Flycatcher	7	green
Spotted Towhee	8	red
Nashville Warbler	12	blue
Black-Headed Grosbeak	10	cyan
Orange-Crowned Warbler	4	yellow
Cassin's Vireo	90	magenta
Unknown bird song	6	dark green
Others	3	dark red

[Briggs 13] Briggs, F., Raich, R., Eftaxias, K., Lei, Z., and Huang, Y.: The ninth annual MLSP competition: overview, in *IEEE International workshop on machine learning for signal processing, Southampton, United Kingdom., Sept*, pp. 22–25 (2013)

[Catchpole 03] Catchpole, C. K. and Slater, P. J.: *Bird song: biological themes and variations*, Cambridge University Press (2003)

[Goëau 16] Goëau, H., Glotin, H., Vellinga, W.-P., Planqué, R., and Joly, A.: LifeCLEF Bird Identification Task 2016, in *CLEF working notes 2016* (2016)

[Holland 83] Holland, P. W., Laskey, K. B., and Leinhardt, S.: Stochastic blockmodels: First steps, *Social networks*, Vol. 5, No. 2, pp. 109–137 (1983)

[Kojima 16] Kojima, R., Sugiyama, O., Suzuki, R., Nakadai, K., and Taylor, E. C.: Semi-Automatic Bird Song Analysis by Spatial-Cue-Based Integration of Sound Source Detection, Localization, Separation, and Identification, in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*IEEE (2016)

[Nigam 00] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T.: Text classification from labeled and unlabeled documents using EM, *Machine learning*, Vol. 39, No. 2-3, pp. 103–134 (2000)

[Otsuka 14] Otsuka, T., Ishiguro, K., Sawada, H., and Okuno, H. G.: Bayesian nonparametrics for microphone array processing, *T-ASLP*, Vol. 22, No. 2, pp. 493–504 (2014)

[Pardo 06] Pardo, J. M., Anguera, X., and Wooters, C.: Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences, *Proceedings of the Ninth International Conference on Spoken Language Processing*, pp. 2194–2197 (2006)

[Schmidt 86] Schmidt, R. O.: Multiple emitter location and signal parameter estimation, *IEEE Transactions on Antennas and Propagation*, Vol. 34, No. 3, pp. 276–280 (1986)

[Sra 12] Sra, S.: A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of I s (x), *Computational Statistics*, Vol. 27, No. 1, pp. 177–190 (2012)

[Suzuki 15] Suzuki, R. and Cody, M. L.: Complex systems approaches to temporal soundspace partitioning in bird communities as a self-organizing phenomenon based on behavioral plasticity, in *Proc. of the 20th International Symposium on Artificial Life and Robotics*, pp. 11–15 (2015)

[Suzuki 16] Suzuki, R., Matsubayashi, S., Nakadai, K., and Hiroshi, O. G.: Localizing bird songs using an open source robot audition system with a microphone array, in *Proceedings of Interspeech 2016*, pp. 2026–2030 (2016)

[Uemura 15] Uemura, S., Sugiyama, O., Kojima, R., and Nakadai, K.: Outdoor Acoustic Event Identification using Sound Source Separation and Deep Learning with a Quadrotor-Embedded Microphone Array, in *ICAM2015*, pp. 329–330, JSME (2015)

[小島 15] 小島 諒介, 杉山 治, 鈴木 麗璽, 中臺 一博 : 音源アノテーション補助のための音源位置を考慮した同定モデル, in *RSJ2015*, 日本ロボット学会 (2015)

[小島 16] 小島 諒介, 杉山 治, 鈴木 麗璽, 中臺 一博 : 音源位置を考慮した音源同定のための確率モデルとその学習, in *RSJ2016*, 日本ロボット学会 (2016)

[奥乃 10] 奥乃 博, 中臺 一博 : ロボット聴覚オープンソースソフトウエア HARK, 日本ロボット学会誌, Vol. 28, No. 1, pp. 6–9 (2010)

[松林 15] 松林 志保, 鈴木 麗璽, 小島 諒介, 中臺 一博 : 複数のマイクロフォンアレイとロボット聴覚ソフトウェア HARK を用いた野鳥の位置観測精度の検討, 第 43 回 AI-Challenge 研究会, pp. 54–59, 人工知能学会 (2015)

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-06 (11/09)

UAV 搭載マイクアレイを用いた 高雑音環境下における音イベント検出・識別の並列最適化

杉山　治[1*], 小島 諒介[1], 中臺 一博[1,3]

Osamu SUGIYAMA[1], Ryosuke KOJIMA[1], Kazuhiro NAKADAI[1,2]

1. 東京工業大学 大学院 情報理工学研究科, 2. （株）ホンダ・リサーチ・インスティチュート・ジャパン

1. Graduate School of Information Science and Engineering, Tokyo Institute of Technology,

2. Honda Research Institute Japan Co., Ltd.

sugiyama@kuhp.kyoto-u.ac.jp, kojima@cyb.mei.titech.ac.jp,
nakadai@jp.honda-ri.com

Abstract

無人航空機 (UAV) に搭載したマイクアレイは, 近くにノイズを発生するローターがあるため, 常に高雑音環境にさらされる. 本稿では, このような UAV に搭載したマイクアレイを用いて音源検出・音源識別をする際に現れる特有の課題に触れ, それらを解決するための並列最適化手法を提案する. 提案システムでは，定位と識別に異なるパラメータセットを用いた並列処理機構を持ち，さらに識別に用いる畳み込みニューラルネットワークのソフトマックス層から得られる確信度によって識別するフレームを取捨選択することで，定位と識別を同時最適化する．また，UAV 搭載マイクアレイによって収集した音声を用いた実験を通じて，提案システムの有効性を示した.

1　はじめに

本稿では，無人航空機 (Unmanned Aerial Vehicle, UAV) を用いた屋外音環境理解に取り組む．UAV に搭載したマイクアレイを用い収集した音を通じて「いつ」「どこで」「なに」といった 5W1H の情報を理解することができれば，例えば，災害時に人が立ち入ることが困難な場所においても UAV で空から被災者の探索を支援することができる．このように我々は屋外環境で音源を検出し, その位置や種類を推定する技術を「屋外音環境理解」とし，屋外音環境理解のための技術を確立するための研究を行っている．UAV に搭載したマイクアレイは，近くにローター音や風切り音などのノイズを発生するローターがあるため，常に高雑音環境にさらされる．本稿では，このような UAV に搭載したマイクアレイを用いて音源検出・音源識別をするときの特有の課題に触れ，それらを解決するためのフレームワークを設計・提案する．

*現在，京都大学医学部附属病院勤務

2　音源検出と音源識別を同時に行う際に現れる特有の課題

UAV にマイクアレイを搭載する場合の大きな課題の一つとして，UAV 自身のローター音や風切り音などのノイズに常にさらされることが挙げられる．このローター音は，UAV の飛行状態にある場合に強くなり，遠方から到来する本来識別したい音イベントの音声信号の大きさはローター音と比較して小さいことが多い．結果として，UAV に搭載したマイクアレイに入力される音声信号は，常に SN 比が悪い状態で収録されることとなる．さらに，そこに環境のノイズが加わるため，UAV に搭載したマイクアレイから収録される多チャネル音声信号を用いた音源定位と音源識別手法は高雑音環境下においてもロバストに動作する必要がある．このような高雑音環境下においては，音源検出と音源識別のパラメータ最適化を同時に行うことが難しい．音源定位をする場合，音源全体を漏れなく検出するために，音のパワーが強い部分だけでなく，その前後の SN 比が低い部分も検出する必要がある．対して，音源識別をする場合には，SN 比が良い部分のみを用いた方が識別精度が高くなる．したがって，音源定位・音源識別それぞれに特化した最適化を行う必要がある．

これまでも UAV 搭載のマイクアレイを用い，音源定位，分離，識別の研究を行ってきた [1, 2, 3]．音源検出と音源定位については，マイクアレイの音源定位手法である Multiple Signal Classification based on incremental Generalized Singular Value Decomposition with Correlation Matrix Scaling (iGSVD-MUSIC-CMS) 法 [4] を提案し，10-20 [m] 程度離れた場所の音源でも UAV 搭載マイクアレイから音源の定位および検出ができることを示した．この手法に加え，これまでに音源分離手法として，マイクアレイを用いた音源分離法として高性能であることが報告されている Geometric High-order Dicorrelation-based Source Separation with Adaptive Step-size control (GHDSS-AS) 法 [5] と，深層学習の一つである畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) [6] を組み合わせることで高雑音環境下でロバストに動作する音源識別手法を提案した [7]．CNN は元来，画像の識別に特化したニューラルネットワークであるが，入力として，縦軸を周波数成分，横軸を時間成分にとったスペクトログラムを画像的特徴量を用いることで，音イベント識別でも有効に作用する．本稿では，これまで個々に検証されてきた音源定位・検出・分離・識別を一つのフレームワークとして統合し，フレームワーク全体として高雑音環境下においても音源定位・識別性能を同時最適化する仕組みを提案する.

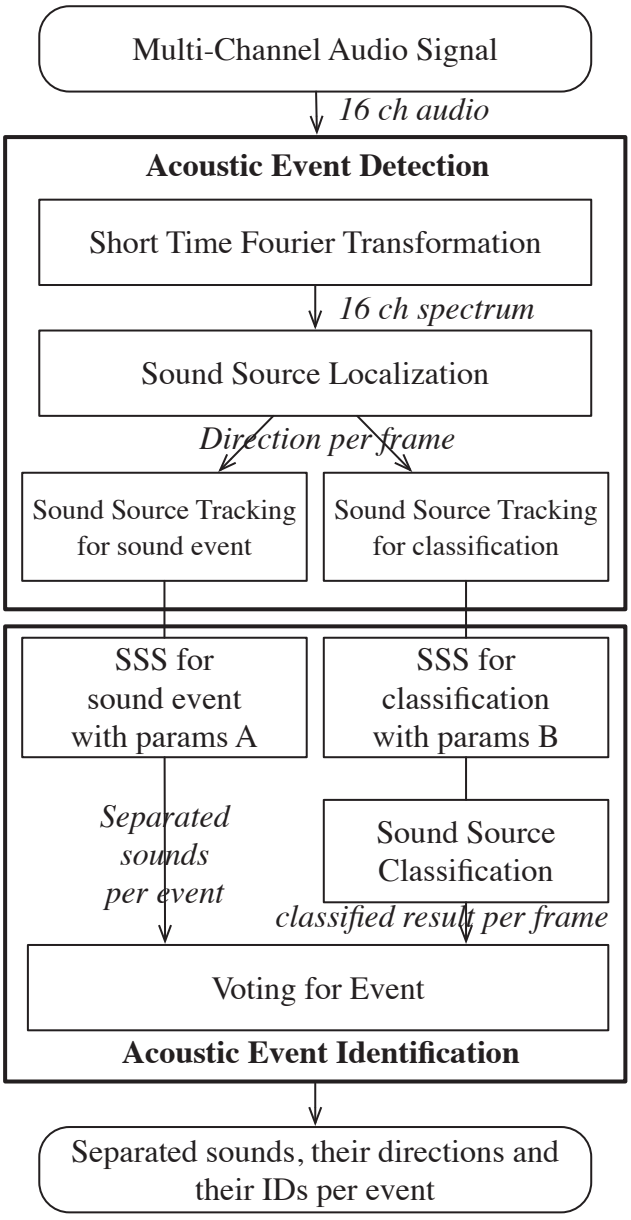


図 1: フレームワークの概要

3 高雑音環境下で音源検出と音源識別を同時最適化するフレームワーク

本稿で提案する高雑音環境下で音源検出と音源識別を行うフレームワークの概要を図 1 に示す．システムは大きく分けて，音イベント検出部 (Acoustic Event Detection, AED) と音イベント識別部 (Acoustic Event Identification, AEI) に分かれ，それぞれに音源定位・検出・分離・識別・統合の仕組みを内包する．

音イベント検出部（図 1，AED) では，短時間フーリエ変換 (STFT) を用い，各チャネルの音声信号を周波数領域に変換する．次に iGSVD-MUSIC-CMS 法を用いて，雑音を抑圧しながら音源定位を行い，MUSIC 空間スペクトログラムを得る．得られた MUSIC 空間スペクトログラムを用いて，音イベントを追跡し，音源区間を音イベント識別部 (図 1，AEI) に出力する．音イベント識別部では，まず，得られた音源区間情報と 16 ch のスペクトラムを利用し，分離音を抽出する．次に，その分離音を CNN を用いて識別し，フレーム別の識別結果を得る．これらフレーム別の識別結果と音イベント検出部から得られた音源区間情報を組み合わせて，音イベント毎の識別結果を求める (図 1, Voting for Event)．識別部の構成については，3.1 節でより詳細に述べる．また，前述したとおり，高雑音環境下においては，音源検出と音源識別において，同じパラメータで性能を最適化することが困難であるため，図 1 中破線で囲んだ領域のように音源追跡と分離に関しては，2 つの異なるパラメータを用いて並列的に区間検出用，識別用のデータを切り出す．この処理については，3.2 節でより詳細に述べる．最後に，音イベントの区間情報と，識別結果をどのように投合するかについて，3.3 節で述べる．

3.1 音源分離と深層学習を組み合わせた雑音ロバストな識別

UAV 搭載のマイクアレイから得られる多チャネル音声信号は，ロータの雑音が常に入ってくるため，SN 比の悪い状態で収録される．このように SN 比の悪い音声信号をそのまま既存の GMM のような識別器にかけても，雑音の方が主要な成分となり，識別することができない．そこで，本稿では課題を分割し，予め音源分離を行うことで，雑音抑圧された分離音を抽出し，その分離音を識別するというアプローチをとる．ここで，音源分離手法としては，前述した GHDSS-AS 法を用いる．GHDSS-AS 法はビームフォーミングとブラインド音源分離の 2 つのコスト関数を組み合わせて分離を行うため，方向性のあるスパースな音声信号（つまり，本稿で対象とする人の発話や救助を求める人工的な音声）を効率よく分離できるものと考えられる．また，識別手法としては，CNN を用いた．CNN は画像の識別に特化した識別手法であり，本稿では，縦軸を周波数成分，横軸を時間成分にとったスペクトログラムを画像的特徴量とみなして，CNN に用いる．フィードフォワード型のフルコネクションのニューラルネットワークと異なり，CNN は全結合されていないため，音声信号の識別で重要であると考えられる時間成分を明に考慮できるものと考えられる．

使用した CNN の構成を図 2 に示す．入力はログフィルタバンク特徴量 20 次元を 20 フレーム分連ねた 20×20 のスペクトログラムとし，畳み込み層 (32 カーネル)，プーリング層（最大値プーリング）を 2 層ずつ連ねた後，出力層で統合し，識別学習を行った．学習には，識別用に調整された分離音を用い，人の発話や，救助を求めるときに人が出すと考えられるホイッスルなどの音に加え，災害現場で検出されそうな救急車の音声などを交え，8 クラスの識別タスクを学習した．最終層の softmax 層の出力は，識別学習だけでなく，3.3 節の音イベントの識別（Voting for Event）にも用いられる．

3.2 定位と識別を同時最適化するための区間検出の並列処理

高雑音環境下で音源検出と音源識別をする場合，区間検出において同じパラメータで性能を同時最適化することが困難である．そこで，本稿では，音源検出と音源識別においてパラメータの異なる 2 つの区間検出処理を並列で動かすことで検出と識別の同時最適化を図る．

音源定位と音源識別で最適な音源区間の概要を図 3 に示す．図 3 に示す通り，音源区間検出では，図中の a) Th, b1) pre-margin, b2) post-margin の各パラメータを用いて，音源区間を検出する．Th は音源と雑音を分ける閾値を，pre-margin は音の開始地点のパワーが低い部分が何ミリ秒続くかを，post-margin は音の終了地点のパワーが低い部分が何ミリ秒続くかをそれぞれ表す．音源定位に最適な区間検出を考える場合，音イベントの開始・終了時点の音源のパワーは低いため，SN 比の低い区間を含んで検出する必要がある．したがって，検出音源の性質に合わせて，pre-margin, post-margin を適切に設定する必要があり，また音源と雑音をわける閾値 Th も音源全体を検出できるように低めに設定する必要がある（図 2，下部の閾値設定参照）．一方，音源識別に最適な区間検出を考える場合，SN 比が大きい音イベントの特徴がよく現れたフレームを用いて識別したほう

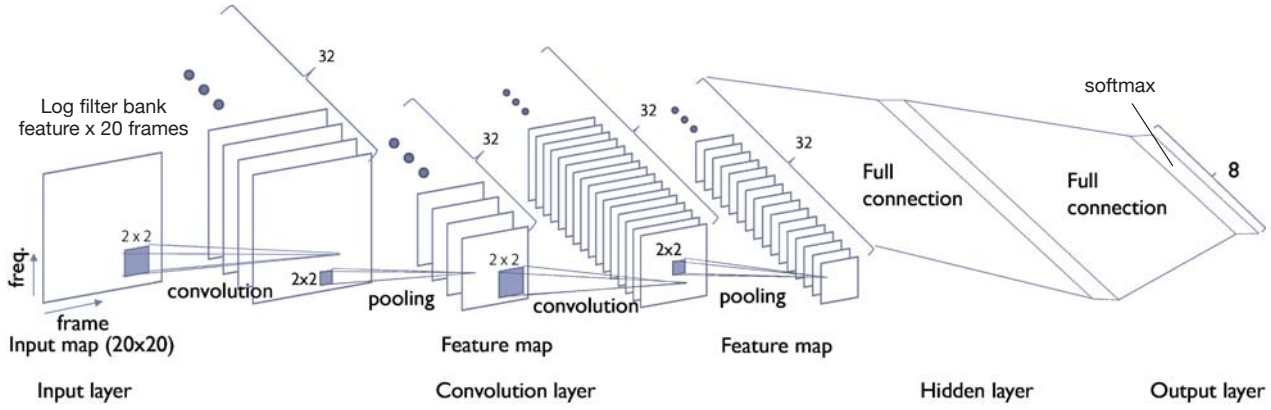


図 2: 識別学習に用いた CNN

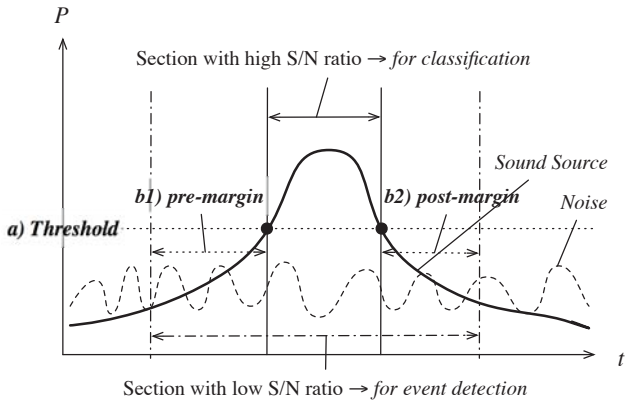


図 3: 音源定位と音源識別で最適な音源区間

図 4: 実験で試用した UAV とマイクアレイ

が良いため，pre-margin, post-margin はなるべく小さい方がよく，閾値 Th も高めに設定する必要がある（図 3，上部の閾値設定参照).

本稿では，これらの 2 つの条件には矛盾があるため同じパラメータを用いた最適化は不可能であると考え，定位用・識別用の 2 つのパラメータセット Th, post-margin, pre-margin によって，並列に定位用・識別用の区間検出を行う.

3.3 確信度が高いフレームを用いた区間識別の最適化

定位用・識別用，2 つのパラメータセットを用いた並列区間検出は最終的に図 1 の音イベント識別部 (Voting for Event) で統合される．この際，音イベント識別のための CNN は SN 比が高い音声信号に合わせて学習しているため，検知したイベントから抽出された音声特徴量全てを対象とするのではなく，図 3 で示した音の特徴を良く表現しているフレームを選択し識別を行うことが望ましい．しかしながら，観測時にはその音声信号の SN 比は未知であり，また，図 3 のモデルで示すように観測区間の中心が必ずしも SN 比が良い区間とも限らない．そこで，本研究では，SN 比の代わりに CNN の最終層であるソフトマックス層の出力のうち，argmax で選ばれたノードの値を確信度とみなし，この確信度が一定値以上のフレームのみを用いて，識別を行うこととする．CNN の畳み込み層のカーネルが各音イベントの特徴を正しく学習しているのであれば，ソフトマックス層から得られる確信度は音の特徴をよく表すフレームで高くなるものと考えられる．本研究では，実験を通して，この仮説を検証した（5.3 節参照）.

4 UAV 搭載マイクアレイによる音環境理解システム

提案したフレームワークに基づき，UAV 搭載のマイクアレイを用いた音環境理解システムを構築した．図 4 に示すように，UAV には，Asctech 社のクアドロコプタ Pelican を用いた．この機体の外周上に等間隔にマイクロホンを 16 個配置しマイクアレイとした．各マイクロホンは，黒い毛状の風防で覆うことで風切り音の低減を図った．これらのマイクロホンの信号は 16 [bit], 16 [kHz] で同期収録される．マイクアレイで収録した音声信号は，ロボット聴覚のオープンソースソフトウェア HARK (Honda Research Institute Japan Audition for Robots with Kyoto University) [8] を用いて，音源定位，音源追跡，音源分離を行い，得られた分離音を入力として Python の深層学習パッケージである chainer[1] で実装した CNN で識別学習を行った.

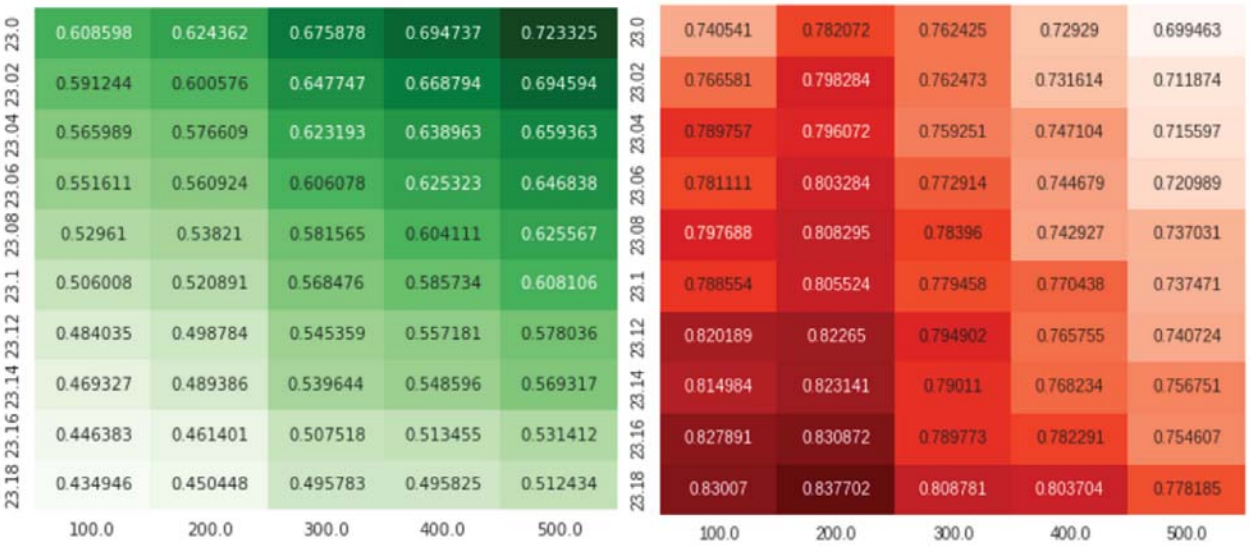
5 実験

本実験の目的は，提案したフレームワークの有効性を検証することである．本稿では，5.2 節で、提案した並列処理が有効に動作しているかどうかの検討，5.3 節で，CNN のソフトマックス層から得られた確信度を用いた音イベント識別において，識別率が向上するかどうかの検討をそれぞれ詳細に述べる.

5.1 データ収集

実験には，RWCP 実環境音声・音響データベースと電子協騒音データベースに含まれる計 8 種類の音源を用い，音声 1 種類 (男性の呼び声) と非音声 7 種類（携帯, 救急車, 拍手, 目覚まし時

[1] http://chainer.org/



a) Recall rate of Sound Source Localization

b) Precision rate of Sound Source Identification

図 5: 定位再現率と識別適合率のヒートマップ

計, シンバル, ホーン, カラスの鳴声) から構成される．音源はスピーカから 3.0-4.0 秒音源を，その後に無音区間を 3.0 - 4.0 秒それぞれ出力し，それを 1 単位として 15 回繰り返して収録を行った．収録は，屋外にて UAV をプロペラを回転させた状態で固定，UAV の中心から 45 [°] 方向，距離 3.0 [m] 地点に音源を設置し収録した. 音源定位は, 各周波数ビンでは, 同時に存在する音源数は高々ひとつであるという仮定の下, iGSVD-MUSIC-CMS を実行した (ただし, 周波数方向に統合を行ったブロードバンド空間スペクトルを用いること, および音源追跡時に一定の音源生存期間 (500 [ms]) が仮定されることから，検出結果には同時に複数の音源が含まれる可能性がある). 音源識別は, 全データの 8 割を学習に，残りの 2 割を評価に用いて，8 種類の音源を識別する 8 クラス識別タスクを行い，K-分割交差検証 (K=5) を行った.

5.2 並列区間検出の性能評価

並列区間検出の性能評価では，3.2 節で述べた Th, pre-margin, post-margin を変化させ，フレームベースの定位再現率，フレームベースの識別適合率から最適なパラメータ群を探索的に求めた．Th は, 事前検討において音源定位時に最も定位数が変化した 23.0 から 23.18 までを 0.02 刻みで変化させた．一方で，pre-margin, post-margin に関しては，100 [ms] 刻みで，100 [ms] から 500 [ms] まで変化させ，定位再現率，識別適合率の推移を検証した．そして，a) 定位用のパラメータを用いた場合，b) 提案システムで 2 つの区間検出を組み合わせた場合の識別適合率を比較し，提案手法の有効性を検討した.

実験結果を図 5 に示す．図 5 は，縦軸を閾値 Th，横軸を pre-margin, post-margin とした時の定位再現率，識別適合率の推移をヒートマップとして表したものである．ヒートマップは各値の推移を見るために，数値の高い部分ほど色が濃く，数値の低い部分ほど薄くなるように図示したもので，直感的に値の推移を読み取ることができる．図を見ると，定位再現率は，Th が低く，pre-margin, post-margin が高くなるにつれて向上するのに対し，識別適合率は，Th が高く，pre-margin, post-margin 共に低くなるにつれて向上することが示された．このことから，定位と識別を単独で最適化するパラメータセットは真逆の設定になることが示された.

次に，2 つのパラメータセットを組み合わせてシステム全体の識別性能を最大化したときの識別率の比較を図 6 に示す.

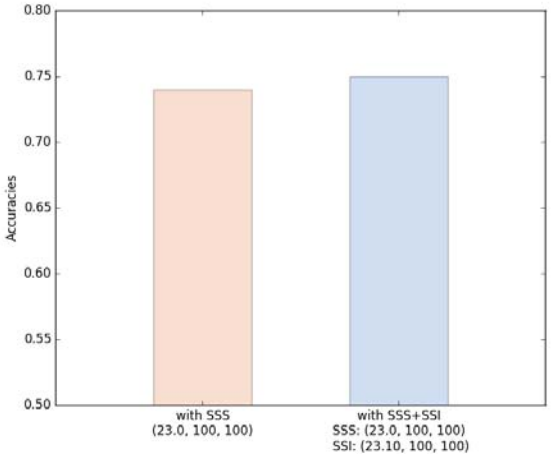


図 6: 識別性能の比較

パラメータセットは，Th を 23.0 から 23.18 まで 0.02 ずつ，post-margin, pre-margin を 100 から 500 [msec] まで変化させて，音源定位と音源識別をおこなうのに最適な組み合わせを探索し，定位：Th, post-margin, pre-margin $= 23.0, 100, 100$，識別：Th, post-margin, pre-margin $= 23.10, 100, 100$ を得た.

図 6 はで左から求めたパラメータセットのうち定位用のパラメータセットで学習した学習器を用いた識別適合率，定位と識別を異なるパラメータセットで同時最適化したときの識別適合率をそれぞれ表す．図 6 を見ると分かる通り，同時最適化した識別適合率は，定位用のパラメータセットを用いた識別適合率よりわずかながら優れた識別性能を示すことがわかる．したがって，本稿で提案する定位と識別の同時最適化機構が機能することが示唆された．一方で，思うよりも識別性能が延びていないことから，単純に定位された区間のフレームを全て用いた場合，含まれる雑音成分によって，識別性能の向上が阻害されていることが予想される．次節でより効率的な音イベント識別について詳細に述べる.

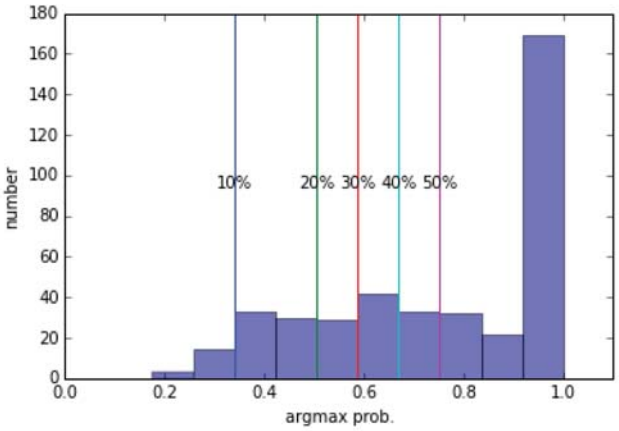


図 7: 確信度のヒストグラム

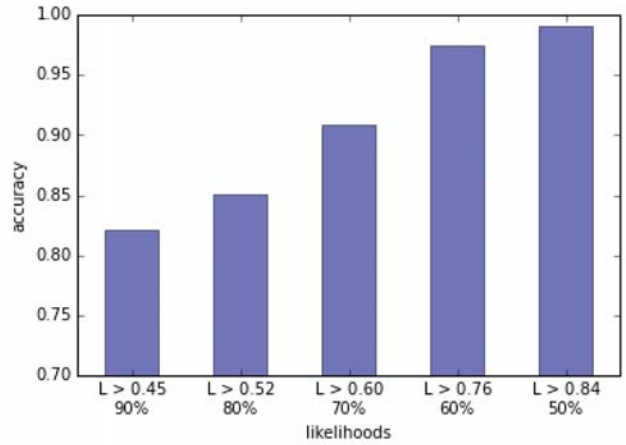


図 8: 確信度に基づいたフレームの取捨選択を行ったときの正答率の推移

5.3 確信度を用いた最良識別区間の選択と識別

次に上述の検討から得られた 2 つのパラメータセットを用いて学習した CNN と定位・区間検出した音イベントのデータセットを用いて，CNN のソフトマックス層の中で選択されたノードの値を確信度とみなしたとき，確信度の高さを閾値として最良の識別区間を選択できるのかを検証した．図 7 にデータセットを CNN に入力することで得られた確信度のヒストグラムを示す．図 7 を見るとわかるように，50%以上のデータの確信度が 0.80 を超える値を示しており，これらの区間のみを用いることで，よりよい識別性能を得ることができると考えられる．本実験では，選択区間が全体のデータセットの 90%, 80%, 70%, 60%, 50%となるように確信度の閾値（順番に 0.45, 0.52, 0.60, 0.76, 0.84）を設定し，それぞれの閾値で CNN の識別性能がどのように変化するのかを調べた．図 8 に，確信度の閾値と得られた識別性能の推移を示す．図 8 を見るとわかるように，確信度の閾値を上げていくことで，識別性能も向上していくことが示せた．したがって，本研究の仮説が検証できたものと考えられる．なお、識別性能が最も高くなるのは，確信度の閾値が 0.84 のときだが，この際，50%以上のフレームが信頼できないフレームとして棄却されるため、区間が短い音イベントなどの識別が困難になる可能性がある．確信度の閾値は，どの程度の精度の音源定位・区間検出を行いたいのかを考慮に入れつつ，調整する必要がある．

6 おわりに

本稿では，UAV 搭載マイクアレイを用いた音イベント検出・識別のためのフレームワークを設計・実装した．UAV に搭載されたマイクアレイはローター音や風切り音が混入するため，常に高雑音環境下での音源の検出と識別を行う必要がある．このような環境下でも，高精度な音イベント検出と識別を同時におこなうため，本稿では，音源分離と音源識別を組み合わせた識別手法と，音源定位から識別までを組み合わせた統一的なフレームワークを提案した．識別手法としては，雑音ロバストな定位手法である GHDSS-AS 法と CNN を組み合わせ，統合フレームワークにおいては，区間検出のパラメータとしてイベント検出と識別時に異なる値を用い，並列に処理する機構を導入した．UAV 屋外飛行実験で収集した実録音のデータを用いて実験した結果，仮説どおり，定位と識別には異なる閾値設定が必要であること，これらの異なる閾値設定を用いて，並列に定位と識別用の区間検出を行うことで，識別適合率を向上することができることを示した．また，同時に学習した CNN のソフトマックス層の出力から得られる確信度を用いることで，音イベントの特徴を表すフレームを取捨選択することができ，識別性能の向上に寄与できることを示した．

今後の課題としては様々な環境音下で学習データを作成し，雑音ロバスト性の向上を図ること，オンタイム処理を行うシステムに組み込むことなどが挙げられる．

謝辞

本研究は、JSPS 科研費 24220006,16H02884,16K00294 および，JST ImPACT タフロボティクスチャレンジの助成をうけた．

参考文献

[1] H. Nakajima *et al.*, Blind Source Separation with Prameter-Free Adaptive Step-Size Method for Robot Audition, *IEEE Trans. ASLP*, 18(6), pp. 1476-1484.

[2] 上村 他，クアドロコプタ搭載マイクロホンアレイを用いた音源分離と深層学習による音源識別，第 33 回日本ロボット学会学術講演会, 2015.

[3] K. Okutani, T. Yoshida, K. Nakamura, and K. Nakadai. Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter. In *IEEE/RSJ IROS*, pp. 3288–3293, 2012.

[4] 大畑他，相関行列スケーリングを用いた iGSVD-MUSIC 法による屋外環境音源探索の向上第 32 回日本ロボット学会学術講演会, 2014

[5] 中村 他，Latent Dirichlet Allocation と Nested Pitman-Yor Process に基づく雑音に頑健な音響イベント同定の検討，第 31 回日本ロボット学会学術講演会, 2013.

[6] S. Lawrence *et al.*, (1997). Face recognition: A convolutional neural-network approach. Neural Networks, IEEE Transactions on, 8(1), 98-113.

[7] 上村 他，クアドロコプタ搭載マイクロホンアレイを用いた深層学習による音声識別，第 15 回計測自動制御学会システムインテグレーション部門講演会, 2014.

[8] K. Nakadai *et al.* Design and Implementation of Robot Audition System “HARK”, *Advanced Robotics*, vol.24, pp.739-761 (2010).

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-07 (11/09)

言語情報を用いた談話機能推定及びロボット頭部動作生成への応用

Dialog utterance classification and nod generation for humanoid robot

劉超然[1]，カルロス石井[1]，石黒浩[1]
Chaoran LIU, Carlos ISHI, Hiroshi ISHIGURO
国際電気通信基礎技術研究所
[1] 石黒特別研究所
[1]**ATR/HIL**

chaoran.liu@atr.jp, carlos@atr.jp, ishiguro@sys.es.osaka-u.ac.jp

Abstract

コミュニケーション中の頭部動作は話者・聴者双方に置いて，会話を円滑化する役割を果たしている．遠隔操作ロボットを介した会話では，操作者側の環境と遠隔地一致しないなどの原因で，操作者の頭部動作をロボットにマッピングするのは不十分である．本稿では，発話機能を架け橋とし，発話音声から頷きを生成するモデルを提案・評価した．話者の音声発話はまず自動音声認識システムによりテキストに変換され，複数の分類器の分類結果投票によって発話機能を推定した．各発話機能クラスに置いた頷きの生起確率し従い，頷き動作パターンの分布から動作特徴を選出し，生成した動作コマンドを音声と合わせてロボットに送る．評価者実験では，提案手法により生成した動作をアンドロイドロボット ERICA（図 1）で再生し，自然さ・人間らしさを評価した．

Figure 1. Android robot ERICA used in this work.

1 はじめに

談話中に見られる発話に伴う頭部動作は，話者と聴者にとって重要な役割を果たしている．動作の表出は話者に心理的な影響を与える．身体動作によって，話者の内部で話が整理でき，条理的な会話に貢献する．聴者にとって，話者の動作をコミュニケーションチャンネルとして，相手の意図や感情を把握することができる．このように，頭部動作は発話内容と強く関連している．コミュニケーションロボットを経由した遠隔談話において，動作の理解や自然な動作の生成は談話を深め，対話者間のコネクションを強くすることができる．

遠隔操作コミュニケーションロボットに置いて，最もシンプルな動作生成は操作者の動作を計測し，ロボットにマッピングする手法である．しかし，先行研究[Tamaki 2011]で報告されたように，メディアを介した遠隔会話に置いた相槌や肯定などの意図を表す際，動作を伴わず音声のみで伝える傾向が見られた．単純な動作マッピングは対面対話時と違った動作をロボットに再現してしまう恐れがある．更に操作者が環境に反応した不意な動作も遠隔地に伝わるべきではない．故に，本稿では操作者の発話に着目し，音声から動作を生成するモデルを提案した．

2 関連研究

発話音声の基本周波数やパワーなどの韻律特徴と話者頭部動作の相関関係は，多数の研究により報告されている[Yehia 2002] [Sargin 2006] [Munhall 2004]．例えば，英語話者の頭部 6 自由度の回転と基本周波数 F0 の相関係数は 0.39 から 0.52 の間に対し，日本語話者に置いて，この相関係数は 0.22 から 0.30 区間になる．従って，F0 と頭部動作の相関性は言語に依存しており，F0 周波数のみで頭部動作を生成するのに不充分だと言える．

しかしながら，韻律情報から頭部動作動作を生成する試みも複数あった．Busso らは HMM ベースのアルゴリズムで，韻律情報から話者の感情を推定し，頭部動作を生成するモデルを提案した[Busso 2007]．渡辺らは，音声の ON-OFF 情報を入力とした頷き生成モデルを考案し，CG イージェントやヒューマノイドロボットに応用した[Watanabe 2004]．

複数の言語において，頭部動作と発話音声の関連性が報告されている．英語において，言葉を強調するとき，上昇調を使う傾向がある[Graf 202]．スウエーデン語において，強調アクセントに伴う表情や頭部動作が非強調時より多く見られる[Beskow 2006]．日本語において，発話権の変更や相槌などを考慮した以下のような談話機能リストが提案され[Ishi 2010]，頭部動作と談話機能の関連関係が報告された．

- k (keep)：話者が発話権を保持．ポーズないしはっきりしたピッチのリセットが伴う(明瞭な句境界)
- k2 (keep)：発話文の中にある不鮮明な句境界(発話権の保持)
- k3 (keep)：話者が発話末の音節を伸ばし，考えていることや発話の途中であることを表現(発話権の保持)
- f (filler)：「えっとー」「あのー」など，考え中である ことを表現する感動詞

- f2 (conjunctions) : 「じゃ」などの接続詞(文末を伸ばしていない短いフィラーとして捉えられる)
- g (give) : 当話者の発話が終了し，発話権を対話者への譲渡
- q (question) : 対話者に確認するなど応答を求める場合(発話権の譲渡)
- bc (backchannels) : 「うん」「はい」などの相槌を表現する感動詞
- su (backchannels) : 「うん」「はい」などの相槌を表現する感動詞
- dn (denial) : 「いいえ」「ううん」などの否定を表現する感動詞

この談話機能リストを用いて頭部動作を生成するモデルが提案された[Liu 2012]．本研究は，[Liu 2012]の延長線上に位置する．一部の談話機能を融合し，大きく三つのクラスに分けた．機械学習によって 談話機能クラスを推定し，頷き動作生成モデルを提案した．

3 発話機能分類

本研究では，発話プレーズを三つのクラスに分類する．一つは明瞭な句境界を有するフレーズ（上述のタグリスト中の‘k’，‘g’，‘q’に該当する），以降このクラスを‘kg’と表記する．一つは相槌で，‘bc’と表記する．最後はその他の‘o’クラス．ここで句境界を言語構造上完了した文と後続文の境界を指す．ポーズか明確なピッチリセットを伴うものが多数占めしている．クラス‘kg’と‘bc’に分類されたフレーズに頷きを生成する．

相槌を独立したクラスに分ける理由として，相槌に使われる語彙，韻律特徴及び相槌に伴う頷きにユニークなパターンが見られるためである．また，自動会話システムとして，ユーザの発話が相槌か割り込み発話かを識別する必要があり，ユーザ発話の種類に対応した発話を生成しなければならない．この研究では，ロボットが発話時の動作生成に注目し，ユーザ発話への対応は今後の研究課題として残す．

3.1 テキストのベクトル化

発話フレーズのテキストを分類器の入力として使うため，ベクトルに変換するプロセスが必要である．本研究では Latent Dirichlet Allocation (LDA) [Blei 2003] トピック用いて言語情報を表現し，分類を試みた．

LDA モデルは生成確率モデルの一種で，階層的構造を持つベーズモデルである．データセット中の文章をハイパーパラメータで決まる有限個数のトッピクの確率分布の形で表現することができる．文章中の各単語が，ランダムに選択されたトピックによって，一定の分布に従って生成された，という基本的な仮説を持つ教師なし学習手法である．LDA モデルの特徴として，トピックセットが Dirichlet 事前分布を有することを前提とし，比較的に小規模なデータセットでも訓練データにオーヴァーフィットする傾向が少ない．

LDA モデルを用いたことによって，発話フレーズをトピックの確率分布として表現することが可能になり，特徴ベクトルの次元数は Bag-of-words（BOW）表現時の辞書の長さからトピックの数まで減らすことができる．本研究では，BOW ベクトルを tf-idf [Salton 1983]によって重み付けし，LDA トピックに変換した．ハイパーパラメータの設定は先行研究[Hoffman 2010]に参考した．トピック数は 200 に設定した．

3.2 SVM分類

サポートベクトルマシン(SVM)[Shawe-Taylor 2000]を用いて発話機能の分類を行なった．SVM は広く研究されていた教師あり学習アルゴリズムで， 分類性能の優れたモデルとして知られている．SVM の学習プロセスは，訓練データの特徴空間に，クラス間のマージンが最大になる超平面を探すプロセスになる．

発話データの中，分類ターゲットとなる三つのクラスに属するサンプルの数が一様ではない．その他‘o’クラスに属するサンプルは全データの半分以上に占めしており，相槌‘kg’に属するサンプルは全体の 2 割に過ぎない．このようなアンバランスデータ対し，分類誤りを最小化した結果，サンプルの少ないクラスに対する分類パフォーマンスは期待できない．極端な場合，少数クラスのサンプルを全て多数クラスとして分類しても，全体の分類精度が大きく下がらない故である．この問題点を解決するため，分類器を訓練する前に，訓練データを cost-proportionate rejection sampling 法[Zadrozny 2003]を用いてダウンサンプリングを行ない，三つのバランスが取れたサブセットを作った．これらのサブセットを用いて，三つの SVM 分類器を訓練した．

本研究で扱う頷きクラス分類問題にとって，適合性が重要な指標となっている．先行研究[Ishi 2010]に報告されたように，頷きがよく見られる発話機能クラス‘k’，‘g’と‘bc’においての頷きの出現確率はそれぞれ 49.3%，47.1%と 84.2%である．言い換えると，比較的に低い再現率でも人の頷き傾向を再現するのに問題にならないのに対し，頷くべきではない発話フレーズで，頷きを生成すると，不自然だと評価される可能性が高い．従って，少数クラスに対する適合率と多数クラスに対する再現率が望ましい分類結果となる．その為，最終的分類結果は一般的に採用された多数投票ではなく，合意投票の結果を用いた．少数クラスである‘kg’と‘bc’クラスに対し，一つの SVM が‘false’と出力すれば，分類結果は‘false’になる（‘o’クラスに属する）．

4 動作分析及び頷き生成

本セクションでは，発話に伴う動作の分析及びそ

の結果に基づいた頷き生成モデルを説明する．

4.1　動作データ

本研究で使われている談話データセットに，発話音声と話者の頭部動作が含まれている．動作の計測にはモーションキャプチャシステムを利用した．マーカーの配置は図 2 に示す．

Figure 2. Motion capture marker set.

計測したマーカーの三次元位置時系列データに対し，頭部位置及びピッチ，ロール，ヨー三軸回転角度への変換を施した．ラベラーにより，頷きの区間が付与されている．

発話と動作の関連関係を分析するため，人工的に書き起こしたテキスト及び発話フレーズの分類（分類器を訓練する際の正解データ）を用いた．

4.1.1　データ前処理

頭部の三軸回転データの大部分は話者の頭部回転を反映しているが，上半身を移動する際，頭部の向き付随的に変わってしまう．データの前処理ではこの部分の動きを取り除く．

上半身の移動による頭部の回転は頭部動作に比べると遅くて比較的に長時間に渡るという性質を持っている．そのため短い時間で見ると，この部分の動きを線形的なトレンドと見なすことができる．各頷き区間に対し，線形回帰を施し，得られた線形成分を上半身移動と見なす．時系列回転角度からこれらの線形成分を除き，余剰成分を頭部の回転とした．

4.1.2　頷きの分析

頷きの種類を大きく二つに分けた．一つは正弦波で近似すると一周期以内のもので，‘nd’と表記する．もう一つは複数周期を含むもので，‘mnd’と表記する．データセットの中，4199 の‘nd’と 1482 の‘mnd’が含まれている．

頷きの周期を計算するため，回転角度に対し，離散フーリエ変換を施した．クラス‘kg’と‘bc’の発話に見られた頷きの周波数と角度の平均分散を図 3 に示す．

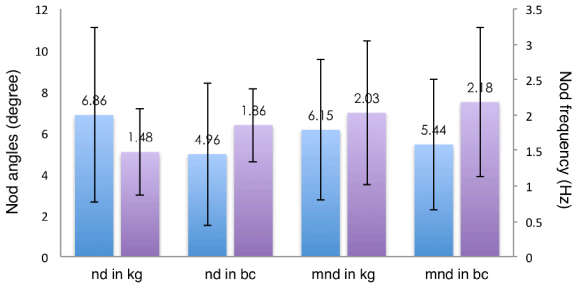


Figure 3. Averages of nod angles and frequencies. Error bars denote standard deviations

‘kg’クラスに見られた‘nd’と‘mnd’の周波数の平均値はそれぞれ 1.48Hz と 2.03Hz で，‘bc’クラスの平均値は 1.86Hz と 2.18Hz である．全体的に‘bc’クラスに見られた頷きは‘kg’クラスより速く，‘mnd’は‘nd’より速い傾向である．頷きの角度に関しては，周波数と相反した傾向が見られた．

全データにある頷き周波数のヒストグラム及び周波数と最大角度の分布を下図に示す．

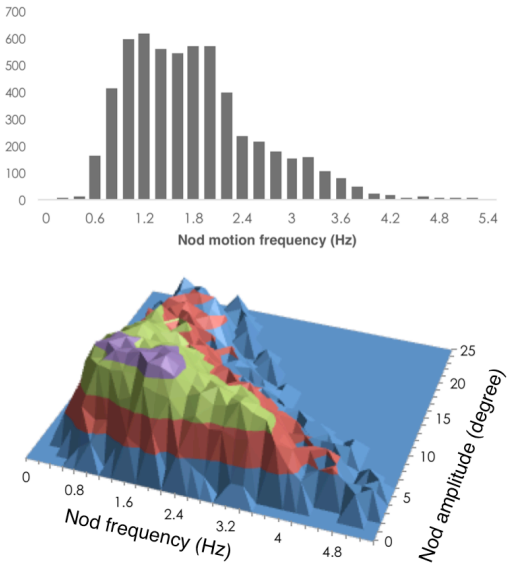


Figure 4. Histogram of nod frequencies and co-occurrence pattern between frequency and amplitude.

4.1.1　音声との関連

発話音声の持続時間，発話スピード，相対振幅，パワー変化率などの音響特徴と動作の相関関係を調べたが，大多数の特徴量に関して，強い相関関係が見られない．一つだけ動作周波数と正の相関関係が確認できたのは，‘bc’クラスにおいた音声パワー変化率である．図 5 に‘kg’と‘bc’両クラスの音声パワー変化率と頷き周波数の分布を示す．

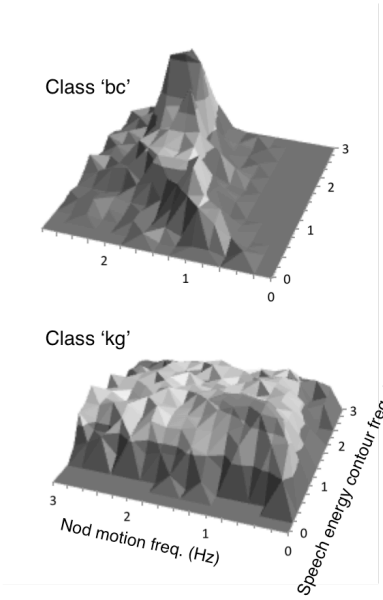


Figure 5. Co-occurrence pattern of nod motion frequency and speech energy contour frequency for classes 'bc' and 'kg'.

'kg'クラスにおいて，音声パワーの変化率と動作周波数の分布は一様分布に類似したのに対し，'bc'クラスにおいて，特に低周波区間で強い相関が見られた．この相関は相槌時繰り返した言葉（"はいはいはい"や"うんうん"など）と共起した頷きだと考えられる．

4.3 頷き生成

対話・頷きデータの分析結果を利用し，発話時の頷き生成モデルを提案した．当モデルは発話音声を入力とし，頷き動作を出力する．全体の処理の流れをブロック図に示す．

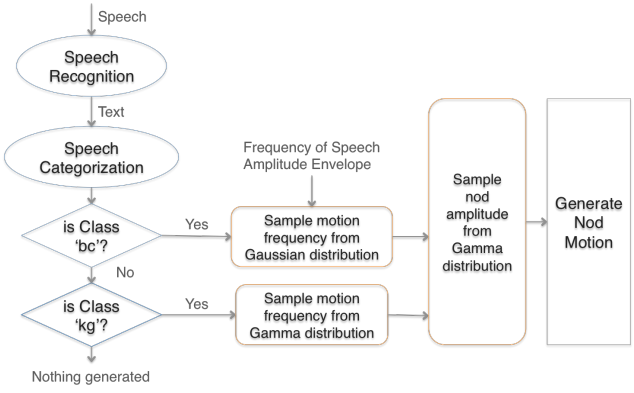


Figure 6. Block diagram of the proposed model.

まず発話音声を音声認識エンジンに送り，テキストに変換する．その後セクション 3 に説明した分類器により，発話機能クラスを推定する．分類結果はクラス'o'の場合，動作を出力せず，次のフレーズに移行する．分類結果は'kg'及び'bc'の場合，頷きの動作を生成する．具体的な流れは以下に説明する．

発話機能分類の結果はクラス'bc'の場合，音声パワーの変化率f_sを利用し，頷きの周波数f_{nd}を決める．図 5 で示した分布を正規雑音が載った線形回帰モデルで近似する．f_{nd}は正規分布$\mathcal{N}(\mu_{fs},\sigma)$からサンプリングする．この中，$\mu_{fs}$以下のように決める．

$$\mu_{fs} = (\boldsymbol{F}_s{}'\boldsymbol{F}_s)^{-1}\boldsymbol{F}_s{}'\boldsymbol{F}_{nd}\times f_s \quad (1)$$

$\boldsymbol{F}_s$と$\boldsymbol{F}_{nd}$は観測データ中の音声パワー変化率と頷き周波数を表す．線形回帰モデルから正規分布$\mathcal{N}(\mu_{fs},\sigma)$の期待値と分散が得られる．

又，発話機能分類の結果はクラス'kg'の場合，図 5 に示したように，頷き動作の特徴と音声特徴に強い相関関係見られないため，頷きの周波数f_{nd}を図 4 に示した全体分布からサンプリングする．この分布をガンマ分布$\Gamma(\alpha,\beta)$で近似する．正規分布ではなくガンマを選んだ理由は，データ分布の形及び周波数パラメータが非負である必要があるため．

頷きの動作を生成するのに，周波数と動作角度が必要であるため，上記の手法で動作周波数f_{nd}を決めてから動作角度を抽出する．各周波数に対応した角度の期待値は図 4 で分かるように，線形回帰を使って求めることが可能である．具体的には，角度をガンマ分布$\Gamma_b(\alpha_b,\beta_b)$からサンプリングする．$\alpha_b$と$\beta_b$は以下のように決める．

$$\frac{\alpha}{\beta} = k_1 f_{nd} + \zeta_1 \quad (2)$$

$$\beta \propto f_{nd} \quad (3)$$

k_1とζ_1は線形回帰で得られるパラメータである．

最後に，頷きの動作を上記の周波数と振幅を持つ正弦波で近似する．人の頷きは下方への回転量が上方より大きいため，正弦波開始位相を$\pi/4$とする．

5 評価実験

生成した頷き動作の自然さを評価するため，被験者実験を行った．先行研究で提案された正解談話機能ラベルに基づく頷き生成モデルを比較対象として用いた．

5.1 実験設定

発話の違いによるバイアスを排除するため，実際の対話ではなく，収録した発話を使って，動作生成を行った．9 の発話フレーズをランダムにデータベースから抽出した．比べ安いように，各フレーズの長さは 15 秒から 30 秒に制限した．

一つの発話に対し，四つの動作を生成した．二つは先行研究で提案された頷きタイミング制御モデルによって生成された．頷き動作は一般的に見られた

人の頷きサンプルを使用した．動作角度を図 7 に示す．その中一つは先行研究のように正解ラベルを利用し，動作の生成を行った．もう一つはセクション 3 で説明した分類結果を用いた．

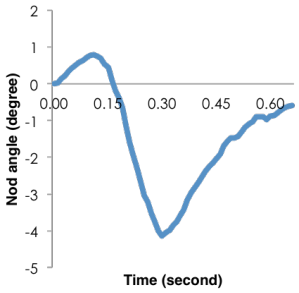


Figure 7. Fixed nod example used in previous work.

残りの二つの動作は本研究で提案した手法によって生成された．頷きの周波数，長さ，振幅は確率分布からサンプリングするので，具体的な動作は生成するたびに変わるため，同様な手法で二つの動作を作成し，検証した．正規分布からサンプリングする際，確率が低いものの，マイナスな値が得られる場合がある．そのようなサンプルはリジェクトして，再サンプリングする．また動作の周波数や角度がロボットの可動範囲を超えたようなサンプルもリジェクトする．

発話音声をテキストに変えるため，Google speech api v2 を利用した．9 の音声サンプルの認識精度は単語ベースで 64.2%であった．これらのテキストをセクション 3 で説明した 200 次元 LDA トピックに変換され，三つの分類器に送った．分類精度は 68.1%であった．分類実験時の精度より低い理由は，音声認識エラーの影響があると思われる．

生成された動作はロボットのアクチュエータコマンドに変換され，20ms 間隔でロボットに送られる．本実験ではアンドロイドロボット ERICA を使った．アクチュエータ配置は図 8 に示す．発話時の口唇動作は「」で提案された手法を用いた．

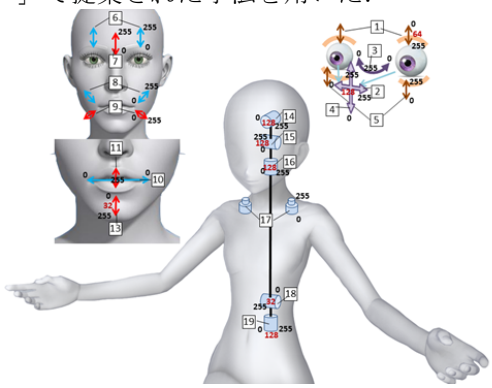


Figure 8. Actuator map for android robot ERICA.

9 の発話サンプル毎に 4 の動作タイプを生成し，ロボットで再現し，ビデオを収録した．21 人の被験者に発話毎にシャッフルしたビデオを呈示し，動作の自然さを 7 段階で評価した．‘1’は最も不自然で‘7’は最も自然を表す．

5.1 実験結果

図 9 に四つの動作タイプに対した主観評価結果を表す．左から一つ目は先行研究で提案された正解ラベルに基づいた頷き生成モデルで，‘fixed-gt’と表記する．二つ目は同じく図 7 に示した頷き動作を利用したもので，‘fixed’と表記した．正解ラベルの代わりに分類器の推定結果を用いた．発話機能分類時，頷きクラス（‘kg’，‘bc’）に対して，適合率を重視したため，一部の頷きクラスの発話が頷かないクラスに分類される傾向がある．その為，‘fixed’動作タイプに含まれる頷きの回数は‘fixed-gt’より少ない．三つ目と四つ目は本研究で提案したモデルによって，生成された動作で，‘sampled1’と‘sampled2’と表記する．

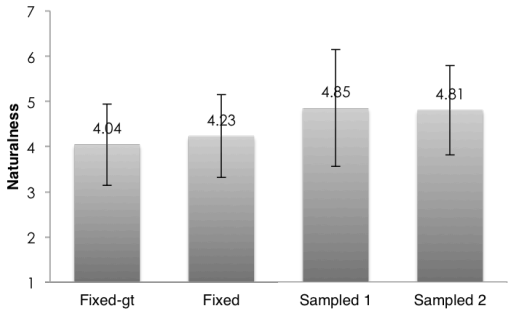


Figure 9. Perceived naturalness for each motion type.

評価結果に対し，有意水準を 0.05 に設定した被験者内一要因分散分析を施した結果，各条件間に有意差が見られた（$F(3,20) = 5.18, p = 0.003$）．

多重比較の結果，同提案手法によって生成された‘sampled1’と‘sampled2’に有意差が見られない（p=0.89）．この結果は平均と分散からも容易に見られる．‘fixed-gt’と‘fixed’の間有意差が見られないが，推定結果を使った‘fixed’は正解ラベルを用いた‘fixed-gt’より高い評価平均値が得られた．人の対話データに，‘kg’と‘bc’クラスの発話に頷き生起確率は約 50%と 80%だったため，正解ラベルを使ってすべての頷きクラスの発話に頷きを生成すると，頷きすぎでかえて不自然になる可能性がある．最後に提案手法の評価結果はこの二つの動作より有意に高い（$v.s\ 'Fixed - gt': p = 0.012$，$v.s\ 'Fixed': p = 0.039$）．

不完全な音声認識結果を用いても，本研究で提案した頷き生成モデルを使って，自然な頷き動作とタイミングを制御することが可能であること，この評価実験の結果が示した．

5 おわりに

発話に付随した頭部動作は，対話者間の感情や意図の伝達に重要な役割を担っている．又，遠隔地にいる人のアバターとしてのヒューマノイドロボットにとって，自然な動作はスムーズな会話に強く影響

する．本研究では発話音声を入力とした自然な頷き動作を生成するモデルを提案した．

人の対話データから，頷きが主に発話文のクリアな句境界や相槌フレーズに生起することが分かった．この発見に基づいて，発話フレーズを句境界文・相槌・その他三つのクラスに分類した．収集した対話データからサンプリングしたサブセットを用いて，複数の SVM 分類器を訓練した．言語情報は LDA トピックの確率分布を利用してベクトライズし，SVM の特徴ベクトルとして使った．複数の SVM 出力の合意投票によって，最終的なクラス分類結果を決める．

人の頷き動作の周波数分布は，ガンマ分布で近似することが可能であり，各周波数に対応した頷きの振幅（最高点から最低点の角度）もガンマ分布に見なせることが，人の対話データから分かる．又，相槌の場合，発話音声のパワー変化周波数が低い発話において，頷き周波数とパワー変化周波数に強い相関が見られた．これらの結果より，発話音声の特徴によって，ガンマ分布若しくは正規分布から頷きのパラメータをサンプリングして，正弦波で頷きを近似する手法を提案した．

提案法の有効性を検証するため，被験者実験を行った．実験結果は，提案の頷き生成モデルは、先行研究の人工的に付与した正解ラベルを使ったモデルより自然な動作を生成できることを示した．

今後の課題として，発話音声から他の動作パターンを生成するや，発話と動作の同期メカニズムを探究することが予定している．

謝辞

この研究は JST，ERATO，石黒共生ヒューマンロボットインタラクションプロジェクトの一環として行われたものです．この研究の一部は JSPS KAKENHI (25220004)の助成を受けたものです．

参考文献

[Tamaki 2011] Tamaki, H., Higashino, S., Kobayashi, M., Ihara, M. Reducing Speech Contention in Web Conferences. *Applications and the Internet*. 75-81. 2011.

[Yehia 2002] Yehia, H.C. Kuratate, T. Vatikiotis-Bateson, E. 2002. Linking facial animation, head motion and speech acoustics. *J. of Phonetics*, Vol. 30, pp. 555-568, 2002.

[Sargin 2006] Sargin, M.E. Aran, O. Karpov, A. Ofli, F. Yasinnik, Y. Wilson, S. Erzin, E. Yemez, Y. Tekalp. A.M. 2006. Combined Gesture-Speech Analysis and Speech Driven Gesture Synthesis. In *proc. IEEE International Conference on Multimedia*, 2006.

[Munhall 2004] Munhall, K.G. Jones, J.A. Callan, D.E. Kuratate, T. Vatikiotis-Bateson, E. 2004. Visual prosody and speech intelligibility – Head movement improves auditory speech perception. *Psychological Science*, Vol. 15, No. 2, pp. 133-137, 2004.

[Busso 2007] Busso, C. Deng, Z. Grimm, M. Neumann, U. Narayanan, S. 2007. Rigid Head Motion in Expressive Speech Animation: Analysis and Synthesis. IEEE Trans. on Audio, Speech and Language Processing, March 2007.

[Watanabe 2004] Watanabe, T. Okubo, M. Nakashige, M. Danbara, R. InterActor: Speech-Driven Embodied Interactive Actor, International Journal of Human-Computer Interaction 17 (1) (2004) 43–60.

[Graf 202] Graf, H.P. Cosatto, E. Strom, V. Huang, F.J. 2002. Visual prosody: Facial movements accompanying speech. In *proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition (FGR'02)*, 2002.

[Beskow 2006] Beskow, J. Granstrom, B. House, D. 2006. Visual correlates to prominence in several expressive modes. In *proc. Interspeech 2006 – ICSLP*, pp. 1272-1275, 2006.

[Ishi 2010] Ishi, C.T. Liu, C. Ishiguro, H. Hagita, N. 2010. Head motion during dialogue speech and nod timing control in humanoid robots. In *proc. of IEEE/RSJ Human Robot Interaction (HRI 2010)*, 293-300, 2010.

[Liu 2012] Liu, C. Ishi, C. T. Ishiguro, H. Hagita, N. Generation of nodding, head tilting and eye gazing for human-robot dialogue interaction. In *Proc. of HRI 2012*, pp. 285-292, Boston, March, 2012.

[Blei 2003] Blei, David M. Ng, Andrew Y. Jordan, Michael I (January 2003). Lafferty, John, ed. "Latent Dirichlet Allocation". *Journal of Machine Learning Research* 3 (4–5): pp. 993–1022. 2003.

[Salton 1983] Salton, G. and McGill, M. editors. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[Hoffman 2010] Hoffman, M., Bach, F. R., Blei, D. M. Online learning for latent dirichlet allocation. Advances in neural information processing systems. p. 856-864. 2010.

[Shawe-Taylor 2000] Shawe-Taylor, J., Cristianini, N. Support vector machines and other kernel-based learning methods. Cambridge University Press, 2000.

[Zadrozny 2003] Zadrozny, B., Langford, J., Abe, N. Cost-sensitive learning by cost-proportionate example weighting. In Proc. Of the Third IEEE International Conf. on Data Mining. 435-442, 2003.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-08 (11/09)

Sequential Deep Learning for Dancing Motion Generation

Nelson Yalta[*1], **Tetsuya Ogata**[*1], **Kazuhiro Nakadai**[*2]
Waseda University[*1], Honda Research Institute Japan Co., Ltd.[*2]
{nelson.yalta@ruri., ogata@}waseda.jp, nakadai@jp.honda-ri.com

Abstract

In recently years, robots have invaded our life in many activities such as manufacture process or social activities as dance. In dancing, robots have shown a good performance following the rhythm of a music using beat-tracking algorithm. In this work, we show an implementation of a deep learning based on sequential learning model for dancing robots. The model is trained without middle states mixing audio information and captured motion from a person, and it can substitute a beat-tracking algorithm. The model generates quasi-realistic dance pattern motion without constraints from the music information and its start position, and following the rhythm beat from a multiple sound source music.

1 Introduction

During last decades, robots have been involved and become part of humans everyday life, possible to perform dangerous tasks instead of humans, or appearing in the media. As robots becomes part of humans everyday life, the should interact constantly with humans. Thus in order to improve this interaction, dance presents as a bound between humans and robots [Oliveira 2015], and for dance performing they should be able to listen with its own ears live sounds. While robots perform a dance, they follow the rhythm on a real-environment robustly responding to music, and generating continuous movements. Some techniques which allows robots to dance, are based on beat-tracking implemented for real time execution. Beat tracking systems are implemented to track the beat timing of music pieces and synchronize movements of a dancing robot with it. Real-time application [Oliveira 2015], allows to track the beat timing, but it also overcome with noise problems and obtain the desired audio information from multiple sound sources, implementing a robust general framework. Beat tracking systems allow robots to perform dance, but also to play instruments [Itohara 2012]. Through a multimodal system, which uses audio and video information, allows robots to play an instrument e.g. a guitar. Also, a method which can generate automatic choreographies by music content analysis was introduced at [Fukayama 2015]. They proposed a probabilistic framework which generate automatic choreography without constraints, and which can satisfy both music content and motion connectivity. These implementations synchronize with the beat timing, a motion that is previously generated and stacked in a database. In this paper, we implemented a system based on deep learning to generate motion positions for dancing robots, directly from an audio information. Deep learning (DL) based models called deep neural networks (DNN) and deep convolutional neural networks (DCNN) have shown successful performance on many signal processing fields such as computer vision [Krizhevsky 2012, Clevert 2015], speech recognition [Sainath 2015a], and natural language processing [Sutskever 2014, Venugopalan 2015]. DL based models implemented for natural language processing task has shown a remarkable performance, demonstrating that can translate correctly long sentences [Sutskever 2014] on translation task, using deep Long-short term memory (LSTM) models. Or also being able to extract information converting video to text [Venugopalan 2015], mixing a DCNN with a deep LSTM (DLSTM) for the task. Generating smooth realistic dance motion shows as a constraint for implementations which only use the audio information. It should be able to track the beat timing, and at a same time separate the desired sound information from a multiple audio input. Thus, we focus on implementing a DL model taking advantage of the sequence-to-sequence model introduced at [Venugopalan 2015]. Implementing a model for video-to-text requires a previous pre-trained DCNN model, which increase the training time i.e. learning cost. For implementing, we remark that is possible to replace a beat tracking system with an end-to-end trained DL model, which uses the audio information as input and mixing with the current position, generates the next position.

The DL model generates a quasi-realistic motion pattern without constraints, separating from a multiple sound source audio, the beat timing performed by drums.

2 Proposed Method

DL models have shown an outstanding achievement in different tasks which have been involved, and sound signal processing task such as speech recognition, they showed outstanding performance[Abdel-hamid 2012, Sainath 2015b]. DL models trained without middle states i.e. end-to-end training, has become the state-of-art in multiple signal processing areas. They can process information from the same field, but also in multimodal fields such as video-to-text task. Thus, we replaced a beat tracking system that synchronize the beat timing and a stacked motion pattern, with a DL model trained end-to-end. The motion pattern of the task is a combined vector of the normalized spatial position of a dancer, and the joint rotations expressed in quaternions obtained from a kinect device. A mixed model which a deep LSTM network is connected after the plain DCNN but not connecting it in a sequential mode, generated random movements from the audio information. However, the motion patterns are not linked with each other and the generated motions are rough. Then, we replaced it with a similar structure implemented at [Venugopalan 2015], and modified the activation functions. Finally, we trained it end-to-end using the audio as input and a sequential motion pattern as the target.

2.1 Deep Learning

Recently, factors such as the development of improved optimization algorithms, the availability of open-source DL libraries and so on, have made the employment of DL possible for classification or recognition problems. DL based models as DNN and DCNN approaches, has shown that can surpass even humans performance. Optimization algorithm which become part of the model such as batch normalization, or training algorithm such as ADAM, or adding noise to the gradients has allowed to improve the performance of DL task on different signal processing fields. Batch normalization introduced at [Ioffe 2015] is a mechanism that allows training with higher learning rates without fast overfitting risk and also can accelerate the training as makes possible to train deeper models. Adaptive moment estimation i.e. ADAM optimization was introduced at [Kingma 2014]. This optimization method allows to implement a robust optimization with a little memory requirement. As ADAM is an algorithm for first-order gradient-based optimization, this method is straightforward to implement and can be used on machine learning problems where it is used large datasets or with high-dimensional parameters models. Adding gradient noise in very deep models [Neelakantan 2016] improves its learning, but it helps generalization and training on complicated neural networks, as it also improves the learning from a poor initialization with almost a zero computational cost. Open-source DL frameworks has allowed researchers to use DL on different tasks. DL frameworks such as Chainer [Tokui 2015], allows to implement flexible DCNN and trained it with multiple optimization algorithm such as Batch Normalization, ADAM optimization or adding noise to the gradient during the training.

2.2 Nonlinearities Activation Functions

Recently researches has shown that the performance of a DCNN also depends on the type of nonlinearity function used in the model. Novel nonlinear activation functions (Figure 1) has been introduced [Clevert 2015, He 2015], improving the accuracy and performance of DCNN. Rectifier neurons e.g. Rectified Linear Unit (ReLU) nonlinearity have shown successfully performance on vision computer [Krizhevsky 2012], but also for sound tasks. The ReLU activation function is defined by:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \tag{1}$$

ReLU activation outputs are non-negative, thus the mean activation is larger than zero, and because of not be zero-centered it can speed down the learning. Then, a nonlinearity called Parametric Rectified Linear Unit (PReLU) was introduced at [He 2015], which thus the implementation of a negative part activation it can speed up the learning. Later, Exponential Linear Unit (ELU) introduced at [Clevert 2015], was showed as a non-linearity that can improve learning characteristics compared to other linear activation functions. ELU defined as:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ a(exp(x) - 1) & \text{if } x < 0 \end{cases}, \tag{2}$$

Where a is a fixed value, and controls the saturation for negative inputs, and *exp* is the exponential function of e. It was showed that networks implemented with ELU activations can speed up learning as bring robust generalization performance compared to ReLUs and PReLUs.

2.3 Sequential Learning

Sequential Learning models i.e. sequence-to-sequence (SEQ2SEQ) model (Figure 2) has been applied for

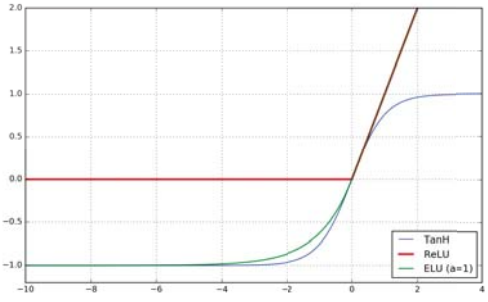


Figure 1: Nonlinearities Activation Functions

natural language processing task such as foreign language translation task [Sutskever 2014] or video-to-text task [Venugopalan 2015], and speech recognition [Graves 2014]. DL models has shown remarkable performance in different task such as speech recognition, where a DNN with Convolutional Neural Networks (CNN), LSTM and fully connected layers has better performance than LSTM models to perform the task [Sainath 2015a]. However, sequential problems such as speech recognition or machine translation are best expressed on sequence with unknown length, and DNN could present limitations due to the fixed dimensionality of its input and targets outputs. SEQ2SEQ models based on LSTM, was introduced as an architecture that can solve general sequential problems. SEQ2SEQ introduced at [Cho 2014] was implemented with Recurrent Neural Networks (RNN) to solve these problems. Despite of RNN based models theoretically can be applied for sequential problems, their performance gets reduced due to the long-term dependency on larger sequences. However, SEQ2SEQ implemented with LSTM models [Graves 2014] have shown better performance in many sequential tasks. LSTM cell unit (Figure 3) is defined as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (6)$$

$$h_t = o_t \otimes tanh(c_t) \quad (7)$$

At a time t, the LSTM computes a hidden state h_t and a memory cell state c_t, for an input x_t. σ is the sigmoid function, and i, f and o are respectively the *input*, *forget* and *output* gate vectors. c is the *cell activator* vector and W is the weight matrix which corresponds to each vector. In applications, such as machine translation, two LSTM layers are used. One LSTM layer is connected to the input and obtains a fixed representation on a time step. Then, the second LSTM layer, which is conditioned to the output of the first, is a recurrent neural network language model. At the same time, it is shown that deep LSTM has significantly performance than shallow LSTM [Sutskever 2014], where each one LSTM layer is replaced by a four LSTM layers. Sequence-to-sequence models has been also tested on information extraction task, they can generate captions from videos [Venugopalan 2015]. Regarding to use only a DLSTM for the task, a DCNN model previously trained, sequentially obtains the features from the frame image of a video, and then the DLSTM generates sequential words. The model learns to associate a sequence of words with the temporal structure of the frames, allowing to handle variable-length inputs and outputs.

2.4 Model Architecture

Beat tracking system can be replaced by DL models for movement generation and synchronization. However, a DL model based on a sequential learning can improve

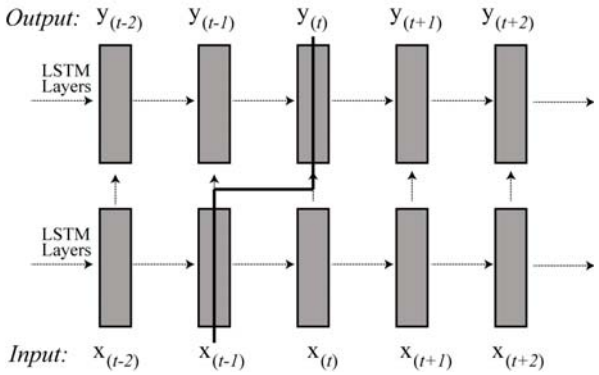


Figure 2: Sequential Learning. The thick line shows the information flow.

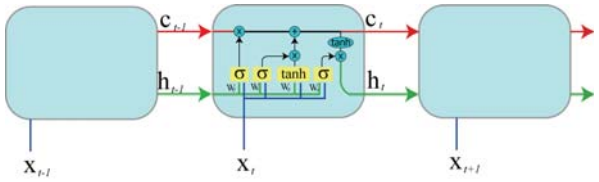


Figure 3: LSTM Cell

the performance and generate smooth motion pattern while it separates the beat information from a multiple sound source audio piece. So we implemented a DCNN +DLSTM model based on [Venugopalan 2015]. The audio has only one input channel. After the input, we connected four Convolutional Layers each one with 33×2 dimension of kernel size. The size of the kernel is empirically set up. After each convolutional layer, a batch normalization and the activation function is implemented. ELU, TanH, and ReLU has been evaluated as activation function of the DCNN part. After the fourth layer, the output has a dimension of 65×1, and this become the input for the DLSTM section. Each block of the sequential section has three LSTM layers with 500 units. LSTM layers with 250 and 1000 units has been also tested. After each LSTM block, a fully connected layer is stacked. Different activation functions have been also tested on the fully connected layer. Each block is connected as showed in Figure 4, where the output of the current last fully connected layer is concatenated with the output of the next first block and it becomes the input of the next second block.

3 Experiments

3.1 Data Preparation

For training and evaluating, we used different latin music genre such as salsa, cumbia, etc. For training, the music tracks are combined with only drum, instrumental and with singer tracks. For visualizing the generated motion pattern, we used a 3D model implemented on a simulator. For preparing the training dataset, we obtained the position information using a Kinect from a person, and synchronized with the music. The position is sampled

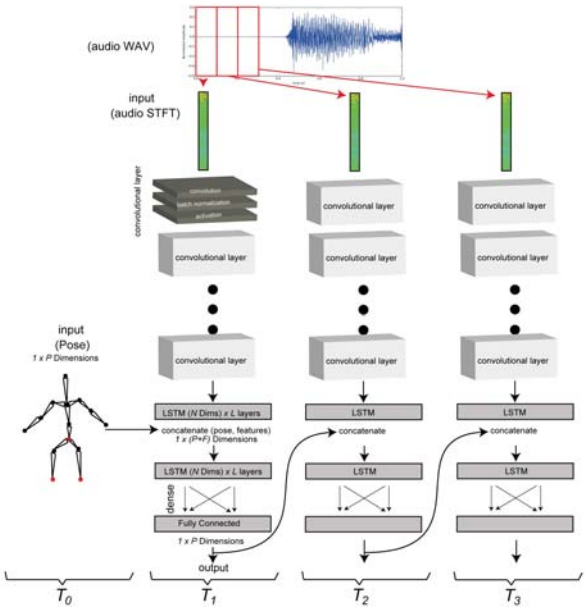


Figure 4: Model Architecture

with a frequency of 30 frames per second (FPS) and the music has a sampling frequency of 16KHz. To synchronize the music with the position, we used a slice of the audio corresponding to the same time of the position. We prepared the input data of the network using shot-time fourier transform (STFT) on the audio slice and the normalized position was used as target. As for the input, we pre-processed according the following steps:

- To synchronize the audio with the positions, we extracted a slice with length of 534 samples (33ms), which corresponds to its position. This extracted slice is converted on H STFT frames of 160 (10ms) samples with a hop of 80 samples (5ms).
- From the STFT, we used the power information, which is normalized between 0.1 and 0.9 on the W frequency bin axis.
- We stacked the H frames, thus the input of the network becomes a $1 \times W \times H$ dimensional input file.

As for the output, we prepared the position vector target as following:

- Using a Kinect device, we capture the D positions angles of a person. From the position angles, we obtain the spatial information of 3 body parts in a vector *(x, y, z)* on meters. And 14 rotations vector in quaternions with denotation (q_x, q_y, q_z, q_w).
- For each vector component, we normalized it using the maximum value of each component, between a range of -0.9 and 0.9.
- Then, the target becomes a vector of 65 dimensions.

3.2 Training Process

For our experiments, we prepared multiple training files using the STFT from around 15 music pieces as input, and a corresponding sequence motion as target. For training, each sequence step has an input of 1 channel audio $\times$ 129 frequency bins $\times$ 5 frames, and to predict we set the next corresponding vector position as target. We trained different end-to-end networks (Table 1) using ADAM solver [Kingma 2014], and Mean Squared Error is set as loss function. No fine-tune or previous training was used for the experiments. We tested different parameters during the training such as the activation function, the training sequence, the units of each LSTM layer, and the number of frames using on the input. ELU, ReLU and TanH nonlinearities are evaluated as activation functions. A sequence of 150 steps is set for the trainings. However, we also evaluated 50 and 100 steps configuration. We used 500 units for each LSTM layer, but we also tested with 250 and 1000 units. As for the input, we used a model to predict the next position using only the previous audio slice. The initial alpha for the solver was set to 10^{-4}, and the target vector was set 45 dimensions. We added noise to the gradient. We compared the training of the models (Table 1) mixing different genres and one same genre on the training dataset. We trained the models for 10 epochs, using a sequential minibatch with a size of 50 files. Each training took approximately 20 hours using a GPU NVIDIA Tesla K80.

Table 1: Models architecture configurations

layer name	output size	12 layers		
input	129×5			
conv1	97×4	33, 16 channels (TanH, ReLU, ELU)		
conv2	65×3	33, 32 channels (TanH, ReLU, ELU)		
conv3	33×2	33, 64 channels (TanH, ReLU, ELU)		
conv4	1×1	33, 65 channels (TanH, ReLU, ELU)		
LSTM1	1×1	250 dims	500 dims	1000 dims
LSTM2				
LSTM3				
fc1	1×1	65 dims fc (TanH, ReLU, ELU)		
concatenate	1×1	65 dims fc1 & 65 dims from previous step		
LSTM4	1×1	250 dims	500 dims	1000 dims
LSTM5				
LSTM6				
fc2	1×1	33, 64 channels (TanH, ReLU, ELU)		

4 Results

We analyzed the performance of the model for motion generation. Each model is trained with a shallow dataset, but the models can generalize the information for the audio input and can keep the motion pattern for trained information.

4.1 Training Process

After training for five epochs, we evaluated the use of different activation function and the performance on the training process. We used the structure described at part 3 Setting a minibatch to 150 sequences and the LSTM

layers to 500 units, we trained models with TanH, ReLU and ELU activation functions at the convolutional section. Figure 5 shows the training loss of each model. We see that despite of training a model from the scratch, the models have a fast convergence on the first iterations. The optimization process can backpropagate the error to the deepest layer; making possible a training without middle states. Regardless the activation function, the training process has the almost same convergence behavior. However, in Table 2 we show that, at the same number of iterations, ELU activation has a lightly lower learning cost compared to ReLU, TanH activations.

4.2 Motion Pattern Generation

For evaluating the performance of each model trained for five epochs, we calculate the Symmetric Mean Absolute Percentage Error (SMAPE) of the dancer's motion pattern, and the motion pattern generated by the models. The SMAPE is defined as:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t| + |F_t|} \tag{8}$$

where F_t is the output pattern from the models, and A_t is the pattern obtain from a real world dancer. Here, we compared that the model can keep and generate the motion pattern using the audio frame information and the previous calculated position. Music pieces used for training have multiple instruments, some pieces contain voices from the singer or the crowd. The dance motion generated has the shape of a wave, thus the movements keeps the beat timing of the music piece and are synchronized with the drum of each music piece. Figure 6 show pattern generated by the models and the comparison with the pattern from a real dancer. Table 3, 4 and 5 show the performance of using different parameters on the training for the motion pattern generation. We compared the motion generated resetting the LSTM each sequence period and following the pattern from the beginning. Also, we compared the evaluation of the processing time for the data forwarding using a GPU GTX Titan X. The default structure was set to ELU activations, with a LSTM of 500 units, and sequence length of 150 steps. In Table 3, we compare the generation error using same training parameters but with different activation functions. We see that ELU activation has a lower error compared to the other functions. However, its processing time is quite higher. Table 4 shows the comparison of using different unit length at the LSTMs layers. The models were trained using ELU activation and the same sequence training number. We see that, using a larger unit length can improve the performance for the motion generation. However, both the time processing and its learning cost are quite higher. Also, using few LSTM units can increase the error of the motion generation. However, the learning cost is lower as its processing time.

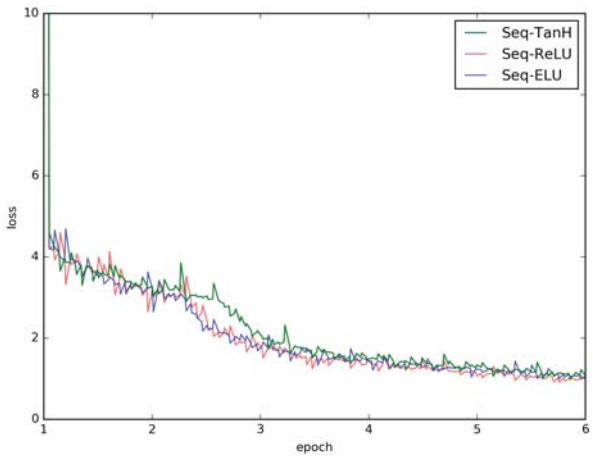


Figure 5: Training graph

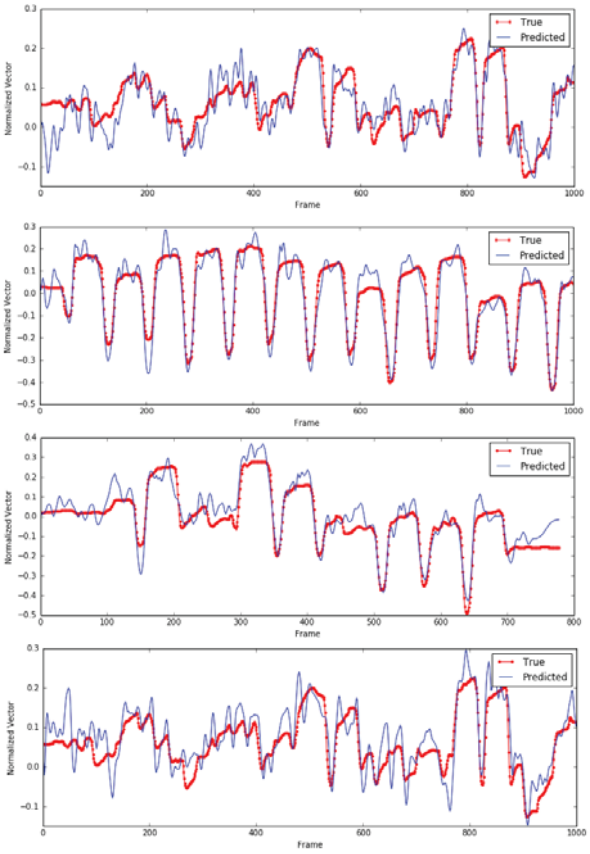


Figure 6: Dance Generated Pattern. Dotted Red: Real World Dancer, Blue; DL Models

Table 5 shows the comparison of using different length sequence on the training. Using the same model, we trained it changing the length sequence used on the training process. We see that the sequence affects the learning cost and the pattern generation error. Training a model with larger sequence improve the performance of the network. Due to the length of the sequence, the learning cost is also affected, increasing the training time. However, due to the models have the same parameters of dimensions and activation function, the processing time for real world application is not affected. Having a larger dataset for the training can lower the

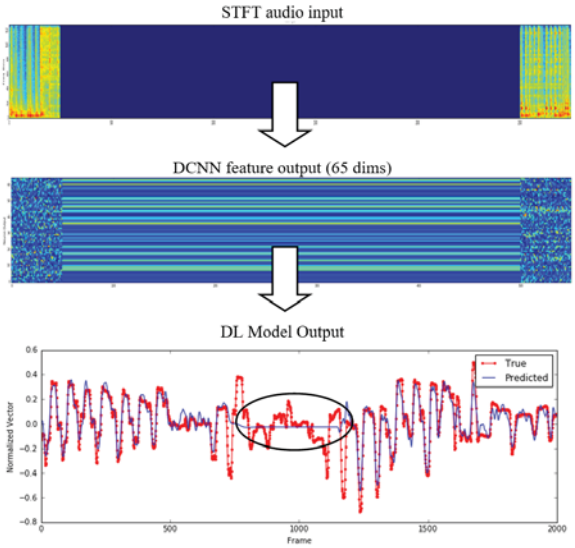


Figure 7: Generated dance pattern at silence input

Table 2: Comparison of Learning cost of activation functions

activation	Training Time (minutes)
TanH	1182.79
ReLU	1152.02
ELU	1139.81

performance of the models. For some tasks it is required larger dataset. However, we see that since the generated pattern is related with the music, a larger dataset with different motion pattern can affect these motion generator models. This becomes a noisy output for the motion (Figure 6).

4.3 Sound dependency

The models generate motion patterns from the previous motion step, but also it is required the sound information. Figure 7 shows the motion generated when the sound is stopped and became silence. We stopped the audio input at the middle of the song, thus the STFT input became blanked. At silence input, the DCNN feature outputs have a same pattern for all the silence frame. This features have been self-learned due to end-to-end training. Therefore, the motion output is stopped. However, after this silence space the sound is restored. The model restarted its pattern generation and generated the pattern from the corresponding audio input frame matching the generated motion with the trained one. Thus, the model can generate pattern from the audio input, but also can continue the generation on non-continues audio inputs.

Table 3: Evaluation of activation functions

Activation	SMAPE (%) without reset	SMAPE (%) with reset	Training Time (mins)	Average Frame Forwarding Time (ms)
TanH	19.73	20.34	1182.79	9.16
ReLU	19.67	20.27	1152.02	8.54
ELU	18.57	19.2	1139.81	10.94

Table 4: LSTM units evaluation

LSTM units	SMAPE (%) without reset	SMAPE (%) with reset	Training Time (mins)	Average Frame Forwarding Time (ms)
250	19.76	20.28	663.99	10.16
500	18.57	19.2	1139.81	10.94
1000	17.81	18.53	1305.85	14.97

5 Conclusion

We proposed a method for motion generation using Deep Sequential Learning which can be trained end-to-end. We showed that the models can generate correlated motion pattern and due to the low forwarding time, they could be used for real robot tasks. The models have a low learning cost and also can be trained from the scratch, avoiding the use of other process such as auto encoding. The proposed model shows reliable performance for motion generation. However, the motion pattern is also affected by the diversity of the training patterns.

References

[Oliveira 2015] J. L. Oliveira, G. Ince, K. Nakamura, K. Nakadai, H. G. Okuno, F. Gouyon, and L. P. Reis, Beat Tracking for Interactive Dancing Robots, Int. J. Humanoid Robot., vol. 12, no. 04, p. 1550023, 2015.

[Itohara 2012] T. Itohara, T. Otsuka, T. Mizumoto, A. Lim, T. Ogata, and H. G. Okuno, A multimodal tempo and beat-tracking system based on audio-visual information from live guitar performances, EURASIP J. Audio, Speech, Music Process., vol. 2012, no. 1, pp. 117, 2012.

[Fukayama 2015] S. Fukayama and M. Goto, Music Content Driven Automated Choreography with Beat-wise Motion Connectivity Constraints, in Proceedings of SMC, 2015.

[Krizhevsky 2012] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., ImageNet Classification with Deep Convolutional Neural Networks, Adv. Neural Inf. Process. Syst. 25, pp. 19, 2012.

Table 5: Sequence length evaluation

Training sequence	SMAPE (%) without reset	SMAPE (%) with reset	Training Time (mins)	Average Frame Forwarding Time (ms)
50	22.99	23.73	247.19	10.94
100	19.16	20.01	774.75	10.94
150	18.57	19.2	1139.81	10.94

[Clevert 2015] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), Under Rev. ICLR2016, no. 1997, pp. 113, 2015.

[Sainath 2015a] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2015-Augus, pp. 45804584, 2015.

[Sutskever 2014] I. Sutskever, O. Vinyals, and Q. V. Le, Sequence to Sequence Learning with Neural Networks, Nips, p. 9, 2014.

[Venugopalan 2015] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, Sequence to Sequence - Video to Text, Proc. IEEE Int. Conf. Comput. Vis., pp. 45344542, 2015.

[Ioffe 2015] S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv, 2015.

[Kingma 2014] D. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, arXiv1412.6980 [cs], pp. 115, 2014.

[Neelakantan 2016] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, Adding Gradient Noise Improves Learning for Very Deep Networks, Iclr, pp. 111, 2016.

[Tokui 2015] S. Tokui, Introduction to Chainer: A Flexible Framework for Deep Learning. 2015.

[He 2015] K. He, X. Zhang, S. Ren, and J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, arXiv Prepr., pp. 111, 2015.

[Graves 2014] A. Graves and N. Jaitly, Towards End-To-End Speech Recognition with Recurrent Neural Networks, JMLR Workshop Conf. Proc., vol. 32, no. 1, pp. 17641772, 2014.

[Cho 2014] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Proc. 2014 Conf. Empir. Methods Nat. Lang. Process., pp. 17241734, 2014.

[Sainath 2015b] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. rahman Mohamed, G. Dahl, and B. Ramabhadran, Deep Convolutional Neural Networks for Large-scale Speech Tasks, Neural Networks, vol. 64, pp. 3948, 2015.

[Abdel-hamid 2012] O. Abdel-hamid, H. Jiang, and G. Penn, Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition, Acoust. Speech Signal Process. (ICASSP), 2012 IEEE Int. Conf., pp. 42774280, 2012.

一般社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-046-09 (11/09)

Using utterance timing to generate gaze pattern*

Jani Even, Carlos Toshinori Ishi, Hiroshi Ishiguro
Hiroshi Ishiguro Laboratories, Advanced Telecommunications Research Institute International, Japan.
even@atr.jp *

Abstract

This paper presents a method for generating the gaze pattern of a robot while it is talking. The goal is to prevent the robot's conversational partner from interrupting the robot at inappropriate moments. The proposed approach has two steps: First, the robot's utterance are split into meaningful parts. Then, for each of these parts, the robot performs or avoids eyes contact with the partner. The generated gaze pattern indicates the conversational partner that the robot has finished talking or not. To measure the efficiency of the approach, we propose to use speech overlap during conversations and average response time. Preliminary results showed that setting a gaze pattern for a robot with a very human-like appearance is not straight forward as we did not find satisfying parameters.

1 INTRODUCTION

During social interaction, the gaze has important regulatory functions [1, 2]. Early work [1, 3] tried to search a systematic relation between gaze and turn-taking. More recent work [4] underlines the collaborative nature of the gaze in turn-taking. The authors in [4] show that "the timing of the listener response is collaborative process, accomplished by joint action". Consequently, a robot holding a conversation with a human should participate in this "collaborative process" in order to have a smooth interaction.

This paper presents an approach to robot gaze control during conversation that take into account this collaborative process. The goal is to make the conversation flow smoother by providing the human with the expected gaze signals that occur during turn-taking.

An expected outcome is to reduce the risk that the listener interrupts the robot. Without signaling, it is quite frequent that the human interrupts the robot because she or he did not understand that the robot intended to continue speaking.

Figure 1: Close-up of Erica.

Using gaze to signal the turn taking is also expected to avoid undesired pauses caused by the listener not taking it's turn fast enough.

In addition to provide the adequate signaling, the gaze pattern should not be unnatural. The implementation of a human-like solution to the problem of unwanted interruption is all the more important as we work with an android robot developed to look very similar to human. This appearance similarity amplifies the sensitivity to unnatural behaviors.

For this reason, the proposed gaze pattern should also display the same habit as human to avert gaze during utterance formulation [5]. Humans tend to avert their gaze during formulation to reduce the cognitive load. This a natural phenomenon we are used to witness during a conversation. Thus, in addition to the turn taking signal, the proposed gaze pattern generation also tries to reproduce the aversion that occurs during formulation.

2 RELATED WORK

The use of gaze by social robot during interaction have been investigated by several authors with different per-

*Research supported by the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project.

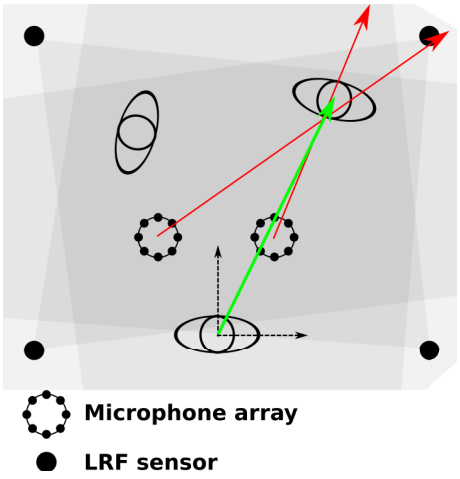


Figure 2: Example of possible sensor network configuration.

spectives. In particular, in [6], the authors recorded human-human conversations in order to estimate statistics from the gaze patterns. Then, these statistics were used to implement the gaze control of a NAO robot. The NOA robot was judged by participants to be more thoughtful and it was able to manage the conversation floor. In this paper, we would like to achieve similar results using a robot that is more human-like than NAO.

3 ROBOT GAZE CONTROL

The proposed gaze pattern generation is designed for Erica [7], a robot that was designed to have a realistic human like appearance, see Fig.1.

The components of Erica that are involved in the gaze control are:

- a sensor network,
- a kinematic model,
- and a closed-loop controller.

The sensor network main role is to track human [8, 9, 10] and determine who is talking [11, 12]. For this purpose a human tracking system is combined with a sound localization system. Figure 2 shows one example of configuration with four laser range finders (LRFs) for tracking humans and two microphone arrays for performing sound localization. During the experiments, the human tracker system was not using LRFs but RGB-D cameras attached to the ceiling of the room [13]. Using the sound localization (the red arrows in Fig.2) it is possible to determine who is talking.

Figure 3 shows the joints involved in the gaze control. The kinematic chain controlling the eyes direction has 7 degrees of freedom (DOF):

- yaw and pitch for the eyes,
- yaw pitch and roll for the neck,
- yaw and pitch for the waist.

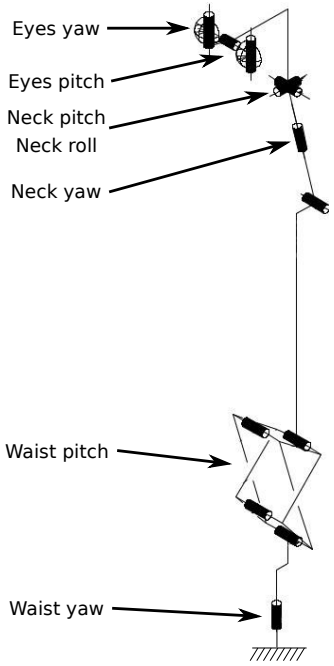


Figure 3: Kinematic chain for the gaze.

However, the current implementation does not use the neck roll.

Pneumatic actuators are used to move the joints. These actuators are controlled by on board PID controllers. The commands are sent to the robot at a frequency of 20 Hz. The robot provides a feedback measured by potentiometers also at the frequency of 20 Hz. The on board PID are tuned to favor smoother movements which results in a lesser control accuracy. Consequently, it is necessary to rely on the feedback to get the achieved positioning.

Using the specifications of Erica, a computer model of the kinematic chain was implemented. The posture of the model is updated when the feedback from the actuators is received. Namely, the model provides an estimate of the current posture of Erica.

The kinematic model provides the current gaze direction of Erica's eyes. The goal of the gaze control is to send command to move the joints of Erica in order to align Erica's gaze direction to the desired gaze direction. Only the eyes are controlled in a closed loop because the accuracy on the eye movement is greater than on the waist and neck.

Erica is able to track a moving person walking in front of her using the gaze control. This is illustrated in Fig.4. The top of Fig.4 shows the yaw of the focus direction (solid line) and the yaw of the gaze direction given by the kinematic model (dashed line). The three other graphs are showing the command values (solid lines) and the potentiometer values (dashed lines) for the control of the waist, neck and eyes yaw. We can note a slight delay, which is expected, and some overshoots. However, the graph for the neck control shows some large errors and the one for the waist some small errors. Then, we can see on the graph for the eyes that the command is different

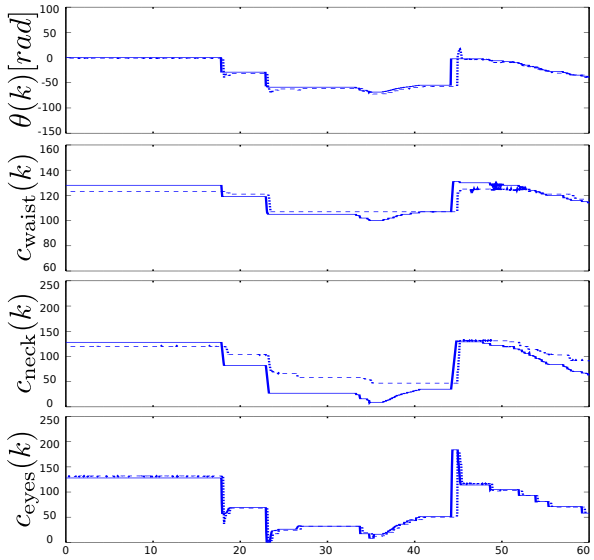


Figure 4: Close-up of the axis command (dashed) and potentiometer feedback (solid) for the yaw.

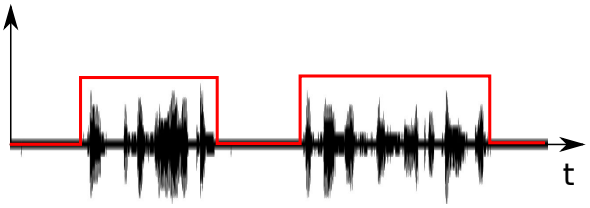


Figure 5: Robot utterance and voice activity (red).

and it compensated for the error as expected.

4 ROBOT GAZE PATTERN

Previous section showed that Erica is able to look relatively precisely to a given direction. In this section, we will discuss the gaze pattern during interaction. In particular, the focus is on the gaze pattern when the robot is talking.

4.1 Gaze pattern timing

Figure 5 shows a typical utterance of the robot. It is possible to have a precise voice activity detection for the robot speech from the text to speech (TTS) module.

The goal is to produce a gaze pattern that presents the adequate cognition and turn taking cues. Such a gaze pattern is illustrated in Fig. 6. The gaze pattern is a succession of gaze aversion and eye contact that have timed in a specific manner.

In particular, if a single utterance is considered, as in Fig. 7, it is split in different phases:

- The cognition phase, denoted by C, starts before the utterance and overlaps the beginning of the utterance. This phase corresponds to the duration during which gaze aversion is expected as the talking would be formulating her or his utterance.
- The final phase, denoted by F, starts before the end of the utterance and persists after the utterance is over. During this phase, the turn taking signal is sent to the conversation partner. If the speaker wants to give the turn, eye contact is sought. But, on the contrary, if the speaker intends to continue talking the gaze is averted.
- The body phase, denoted by B, is the duration between the end of the cognition phase and the beginning of the final phase. During this phase, a succession of short gaze aversions and eye contacts occurs.

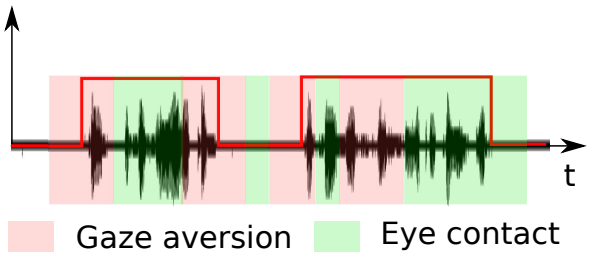


Figure 6: Gaze pattern for two consecutive utterances.

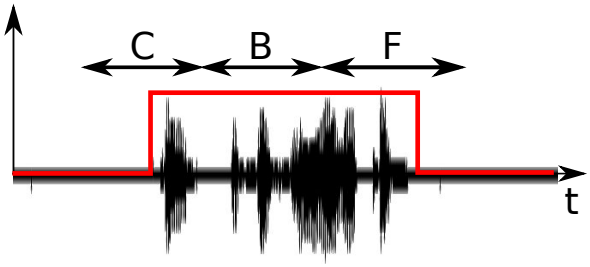


Figure 7: The three utterance phases.

In order to produce the gaze pattern, it is necessary to anticipate the start of the utterance to be able to perform the aversion of the cognitive phase and the end of the utterance to be able to start signaling the turn taking in advance.

The sequence of actions that results in the robot speaking is indicated on the time line of by Fig. 8:

- a speak request is sent at time S to the TTS module,
- the synthesized speech is ready at time T,
- the lip synchronization commands are sent to the robot at time C,
- The lip movement and the sound production start at time V.

The delay d between speech request and actual speech production is due to the necessity to synchronize the speech sound with the lip motion. Because of the pneumatic actuation, the shortest possible delay is 500 ms. The expected end of the utterance is known in advance as the duration D of the utterance is deduced from the synthesized speech. Thus, it is possible to access in advance the voice activity of the robot and all the timing information necessary to create the gaze pattern are readily

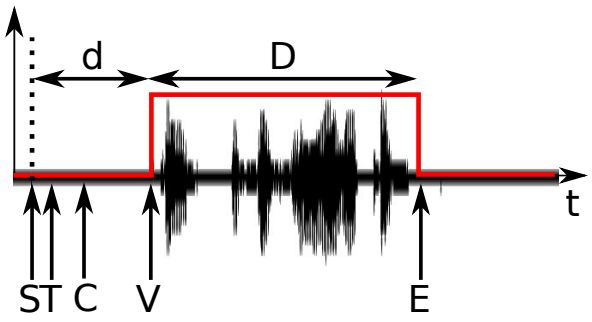


Figure 8: Sequence of events during an utterance production.

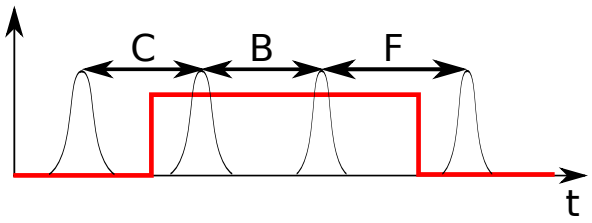


Figure 9: Sampling phase boundaries from Gaussian distributions.

available. Note that to generate a cognition phase longer than 500 ms, it is necessary to add a pre-delay.

In order to have a more natural gaze pattern, it is important to introduce some variability in the different phase durations. In the current implementation, the start time and end time of the different phases are sampled from Gaussian distributions. Figure 9 shows that the phases are determined by four distributions.

In the body phase, the gaze pattern is composed of short aversions and eye contacts. This is done by sampling the aversion duration and the duration between successive aversions from two Gaussian distributions.

Note that when the robot is listening, the gaze pattern is also a succession of aversions and eye contacts. However, the duration of the aversions and the intervals between them are shorter. These patterns are generated this way as humans also tends to do shorter and less frequent aversion when listening than when talking [1].

4.2 Gaze pattern direction

The eye contact is performed by having the robot look at the human that is detected by the sensor network. The sensor network gives the position of the person in the room and the height of that person (the top of the head). In the current implementation, the gaze controller set the robot to look at a fixed offset of 0.15 meters from the top of the head.

During gaze aversion, an offset is added to the gaze controller that results in the eyes of the robot being averted from the person. The gaze aversion offset is characterized by two angles θ in the horizontal plane and ϕ in the vertical plane.

To introduce randomness in the gaze aversion, first the "general direction" is selected among "up", "down",

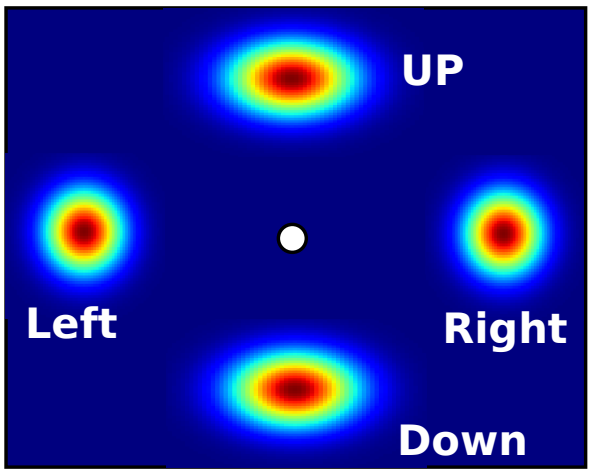


Figure 10: Sampling gaze directions from Gaussian distributions in the four "general directions".

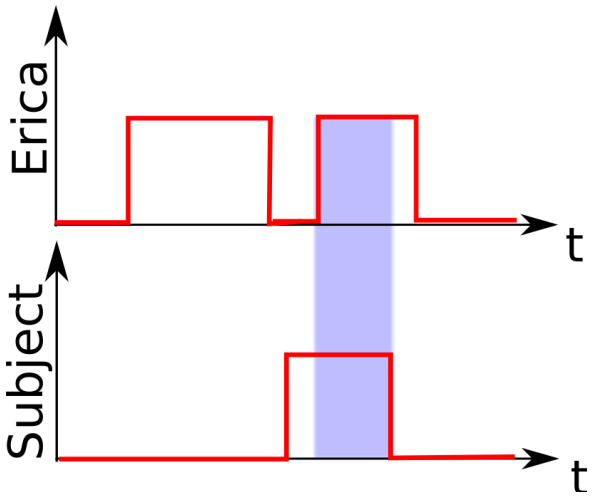


Figure 11: Speech overlap caused by missing the signal indicating a short pause.

"left" or "right". Then, the magnitude of the aversion is sampled from Gaussian distributions, see Fig. 10.

5 EXPERIMENTAL SYSTEM

To asses the effect of the gaze pattern on the conversation, it is necessary to define a measure of performance. For this purpose, a conversation monitoring system is used. Using the microphone arrays of the sensor network, the speech activity of the subject conversing with Erica is logged. The utterance timing and the gaze pattern information are also recorded at the same time.

Figure 11 shows an overlap situation. The subject did not understand that Erica was just making a short pause and took the talking turn. Even if the subject voice is detected it is usually too late to avoid a slight overlap as the robot commands are sent in advance. This is a typical case where the robot should use gaze aversion to signal the turn is not over. Thus, a measure of performance is the duration of the overlap (the blue region in Fig. 11).

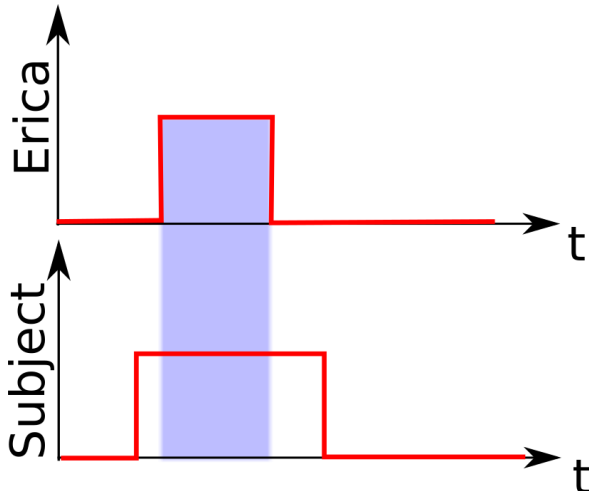


Figure 12: Speech overlap caused by missing the signal indicating the robot is about to talk.

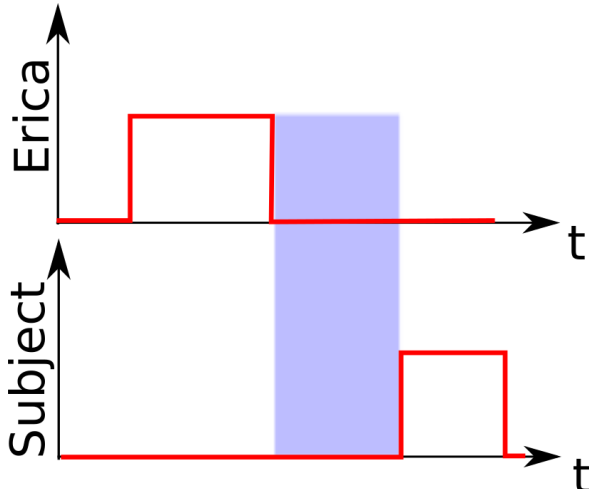


Figure 13: Example of delayed answer.

Another overlap situation is the one when the subject could not anticipate that the robot was about to talk. Then, the subject starts talking just before the robot resulting in an overlap. In this case, illustrated in Fig. 12, not signaling the cognition phase by a gaze aversion resulted in the overlap (in blue).

In addition to speech overlaps, the conversation is not smooth when the subject takes much time to answer to the robot because the robot did not signal properly the end of turn. Thus, measuring the response time of the subject, the blue duration in Fig. 13 is another good indication of performance.

For a given conversation between a subject and the robot, it is possible to calculate the total speech overlap and the average response time. These two values are used as the measure of performance for the generated gaze pattern. Namely, between two candidate gaze patterns the one that results in less speech overlap and a smaller average response time is considered better.

In order to compare different gaze patterns for different subjects, the random generation of the phase timing are only computed one time and stored. Then, it is possible to compare the same realization of the gaze pattern by replaying the stored version to each subjects.

Preliminary experiments have showed that creating the gaze pattern using Gaussian distribution is not straight forward. The first attempts used the means and standard deviations that were estimated from human-human conversation in [6]. However, the generated gaze patterns were not satisfying. In particular, the cognition and final phase tended to be too long for the robot utterances. The reason is maybe that the subjects in [6] were familiar with each others and had conversations composed of rather long utterances. In comparison, the robot-subject utterances tend to be shorter. Another possible explanation is cultural difference as the results reported in [6] are for English whereas we conducted our experiments in Japanese. It is also possible that we are more sensitive to the gaze pattern mismatch as Erica is more human-like than NAO.

6 CONCLUSIONS

In this paper, we motivated the need for a gaze pattern generation that helps the robot to have a smoother conversation. We presented the architecture of the system and explained the concept of the gaze pattern generation. However, the implementation and testing of the system was not done yet as preliminary results showed that finding a reasonable set of parameters for our robot is not as simple as expected. Thus, the focus now is on adapting the gaze pattern statistics to our specific robot in order to proceed with the evaluation. A possibility is to generate the statistics by taking into account the duration of the utterance when creating the gaze pattern.

References

[1] Kendon A., "Some functions of gaze-direction in social interaction," *Acta Psychol.*, vol. 26, no. 1, pp. 22–63, 1967.

[2] M. Argyle and M. Cook, *Gaze and mutual gaze*, Cambridge University Press, 1976.

[3] S. D. Jr. Duncan, "Some signals and rules for taking speaking turns in conversation," *Journal of Personality and Social Psychology*, vol. 23, no. 2, pp. 283–292, 1972.

[4] J. B. Bavelas, L. Coates, and T. Johnson, "Listener responses as a collaborative process: The role of gaze," *Journal of Communication*, vol. 52, no. 3, pp. 566–580, 2002.

[5] A.M. Glenberg, J.L. Schroeder, and D.A. Robertson, "Averting the gaze disengages the environment and facilitates remembering," *Memory and Cognition*, vol. 26, pp. 651–658, 1998.

[6] Sean Andrist, Xiang Zhi Tan, Michael Gleicher, and Bilge Mutlu, "Conversational gaze aversion for humanlike robots," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, 2014, pp. 25–32.

[7] Hiroshi Ishiguro et al., "Erato ishiguro symbiotic human-robot interaction project," `http://www.jst.go.jp/erato/ishiguro/en/index.html`, 2015.

[8] Jae Hoon Lee, T Tsubouchi, K Yamamoto, and S Egawa, "People tracking using a robot in motion with laser range finder," 2006, pp. 2936–2942, Ieee.

[9] D.F. Glas et al., "Laser tracking of human body motion using adaptive shape modeling," *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 602–608, 2007.

[10] L. Spinello and K. O. Arras, "People detection in rgb-d data.," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[11] C.T. Ishi et al., "Evaluation of a music-based real-time sound localization of multiple sound sources in real noisy environments," *Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2027–2032, 2009.

[12] C.T. Ishi, J. Even, and N. Hagita, "Using multiple microphone arrays and reflections for 3d localization of sound sources," *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3937–3942, 2013.

[13] D. Brscic, T. Kanda, T. Ikeda, and T. Miyashita, "Person tracking in large public spaces using 3-d range sensors," *Human-Machine Systems, IEEE Transactions on*, vol. 43, no. 6, pp. 522–534, 2013.

〒162 東京都新宿区津久戸町 4-7 OS ビル 402 号室 03-5261-3401 Fax: 03-5261-3402

本研究会についてのお問い合わせは下記にお願いします.

SIG-AI-Challenges home page (WWW): `http://www.osaka-kyoiku.ac.jp/~challeng/`