修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院　情報理工学研究科　情報・ネットワーク工学専攻　博士前期課程		
氏　　名	竹之内 翔太郎	学籍番号	1731100
論 文 題 目	日常生活音からのリアルタイム ADL 認識方法の研究		

要　旨

人間の行動や心情などを基にして，状況に応じて最適な制御ができるサービスが注目されているが，そのサービスを有用なものにするには，高次情報を得るためにセンサデータから得る低次情報が重要になる．そこで本研究では，ADL(日常生活行動)や心情などの把握を目的として，生活音や非言語音を話声や雑音と識別しながらリアルタイム認識ができるシステムの開発を行った．多種類の非言語音および生活音を対象としてリアルタイム認識を行った先行研究において，使われた認識手法によって，本研究で認識したい音声に対しても使えるかどうかについて検証した．その結果,「話声と非言語音が共存していないこと」や「雑音入力による誤検出対策が行われていない」という課題があり，さらにその手法が話声や非言語音の認識に向いていないという仮説を得た．そこで,「話声や雑音と識別するための手法」や「非言語音認識に適した状態定義手法」に関する既存研究について調査し，リアルタイム認識時の要件についても考慮した上で認識手法を提案した．さらに，提案手法に合った音声認識エンジンを用いてリアルタイム認識の実装を行うことにした．提案手法の認識精度を検証することを目的に，様々な話者や環境下での音声を使って 3 種類の評価を行った．その結果，疑似音素列定義による非言語音同士での分類はそれなりの結果となったものの，連続音声からのリアルタイム認識を想定した処理を含めた場合,「非言語音の検出率」や「雑音入力による非言語音の誤検出」に関して課題が残った．その一方で話声による生活音および非言語音の誤検出は抑えることができたことに加えて，生活音については 1 種類を除いて比較的正確な認識ができていた．また発話中に笑った場合でも，リアルタイム認識時と同様の設定で約 65%の割合で笑いを検出することができたため，連続音声からのリアルタイム笑い声検出には本手法が有効になると考えた．

平成 30 年度修士論文

日常生活音からの
リアルタイムADL認識方法の研究

電気通信大学大学院 情報理工学研究科
情報・ネットワーク工学専攻

学籍番号: 1731100
氏名: 竹之内翔太郎

指導教員　　: 沼尾雅之 教授
副指導教員　: 寺田実 准教授

提出日：平成31年2月27日

概要

人間の行動や心情などを基にして，状況に応じて最適な制御ができるサービスが注目されているが，そのサービスを有用なものにするには，高次情報を得るためにセンサデータから得る低次情報が重要になる．そこで本研究では，ADL(日常生活行動) や心情などの把握を目的として，生活音や非言語音を話声や雑音と識別しながらリアルタイム認識ができるシステムの開発を行った．多種類の非言語音および生活音を対象としてリアルタイム認識を行った先行研究において，使われた認識手法によって，本研究で認識したい音声に対しても使えるかどうかについて検証した．その結果，「話声と非言語音が共存していないこと」や「雑音入力による誤検出対策が行われていない」という課題があり，さらにその手法が話声や非言語音の認識に向いていないという仮説を得た．そこで，「話声や雑音と識別するための手法」や「非言語音認識に適した状態定義手法」に関する既存研究について調査し，リアルタイム認識時の要件についても考慮した上で認識手法を提案した．さらに，提案手法に合った音声認識エンジンを用いてリアルタイム認識の実装を行うことにした．提案手法の認識精度を検証することを目的に，様々な話者や環境下での音声を使って 3 種類の評価を行った．その結果，疑似音素列定義による非言語音同士での分類はそれなりの結果となったものの，連続音声からのリアルタイム認識を想定した処理を含めた場合，「非言語音の検出率」や「雑音入力による非言語音の誤検出」に関して課題が残った．その一方で話声による生活音および非言語音の誤検出は抑えることができたことに加えて，生活音については 1 種類を除いて比較的正確な認識ができていた．また発話中に笑った場合でも，リアルタイム認識時と同様の設定で約 65% の割合で笑いを検出することができたため，連続音声からのリアルタイム笑い声検出には本手法が有効になると考えた．

目次

第 1 章

はじめに

1.1 背景

労働人口減少の解決や快適空間の提供などを目的として，状況に応じて最適な制御ができるサービスが注目されている．そのようなサービスの例として，スマートハウス [1] や感情推定を用いた電子機器制御 [2] などが挙げられる．そのようなサービスでは，最適な制御が行えるような高次情報を得るために，センサデータから得る低次情報が重要になる．特に，センサデータから ADL(Activity of Daily Living; 日常生活行動) や場面状況などを，リアルタイムで認識できるようにすることが重要になると考えられる．

図 1.1 は，Helal らによって提案されたスマートハウスのサービスにおけるミドルウェアの構成図であり，知識層 (Knowledge Layer) におけるセンサデータなどから得た低次情報から，サービス層 (Service Layer) で得た高次情報を経て，アプリケーション層 (Application Layer) にてサービスを提供されるまでの構成が表されている．Helal らは，低次情報を物理層にて得るために「家電利用状況の情報」を挙げているが，近年はより多くの低次情報を得るために，様々なセンサを用いた手法が行われている．

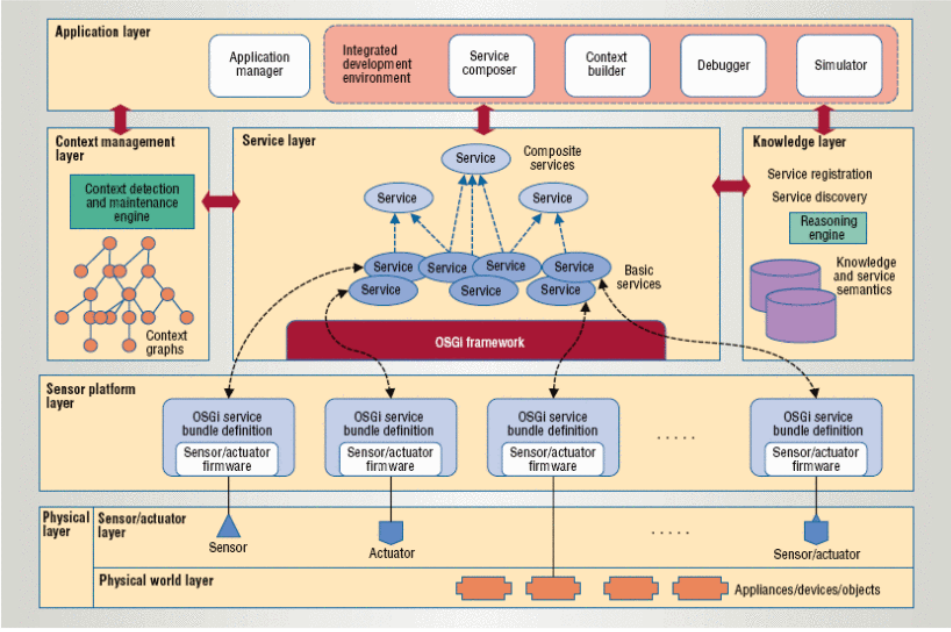

図 1.1 Gator Tech スマートハウスのアーキテクチャ [1]

特に [2] では，音声情報などを用いて人の心情把握に繋げるための枠組みが提唱されており，把握した情報を基に電子機器類を制御することによって，快適空間を創出されるという構成になっている (図 1.2).

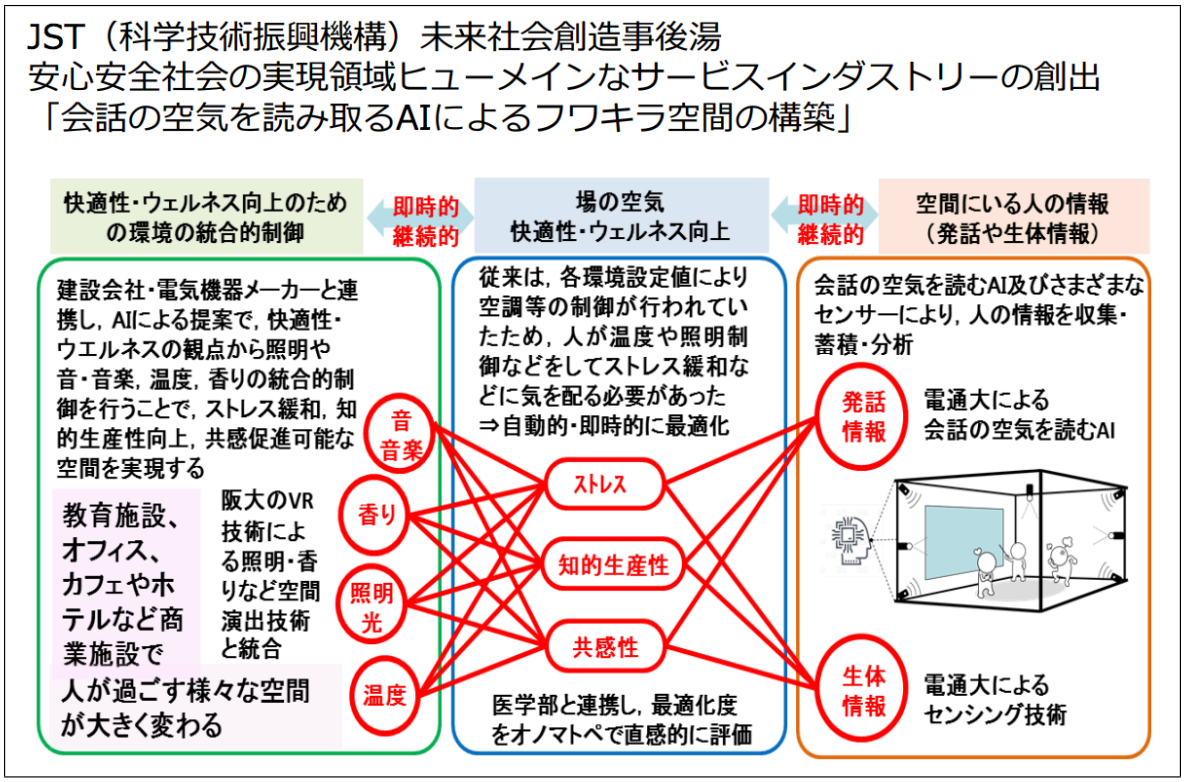

図 1.2　感情推定を用いた電子機器制御サービスの構成 [2]

そこで本研究では，マイクからの音情報を用いて ADL などをリアルタイム認識するためのシステム開発を行う．音情報を使うことにした理由は，ADL の際に生活音 (歯磨きやキータイプ音など) を伴うことが多いことや，非言語音 (咳や笑い声など) に感情的なメッセージを持つことに注目したためである．

1.2 目的

本研究の主たる目的は，多種類の非言語音や生活音に対してリアルタイム認識ができるアプリケーションの開発である．また実用性を考慮した結果，話声や雑音との識別しながら認識を行うことについても認識手法に取り入れることにする．

話声以外の音を多種類で認識するための研究は，主にライフログ作成や高齢者見守りシステム，異常検知などへの応用を目的としてこれまでにいくつか行われている．このような研究におけるタスクは，下図のように分かれる [3]．なお，下図における「特定度」は同じ種類の音同士での類似度の高さを指し，「時間的参照範囲」は音声信号全体の継続時間の長さを指す．また，当頁の底部に下図で使われた用語に対する説明を記載する*1*2*3

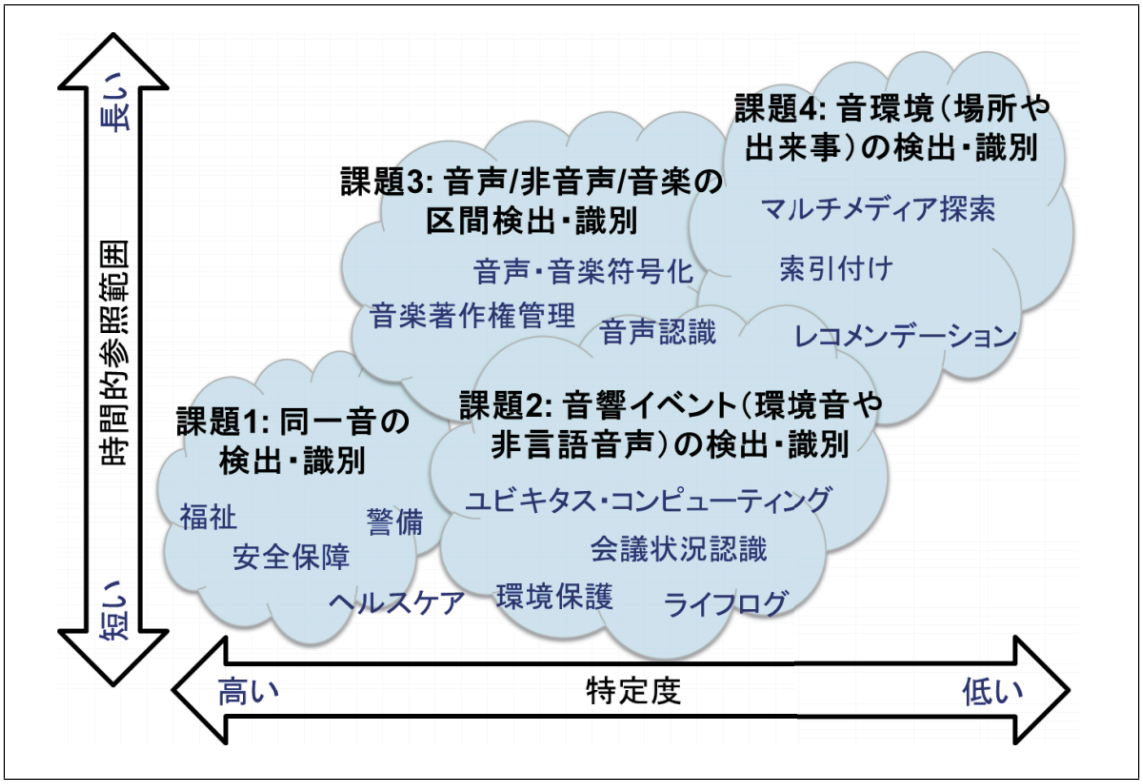

図 1.3 多種類の音響シーン認識における認識対象の分類 [3]

話声以外の多種類の音声に対してリアルタイム認識を行った先行研究における認識対象は，課題 1,4 にあたるものが多かった．そこで，非言語音や生活音 (上図の課題 2,3 に相当) が持つ音響的性質を考慮した上で認識手法の構築を行う．

*1 「同一音」— 警報ブザー音や銃声のように，データベースに集められた全く同一に近い音を指す．
*2 「非言語音声」— 笑い声や咳のような，人から発する音であるが話声ではない音を指す．なお，話声は「言語音」や「音声」と呼ばれる場合もある．
*3 「音環境」— 駅やレストランのように，その場所や出来事においての音を指す．

1.3 本論文の構成

本論文の章構成は，次のとおりである．

第 1 章

本研究の背景および目的について述べる．

第 2 章

提案に向けて重要な要素について踏まえた上で，調査した関連研究ついて述べる．

第 3 章

本研究の認識対象について考慮した上で，提案する認識手法について説明する．

第 4 章

提案手法によって，実際にリアルタイム認識を実装するために行ったことについて説明する．

第 5 章

提案手法に対する評価について述べる．

まず実験の目的と構成について述べてから，実験の結果とそれに対する考察について記す．

最後に提案における各部分での評価について述べる．

第 6 章

本研究のまとめを行い，今後の課題について述べる．

第 2 章

関連研究

本研究での研究目的に向けた認識手法を構築するために重要な要素について検討を行った．

特に，本研究での認識対象となる生活音や非言語音を，話声やノイズと識別して検出することに関して検討した．その結果，マイクからの音声ストリームに対して，次の 3 つをリアルタイム処理でできるようにすることが特に重要になると考えたので，下記に関する既存研究を調査した．

認識手法の構築に向けて重要な要素

1. リアルタイム認識が実際にできること
2. 話声/雑音と識別しながら，生活音や非言語音を検出すること
3. 非言語音の性質を表現した上で状態定義を行うこと

まず最初に，生活音や非言語音など多種類の音声に対する認識手法に関する既存研究を調査した．その中でも特に，リアルタイム認識ができるものについて，認識手法 (モデルや特徴量など) に注目しながら調べた (2.1.1 節)．その結果，認識対象に生活音や環境音が多く含まれていたが，話声と非言語音が共存していなかったため，その認識手法が非言語音認識に適しているのかどうかについて検討すべきだと考えた．そこで，2.1.1 節での認識手法に近いアプローチで，話声や多種類の非言語音の認識を行った研究について調査した (2.1.2 節)．

続いて，リアルタイム処理に適した雑音入力対策を行う手法に関する研究について調査した (2.2 節)．

そして最後に，非言語音認識に適した状態定義手法に関する研究を調べた (2.3 節)．

2.1 生活音・非言語音認識に関する研究

2.1.1 リアルタイム性をもつ認識手法

ウェアラブル端末 (特にスマートフォン) が普及してきたことに伴って，行動認識や異常検知を行うことを目的に，マイクから生活音・環境音のリアルタイム認識アプリケーションの開発に関する研究がいくつか行われている [4][5].

Rossi ら [4] は，生活音や音シーンなどの認識を行うアプリケーション AmbientSense の開発を行った．音響特徴量としては，窓幅 1 秒の音声から MFCC(12 次元) を数十フレーム計算し，各フレーム間での平均および標準偏差を「正味の音響特徴量」として使用している．そして，その特徴量を SVM を用いて分類することで認識が行われるようになっている．計 23 種類の認識対象 (表 2.1) に対する認識性能の評価を行い，5-Fold Cross Validation での評価の結果，全体で 58.45% の認識率を得た．

表 2.1 Rossi らの研究で認識対象となった音 (全 23 種類)[4]

話声	ビーチ	フットボール	髭剃り
皿洗い	シンク	歯磨き	犬の鳴き声
バス	森林	街路	自動車
電話着信音	トイレ水洗音	椅子	駅
掃除機	コーヒーマシン	降雨音	洗濯機
タイピング	レストラン	鳥の鳴き声	-

Pillos ら [5] は，生活音のリアルタイム認識を行うアプリケーションを開発した．[4] とは異なり，常時認識が行われるのではなく，音が出た時のみ認識処理が行われるようになっている．これは，連続音声からの単発音検出の処理が導入されており，音のパワー変化度が閾値を超えた時に有音と判定されるようになっている．この手法に対する認識制度の評価実験を行い，生活音を中心に計 10 種類 (表 2.2) に対する認識精度の評価を行った．計 5 種類のモデルを使って比較しながら，5-Fold Cross Validation で検証した結果，MLP(Multi Layer Perceptron; 多層パーセプトロン) を使った時に全体で 74.5% の認識率を得た．

表 2.2 Pillos らの研究で認識対象となった音 (全 10 種類)[5]

くしゃみ	赤ちゃんの泣き声	点火音	降雨音	波の音
犬の鳴き声	鶏の鳴き声	時計の音	ヘリコプター	チェーンソー

2.1.2 認識対象に生活音・非言語音が多く含まれた研究

[4][5] でも用いられた SVM を認識器として，話声や非言語音のような「人が発する音」を対象に，認識を行った研究は柴田ら [6] が行った．話声や非言語音 (咳音・笑い声) と環境音の計 4 種類に対して，SVM を用いて認識が行われるようになっているが，その際に非言語音認識に適した音響特徴量に関して検討がされている．評価の結果，様々な音響特徴量を用いたものの全般的に咳音の認識精度が悪く，最良の時でも適合率 0.01, 再現率 0.05 となった．著者は結論で，GMM(Gaussian Mixture Model; ガウス混合分布モデル) でのフレームベース識別器の使用や，HMM(Hidden Markov Model; 隠れマルコフモデル) のような音の時間的変化を表現するモデルの使用を検討したいとの記述があった．

非言語音や生活音が認識対象に多く含まれた研究が，Shaukat ら [7] らの研究がある．モデルとしてアンサンブル分類器を使用し，音響特徴量としては MFCC と LPC などを組み合わせてフレーム間での平均と標準偏差を計算したものが使用している．非言語や生活音を中心とした計 18 種類の音声に対する認識精度の評価を行った結果，50-50 オープンテストを行った時に 77.9% の認識率となり，同じデータセットを用いた先行研究 [8] よりも良い結果を得ることができた．ただし，認識対象に多くの非言語が含まれていたものの，認識対象に話声が含まれていなかった．また，音声区間の始点・終点が既知である上で特徴量を計算する必要があるため，連続音声から正確に認識が行われるかについての検討はなされていなかった．

表 2.3 Shaukat らの研究で認識対象となった音 (全 18 種類)[7]

呼吸音	咳	皿洗い
ドアを閉じる音	ドアを開ける音	髭剃り
泣き声	叫び声 (男性)	叫び声 (女性)
ドライヤー	拍手	タイピング
笑い声	紙をめくる音	ガラスの割れる音
くしゃみ	水の音	あくび

2.2 雑音入力に対する対策の研究

多種類の音響イベント認識を行いたい場合，認識対象の音が出た際に正しく検出することができて，尚かつ未検出や誤検出を防ぐことは最も重要なことになる．それに加えて，雑音 (呼気ノイズなど) が入力された場合でも誤検出されないことも重要である．なぜなら，実際に認識システムを使いたい場合，マイクに雑音が入ってくる可能性があるためである．そこで雑音入力による誤検出対策手法に対する研究を調査することにした．

本研究で認識対象としている，多種類の生活音や非言語音をノイズと識別して認識するための先行研究は殆ど無かった[*1]ため，従来の話声用音声認識における雑音棄却手法について調査した．

話声と話声以外が含まれる音声の中から話声区間を検出する，VAD(Voice Activity Detection; 音声区間検出) の手法について，石塚らがまとめている [9]．VAD には色々な手法が存在するが，バッチではなくリアルタイム処理をしたい場合，計算量との兼ね合いで手法が限られている．計算量が少ない VAD の手法としては，振幅やゼロ交差数を用いた手法や，GMM による尤度計算を用いることが多い．

GMM を用いた VAD の研究として，Lee ら [10] は雑音にロバストな音声認識システムの開発を目的に，話声をノイズや非言語音と分類するための研究を行った．公共施設に長期間集音した音声に対してラベル付けを行った結果，ノイズが入力された頻度は多いことが分かった．そこで，話声に加えてノイズや非言語音 (咳・笑い声) に対する計 5 クラスの GMM を学習させた上で評価を行った．その結果，GMM は発話内容の違いによる影響を受けにくい (text-independent) という利点があることが分かった．なお，この音声認識システムで雑音棄却が行われている様子は web 上に掲載されている[*2]．

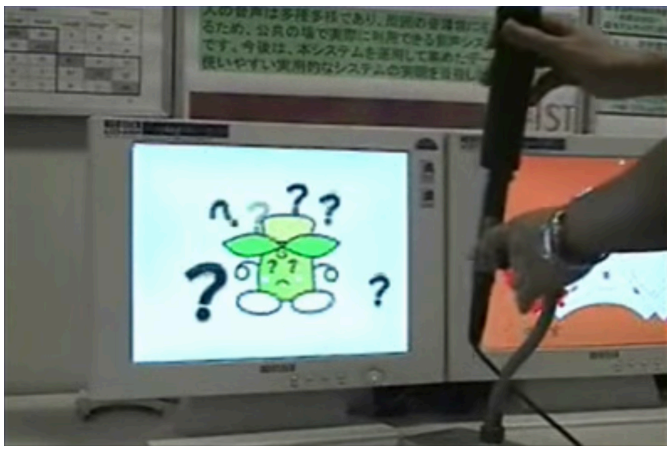

図 2.1 Lee らが開発した雑音に頑健な音声認識システム

[*1] [6] では，話声と非言語音 (咳・笑い声) を環境音と識別する形で認識を行っていた

[*2] https://www.youtube.com/watch?v=3-JWN6ZPezA

2.3 特定の非言語音認識を主目的とした研究

この節では，特定の非言語音に対する認識を主目的とした研究について触れる．

2.3.1 話声認識用音素モデルを用いた非言語音認識手法

高橋ら [11] は，咳に対するモデルそのものを定義するのではなく，咳に対する疑似音素列を定義した上で，各音素に対する既存の話声認識用音響モデルを用いて咳音検出を行う手法を提案した．

咳音に対する疑似音素列を定義することで，既存の話声認識用音響モデルを用いて咳音検出が行われるようになっている．なお，リアルタイム音声認識エンジンである Julius で実装されたためリアルタイムでの咳音検出が可能である．

評価を行った結果，話声と識別する形で咳音を検出するには，咳に対する音響モデルの学習したものを使用するよりも，咳音に対する疑似音素列を使用した方が認識率が良好になることが分かった．

2.3.2 誤検出防止に重点がおかれた咳音検出手法

Drugman ら [12] は正確な咳音検出を目的として，接触型マイクを気管と胸郭に 2 つ装着してそれぞれに対する音響モデルを用意する形で，咳音の検出を行う手法を提案した．

多くの被験者が発した咳音を音声データとして使用して評価をした結果，咳音の検出率は 94.6% と優れた結果を得た．また評価時には，咳の検出率についてだけでなく，咳以外の音による誤検出についての検証がされており，咳と同じく強い呼気を伴う音声 (笑い声や呼気ノイズなど) による誤検出について評価した結果，呼気ノイズによる誤検出は殆ど無かったものの，笑い声入力による咳の誤検出率は 15.6% となった．

2.3.3 疑似音素モデルを用いた笑い声分類手法

笑い声の感情的分類を音声から行うことを目的として，大原ら [13] は笑い声に対する疑似音素定義を用いた笑い声分類手法を提案した．

笑い声の音声から，計 4 種類の疑似音素を定義してから手動でラベリングを行い，各疑似音素に対して学習させた HMM から笑い声の分類を行った．

計 4 種類の笑い声 (大爆笑, 普通の快の笑い, 不快の笑い, 社交笑い) に対する分類精度の評価を行った結果，全体で 85% の認識率を得た．

第 3 章

提案

3.1 認識手法提案に向けた課題

第 2 章の冒頭で述べた，「認識手法の構築に重要な要素」に向けて課題になる箇所とそれに対する解決策について，先行研究調査を経て考えたことについて述べる．

実際にリアルタイム認識ができるようにすること

まず本研究で認識対象となる，生活音や非言語音の音響的性質について考慮した結果，無音時には認識を行うべきではないと考えた．[4] のように音シーン (駅や森林など) が認識対象に含まれていれば，音量が小さい時でも認識結果を出す必要があると考えられる．しかし，生活音や非言語音や話声が「音量が大きい音」を伴っていることを踏まえると，無音時には認識処理を行うべきでないと考えた．無音時に認識処理をすべきでないもう一つの理由として，認識結果導出時に要する音響特徴量の計算コストが大きいことである．特にリアルタイム認識時においては，特徴量計算を常時行うのは計算資源的にも望ましくないといえる．

そこで [5] のように，まずは「有音の検出」を行ってから初めて特徴量計算を行う形で認識結果を得るようにした方が良いと考えた．「有音の検出」を少ない計算コストで行いたいため，[9] で紹介された，振幅などを用いた方法を使うべきだと考えた．計算コストを抑えて「有音の検出」を行った方が良いと考えた．

リアルタイム認識時に使用するモデルについて検討を行った．[6] の評価結果から，[4][5] で使われた SVM や MLP は，非言語音や話声のような「人から出る音の認識」には向いていないという仮説を得た．そこで，[6] が結論で言及していた，HMM などの音響信号の時間的変化を考慮した識別器を非言語音認識にも使うことを検討した．その理由は 2 つある．

1. HMM を用いたリアルタイム認識は，話声用音声認識システムで既に使われているため
2. 話声用音声認識システムの音響モデルは，声の音響的特徴を表現したモデルであることから，「人から発する音」である非言語音認識にも応用できることを期待したため

話声/雑音と識別しながら，生活音や非言語音を検出すること

話声は発話内容の違いから音響的特徴が幅広いため，発話内容による影響を受けにくい形 (*text-independent*) で，生活音などと区別するための処理が必要と考えた．リアルタイム認識時ではこの処理を計算コストを抑えて行う必要があるが，その場合は GMM による尤度情報を用いることで，話声や雑音などとの識別を *text-independent* で行えると考えた [9]．

Lee ら [10] の研究で，話声やノイズの他に非言語音 (咳・笑い声) に対する GMM のクラスを用意することで良好な結果を得ていたことから，この他に生活音などの GMM を追加することによって，非言語音や生活音を話声・雑音と識別しやすくなると期待した．

非言語音に対する状態定義を適切に行うこと

非言語音に対するモデルを用いて認識を行いたい場合，次のような課題が存在する．

1. 音響モデルによる音響的特徴の表現が難しいこと
2. 手動ラベリング時の位置決めやラベル定義が属人的になりやすいこと

例えば「笑い声」に対する音響モデルを HMM で定義したい場合，笑い声の長さ (音節数) が不定になりやすいため，適切な状態数に対する検討が難しくなるという課題がある．また，単に笑い声に対するモデルを定義するのではなく，[13] のように笑い声に対して疑似音素モデルを新たに生成して認識を行いたい場合，ラベルの位置決めなどが難しく属人的になりがちで再現性に関する課題があると考えられる．

上記のモデル定義時における課題への対策となるように，非言語音の認識手法を考えることにした．そこで，オノマトペ指向の音声認識器を作ることで非言語音認識を行うことを考えた．またその際に，話声認識で使われている音響モデルを利用できるのではないかと考えた．これは，非言語音がオノマトペでよく表現されることに加えて，非言語音が話声と同様に「人から発する音」であることに注目したためである．

そこで，咳音だけでなく他の非言語音 (笑い声など) の認識にも，[11] と同様に話声用音素を用いた疑似音素列定義が使えれば，再現性の良い手法になると考えた．

以上より，認識手法構築時の課題に対する解決策についてまとめた結果，次のことを提案に導入することを考えた．

- 無音時に認識処理を行わず，振幅などを基準に有音を検出してから認識処理を始める
- GMM による尤度情報によって，話声や雑音と識別しながら生活音・非言語音の認識を行う
- 話声認識用の音素に対する音響モデルを用いて，非言語音認識を行う
  - その前に非言語音の区間を検出する必要があるので，GMM の尤度情報から非言語音区間を検出する
  - 各種類の非言語音に対する，疑似音素列定義が必要

3.2 提案手法の方式

非言語音や生活音など多種類の音声を，話声やノイズと識別しながら認識を行うための，リアルタイム認識手法を提案する．

3.1 節での検討を経て，マイクからの音声ストリームから認識結果が導出されるまでの流れを，以下のようにすることにした．

1. 音声ストリームから，「有音区間」を検出
   - 古典的な VAD 手法である，振幅などを用いた手法で行う
   - これにより，無音時に認識処理が行われないようになる
2. 検出された有音区間には未だどの種類の音声が含まれるのかは分からないため，GMM で各クラスに対する尤度を計算
   - ここで初めて音響特徴量を計算
   - 使用する GMM のクラスは，生活音・非言語音に加えて話声やノイズについても追加
3. 非言語音に属するクラスが最尤となった場合，非言語音の音声区間を検出
   - 非言語音認識に疑似音素定義を使用する場合，その前段階として非言語音声区間の始点/終点を出す必要があるため
   - 区間の長さが一定以下であった場合，誤検出と判断して棄却
4. 非言語音の音声区間が有効であった場合，話声認識用音素モデルを用いて音素列観測を行う
   - 咳や笑い声など，非言語音に対する疑似音素列を予め定義しておく
   - 非言語音認識用の疑似音素モデルを新たに学習したものを用いる場合，ラベルの位置決めなどが難しく属人的になりやすいという問題点があると考えられる．そこで，話声認識で使われている既存の音響モデルを使用することで，この問題の解決を試みる

3.3 提案手法の構成

3.2 節での手法で，音声ストリームから認識結果が出力されるまでの流れは図 3.1 のようになる．

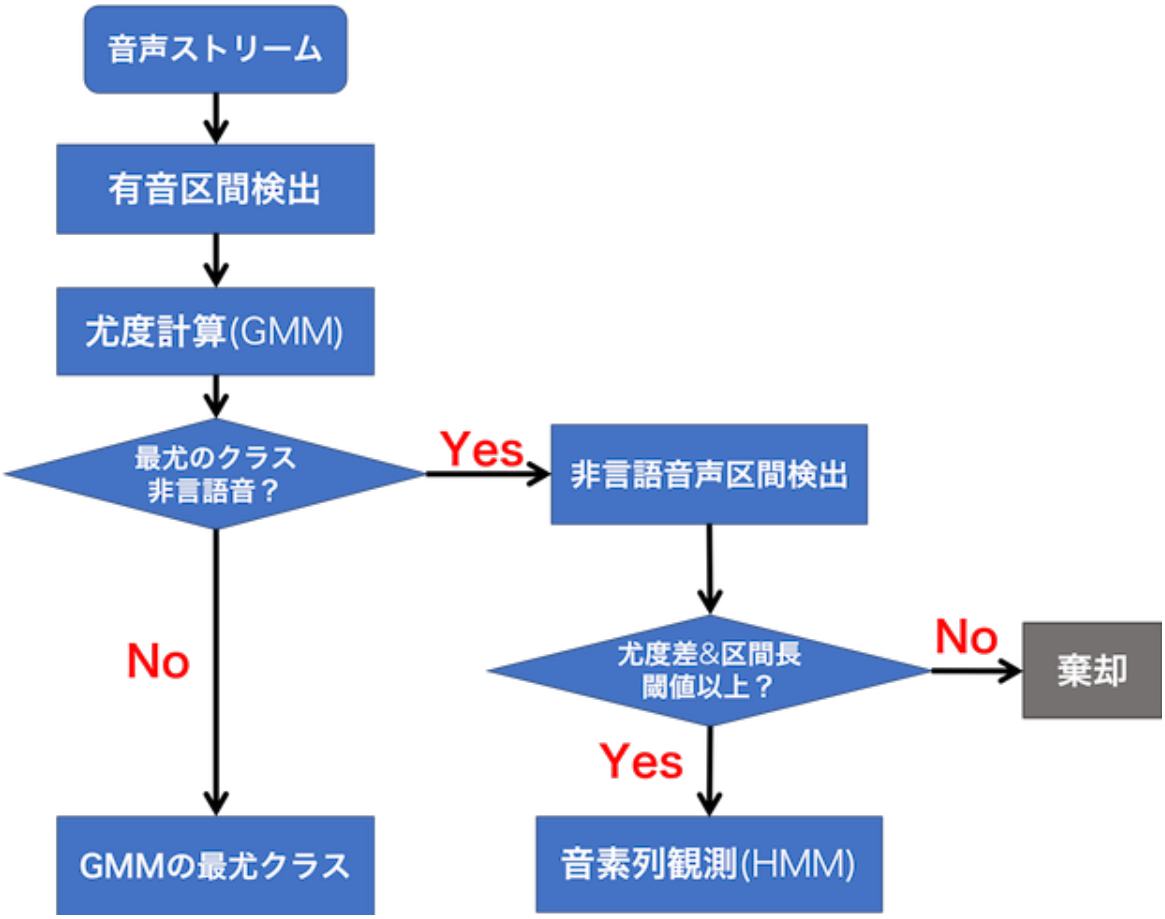

図 3.1 音声入力から認識結果出力までの流れ

次頁では，上図における各部分での処理について説明する．

有音区間検出

まずは単純に，音声ストリームから「音がある区間」を検出してから，以後の認識処理を行うようにした．

検出にあたり，古典的な VAD 手法 [9] である振幅やゼロ交差数などが閾値を上回った区間を検出する手法を取ることにした．

尤度計算 (GMM)

検出された有音区間に対して音響特徴量を計算し，GMM から各クラスに対する尤度をフレーム毎に計算することによって，有音区間がどんな種類の音から構成されているのかを導出する．その際，非言語音の区間が最尤であった場合は，後続の「非言語音区間検出」の処理を行うことにする．そうでない場合 (最尤クラスが生活音などの場合) は，GMM の最尤クラスを認識結果として認識処理を終了する．

話声と識別する形で認識できるようにしたいので，生活音や非言語に対する GMM だけでなく，話声やノイズについても GMM を用意することにした．

非言語音声区間検出

尤度計算で非言語音に属するクラスが最尤になった時，非言語音区間の検出を行う．方針としては，話声用音声認識システムで話声区間検出に用いる VAD[9] を，非言語音の区間検出をするような形式をとった．

具体的には，非言語音とそれ以外 (生活音や話声など) との尤度差を計算し，尤度差が閾値を上回ってから下回るまでの区間を非言語音の音声区間とするようにした．また，その際に検出された区間が極端に短い場合は誤検出されたとみなして棄却するようにした．

音素列観測 (HMM)

非言語音区間が検出されたら，その区間に対して音素列を観測することで非言語音同士の分類を行う．音素列観測時に使用する音響モデルは，[11] と同様に話声用の音声認識で使われるモデルを使うことにした．

具体的な処理としては，各種類の非言語音に対する疑似音素列を定義し，その中から最尤となった疑似音素列を求めることで，非言語音同士を分類するようになっている．

3.4 非言語音に対する疑似音素列定義について

図 3.1 における「音素列観測 (HMM)」に関する定義について述べる．

咳や笑い声など様々な非言語音に対し，[11] と同様のアプローチで疑似音素列を辞書定義に行った上で，オノマトペ指向の非言語音認識を行う．

話声認識用音素を用いて，非言語音に対する疑似音素列を定義しておく必要があるが，咳に対しては [11] で行われていたものの，咳以外の非言語音 (笑い声など) に対する同様のアプローチで認識を行った研究は今まで無かった．そこで，非言語音に対して音響的分析がされた既存研究を調査した上で，疑似音素列定義を行うことにした．

本研究では，「笑い声」「咳」「いびき」 の計 3 種類の非言語音を認識対象とした．この 3 種類にした理由は，発生頻度が高いと考えられることに加えて，音響的な分析を行った既存研究があること，また評価実験時に使う音声データを入手できたためである．

オノマトペ指向の非言語音認識を行うにあたって，非言語音に対する疑似音素列を定義しておく必要がある．疑似音素列定義を利用して非言語音認識を行った研究は，咳音に対しては高橋ら [11] が行っていたが，笑い声やいびきに対しては同様のアプローチから認識が行われた既存研究は無かった．そこで，咳やいびきに対して音響的分析に関する既存研究を調査し，それを基に疑似音素列を定義して非言語音認識を行うことにした．

咳に対する疑似音素列定義

咳音に対しては，高橋ら [11] が定義した音素列定義 (表 3.1) を参考にすることにした．その理由は，本研究で使用する音響モデルが全く同じものであるためである．

表 3.1　高橋らが定義した咳音に対する疑似音素列定義 [11]

/f/-/u/-/q/
/u/-/q/
/z/-/u/-/q/
/u/-/f/-/u/-/q/

なお，上表での音素/q/は促音[*1]に相当する音素である．

笑い声に対する疑似音素列定義

笑い声に対する音響的分析に関する既存研究調査を経て，定義することにした．

笑い声の音声合成手法について，Urbain らが [14] 提案しており，笑い声は「子音/h/に近い音と母音 1 つの繰り返し」から構成されていることに注目していた．

そこで，次のような疑似音素列定義を行うことにした

- 子音/h/ + 母音 1 つ (+ 促音/q/) を 2 回以上繰り返す

いびきに対する疑似音素列定義

笑い声と同様に，いびきに対する音響的分析に関する既存研究調査を経て，定義することにした．

寺井ら [15] は，いびきに対する音声波形の分析を経て，/ウ/または/オ/の母音性が見受けられることを発見したことに加えて，聞き取った時の擬音語表現が「グー」「ガァー」など長音で終わる頻度が非常に高かった．この 2 つに注目し，いびきに対して長母音/u:/または/o:/で表現することにした．

[*1] つまる音を表す．日本語では「っ」「ッ」で表記される．

第 4 章

実装

実際にリアルタイム認識を行うために実装したことについて，本章で述べる．

4.1 実装の構成

提案手法 (図 3.1) に合った音声認識アプリケーションを調査した結果，Julius[16][17] を用いて実装を行うことにした．

Julius には以下のような機能が実装されている．この機能を利用して，認識対象を生活音や非言語音などに拡張する方針で実装することにした．

- 自前で学習したモデル (HMM や GMM) を使うことができること
  - HTK[18] フォーマットのモデルをインポートすることが可能
- VAD が実装されており，その際のパラメータ設定に対する自由度が高いこと
- 音響モデルだけでなく，言語モデル (認識時の文法定義など) の定義を自由に行えること

次頁の図 4.1 は，Julius での提案手法の認識における構成図である．

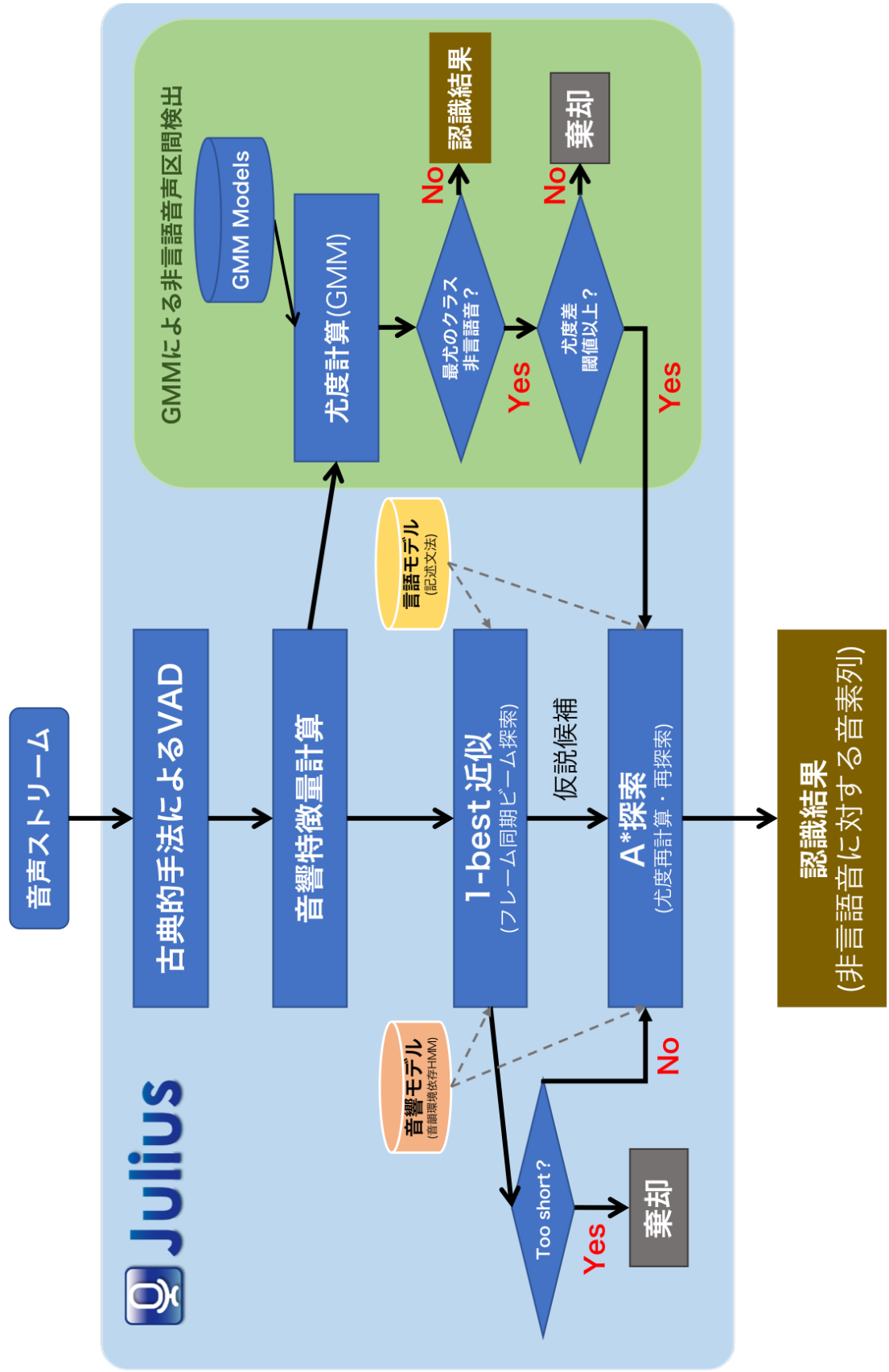

図 4.1 Julius を用いた提案手法の実装構成

4.1.1 提案手法との対比

Julius で認識結果が導出されるまでの流れ (図 4.1) について，提案手法の構成図 (図 3.1) と対応させながら述べる．

1. まずは「古典的手法による VAD」として，振幅やゼロ交差数を基準に
   「音の有る区間」を検出後，MFCC ベースの音響特徴量を計算
   - 提案における「有音区間検出」に対応
2. 有音区間に対して，GMM による非言語音区間検出処理 (図 4.1 の緑枠部) を行う
   - GMM の各クラスに対する尤度をフレーム毎に計算する．最尤のクラスが非言語音かどうかで，以後の処理が変わる (提案の「最尤のクラス非言語音？」に対応)
   - 最尤クラスが非言語音ではなく，**生活音や話声などの場合**はこの結果が認識結果となり，認識処理を終了 (提案の「GMM の最尤クラス」に対応)
   - 最尤クラスが**非言語音の場合**は，非言語音/非言語音以外との尤度差を計算して，非言語音声区間を受理するかどうかを判定認識処理を終了
     (提案の「**尤度差**&区間長 閾値以上？」に対応)
3. 非言語音声区間が受理された場合，「非言語音に対する認識計算 (1-best, A*探索)」を行う
   - 音響モデル/言語モデルを用いて，最尤の音素列を求める
     (提案の「音素列観測 (HMM)」に対応)
   - その際に区間長が閾値に満たないものは，誤検出とみなして棄却する
     (提案の「尤度差&**区間長** 閾値以上？」に対応)
   - 観測された音素列が，非言語音に対する認識結果となる

4.1.2 話声認識時との違いについて

本来は話声認識用として使用されている Julius を，非言語音や生活音などの認識に向けて行ったことについて本節で述べる．

下の図 4.2 は，話声認識時における Julius での処理の構成図である．

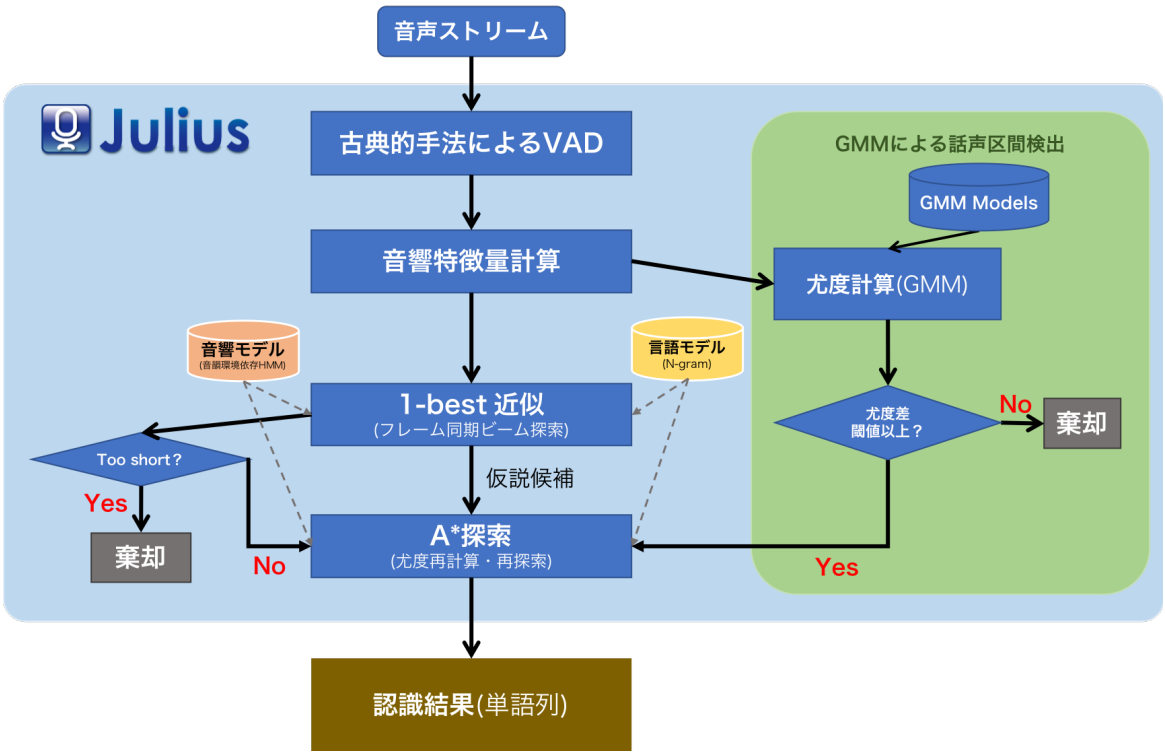

図 4.2 Julius での話声認識時の構成

本研究での提案手法 (図 4.1) と話声認識時 (図 4.2) との違いについて述べる．

- 話声認識時では話声区間が検出される際の処理 (図 4.2 の緑枠部) を変更し，提案手法では非言語音音声の区間検出を行う
- 話声認識時では GMM の最尤クラスによる情報が利用されない (話声のクラスが最尤であるかどうかしか注目しない) ことが多いが，提案手法では最尤クラスの情報を利用して生活音などを検出するようにしている (図 4.1 の緑枠部の「認識結果」が該当)
- 言語モデルの方式が異なる (話声認識時は N-gram，提案手法では記述文法)

4.2 音声認識エンジン Julius

この節では，リアルタイム認識の実装するために使用した Julius の仕様などについて述べる．

4.2.1 Julius の実行に必要な定義について

Julius で音声認識を行うためには，以下の定義が必要である．

- 言語モデル (Language Model)
- 非言語音声区間検出用 GMM
- 音響モデル (Acoustic Model)
- config 定義

以下では，各種の定義について触れていくことにする．

言語モデル (Language Model) [*1]

言語モデルは，単語間での接続関係決定するためのモデルである．Julius で使用できる言語モデルは計 3 種類[*2]あるが，非言語音認識時に辞書定義する語彙は多くないため，本研究では記述文法を使用することにした．

記述文法は，想定される単語 (音素列) のパターンを形式言語の形で定義するものである．与えられた文法上のパターン内で，最尤の候補を選出する形で認識結果が出力されるようになっている．なお，記述文法は 2 つのファイルを使用する．

- 文法定義ファイル (grammar ファイル)
  - 単語カテゴリ間での接続に関する制約を定義
  - BNF 形式で記述
- 辞書定義ファイル (voca ファイル)
  - 各単語カテゴリごとに，単語表記と音素列 (読み) を登録

記述文法と辞書定義のファイルを作成した後，mkdfa.pl [*3] コマンドを実行して DFA(Deterministic Finite Automaton; 有限オートマトン) に変換することによって認識時に使用することが可能になる．

本研究で非言語音認識を行うために記述文法に定義した内容については，4.5 節で触れる．

[*1] https://julius.osdn.jp/juliusbook/ja/desc_lm.html
[*2] 記述文法, 孤立単語認識, 単語 N-gram
[*3] https://julius.osdn.jp/juliusbook/ja/mkdfa_pl.html

非言語音区間検出用 GMM

GMM による尤度情報から非言語音区間検出を行うためのモデルである．

HTK フォーマットのモデルをバイナリファイルに変換することで，自分で学習させた GMM を Julius にインポートすることが可能になる．詳しくは 4.4 節にて説明する．

本研究における認識対象より，生活音や非言語音に加えて，話声やノイズに対する GMM を用いることにした．

音響モデル (Acoustic Model) [*4]

音素ごとの音声波形パターンのモデルである．

本研究では，非言語音に対する音素列観測を行う際に使用する．モデルについては，Julius に付属する日本語話声認識用の HMM[*5] を使用した．このモデルは約 60 時間にわたる大量の音声データ [19] から学習されている．

なお音響特徴量は，MFCC ベースであれば非言語音区間検出用 GMM と異なるものを使用可能である．

config **定義** (.jconf **ファイル**)

認識時のパラメータや引数に対する設定を行う．

提案で認識を行うために，調整を検討したパラメータについては 4.2.2 節で触れる．

Julius ではマイク入力からのリアルタイム認識だけでなく，音声ファイルからの認識も可能である．その際に，2 つの引数 (`"-realtime","-cutsilence"`) を指定することによって，音声ファイルからでもマイク入力からのリアルタイム認識時と同じ処理 (有音区間検出など) を行うことができる．

[*4] `https://julius.osdn.jp/juliusbook/ja/desc_am.html`

[*5] `https://github.com/julius-speech/dictation-kit/blob/master/model/phone_m/jnas-mono-16mix-gid.binhmm`

4.2.2 認識時のパラメータなどに対する設定について

この節では，音響モデルなどに対する設定や，Julius での認識時にチューニングを検討したパラメータなどについて述べる．

まず，音響特徴量や音響モデル (HMM)，非言語音区間検出用の GMM などに対する仕様は，表 4.1 のとおりである．

表 4.1　音響特徴量および音響モデルに対する設定

サンプリングレート	16 kHz
ウィンドウ幅	25 ms(フレームシフトは 10 ms)
音響特徴量 (GMM)	MFCC(0〜12)+ΔMFCC(0〜12)+ΔΔMFCC(0〜12)
音響特徴量 (HMM)	MFCC(1〜12)+Power(1)+ ΔMFCC(1〜12)+ΔPower(1)+ ΔΔMFCC(1〜12)+ΔΔPower(1) (CMN)
GMM の混合数	32
日本語用音素に対する HMM	5 状態 16 混合 Monophone 音響モデル

続いて，本来は話声認識エンジンである Julius で非言語音や生活音認識を行うために，調整したパラメータについては下記のとおりである．提案の構成図 (図 3.1) と対応づけながら説明する．

表 4.2　Julius で調整したパラメータ

提案手法における該当箇所	引数名
有音区間検出	`"-lv"` (振幅) `"-zc"` (ゼロ交差数)
最尤のクラス非言語音?	`"-gmmreject"` (非言語以外のクラスをここに指定した)
尤度差&区間長閾値以上?	`"-gmmup"` (非言語音声区間開始となる閾値) `"-gmmdown"` (非言語音声区間終了となる閾値) `"-rejectshort"` (検出された区間長が閾値以下の入力を棄却)

4.2.3 音声ファイル入力による認識結果について

評価実験 (第 5 章) では，音声ファイル入力による認識結果から評価を行っている．実験においては，マイクからのリアルタイム認識時と同じ処理を，音声ファイル入力時でも行うようにすることを基本としているため，その時の認識結果に関する仕様について本節で説明する．

下の図 4.3 は，咳音の音声データを入力した時の出力結果である．先頭から 5 行は入力された音声データのファイル名や形式などを表し，認識結果は 6 行目以降に出力されている．下図の場合，検出された有音区間から GMM による尤度計算を行った結果，非言語音声区間が 1.0 秒から 1.69 秒間検出されたため，それに対する音素列観測を行った結果，咳と認識された．

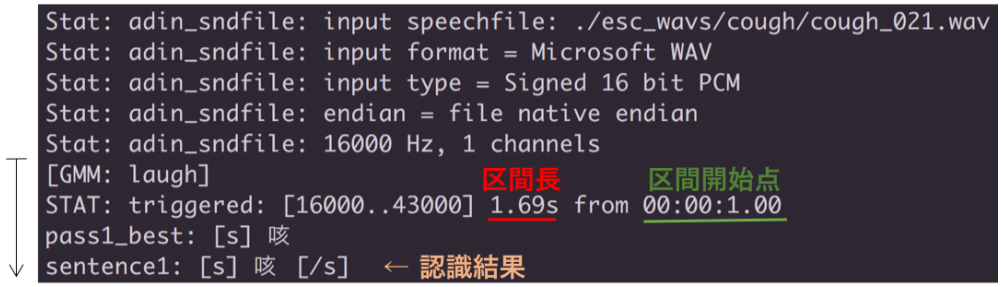

図 4.3 音声ファイル入力時の出力結果
(マイク入力からのリアルタイム認識時と同じ処理を適用)

なお，Julius の仕様上の関係で，同じ始点時刻から複数の有音区間が検出される場合があるが，その場合は最長マッチングを行い，区間長が最長のものを有効とするようにした (図 4.4).

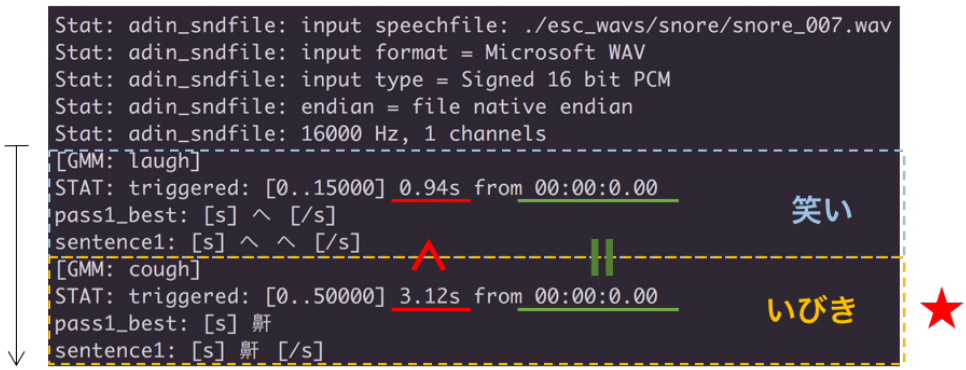

図 4.4 最長マッチングの適用事例

4.3 音響特徴量について

音声から抽出する特徴量については，MFCC(Mel-Frequency Cepstrum Coefficiency; メル周波数ケプストラム係数) を使用することにした．MFCC は人間の聴覚特性を表現することに適した音響特徴量であり，話声に対する音声認識用に提唱されたものであるが，話声以外の多種類の音声を分類する先行研究 [4][5][6][7][11] でも頻繁に使われているため，本手法では MFCC を使うことにした．MFCC の計算手順 [18] は，図 4.5 のようになっている．

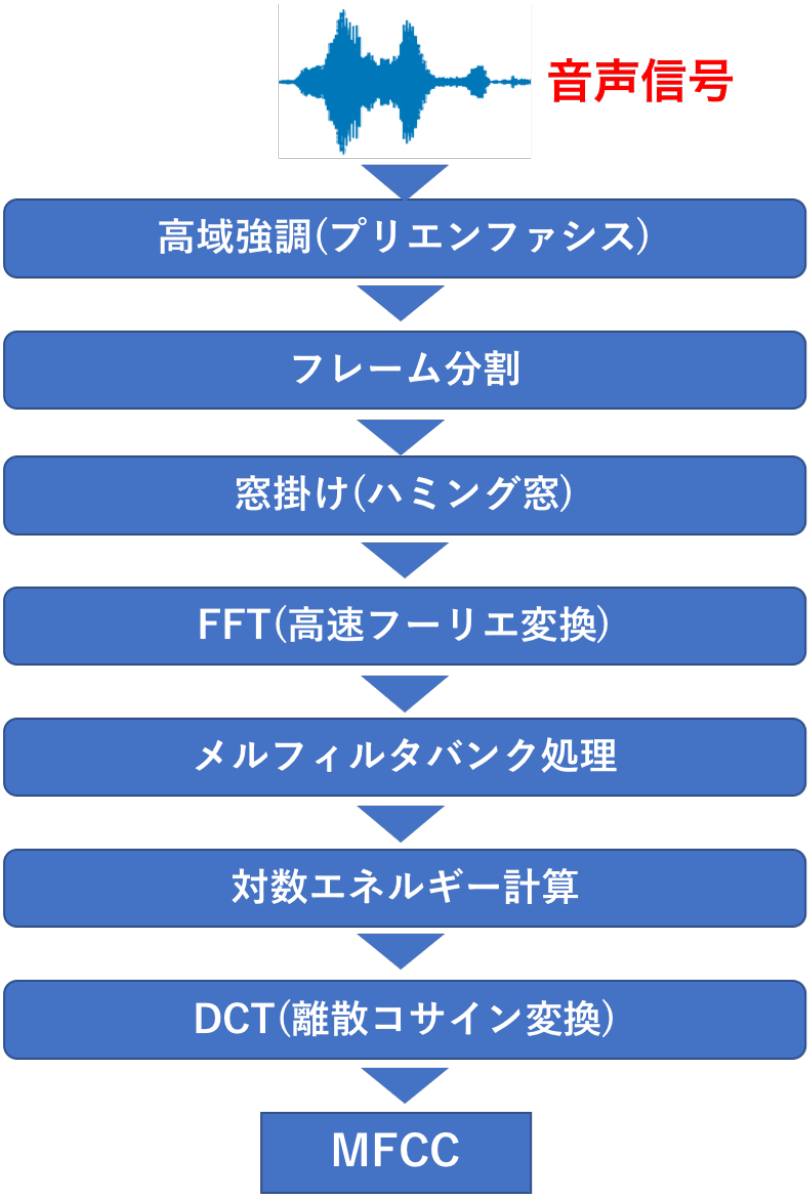

図 4.5 MFCC の計算ステップ

4.4 HTK(Hidden Markov Model ToolKit) について

本研究では，非言語音声区間を検出するために用いる GMM を生成することを目的に，HTK フォーマットの GMM を使用した．HTK フォーマットのモデルを Julius にインポートすることで，自分で定義した音響モデルを用いてリアルタイム認識が可能になる．なお，GMM は中間状態数が 1 つだけの HMM と同値であることから，3 状態[*6]の Left-to-Right HMM を GMM として使用している．

HTK は，HMM の状態定義[*7]や学習を行うためのコマンドが揃ったツールキットであり，Steve Young と Phil Woodland によって 1993 年に開発された．音声認識における利用を想定して開発されたものであるが，HMM が時間的変化の表現に適していることなどから，音声合成 [20][21] やジェスチャー認識 [22]，文字認識 [23][24] や遺伝子解析 [25] の研究など様々な分野における研究で使用されている．

4.4.1 音響特徴量定義

HTK では，モデルに対する学習を行う前に予め特徴量を指定しておく必要がある．Julius で使う GMM を生成するための手順は，Julius 側で資料を掲載[*8] しており，この資料を参考にして特徴量の種類を表 4.1 に合わせる形で行った．

```
SOURCEFORMAT = WAV          # 入力する音声データはwavファイル
SOURCEKIND = WAVEFORM
SOURCERATE = 625            # 625*100nsec = 16kHz
TARGETKIND = MFCC_0_D_A     # MFCC(0番目を含む)+⊿MFCC+⊿⊿MFCC
TARGETRATE = 100000.0       # フレームシフト10ms
WINDOWSIZE = 250000.0       # ウィンドウ幅25ms
USEHAMMING = T              # ハミング窓を使う
PREEMCOEF = 0.97            # 高域強調時のプリエンファシス係数
NUMCHANS = 24               # フィルタバンクのチャネル数
NUMCEPS = 12                # 12番目までのケプストラム係数を使用
ZMEANSOURCE=T               # フレーム単位の直流成分除去
ENORMALISE=F                # 対数エネルギー項を正規化
ESCALE=1.0                  # 対数エネルギー正規化のスケーリング係数
TRACE=0
RAWENERGY=F
```

図 4.6 HTK 用の特徴量定義ファイル (config_hcopy)

[*6] 開始状態，中間状態，終了状態の 3 つ

[*7] 状態数，混合数，共分散行列の型など

[*8] http://julius.osdn.jp/index.php?q=doc/gmm.html

4.4.2 モデル定義

HTK フォーマットの初期モデルの生成を行う．その際，混合数や状態数など状態定義に関する設定を行う必要がある．そのため，状態定義用のファイル (図 4.7) を作成した上で，MakeProtoHMMSet [*9] コマンドを実行することによって，初期モデル (図 4.9) を作成する必要がある．

```
<BEGINproto_config_file>
<BEGINsys_setup>

hsKind: P                       # 連続HMM
covKind: D                      # 対角共分散行列
nStates: 1                      # 中間状態数は1つだけ
nStreams: 1                     # ストリーム数は1つだけ
sWidths: 39                     # 特徴量の次元数
mixes: 32                       # 各状態における混合数
parmKind: MFCC_0_D_A            # 音響特徴量の種類
vecSize: 39                     # 特徴量の次元数
outDir: proto_gmm               # 出力先のディレクトリを指定
hmmList: targetlist_gmm.txt     # 作成したいクラスの一覧を指定

<ENDsys_setup>
<ENDproto_config_file>
```

図 4.7 状態定義用のファイル

なお，生成したい GMM のクラスをテキストファイル (図 4.8) に予め指定しておく必要がある．本研究では，生活音や非言語音を話声やノイズとも識別しながら行いたいため，笑い声 (laugh) やタイピング (key)，話声 (speech) などを指定している．

```
laugh
cough
key
speech
noise
...
```

図 4.8 GMM のクラス一覧を指定したファイル (targetlist_gmm.txt)

[*9] https://github.com/ibillxia/htk_3_4_1/blob/master/samples/HTKDemo/MakeProtoHMMSet

```
  ~o <VecSize> 39 <MFCC_0_D_A> <StreamInfo> 1 39
  ~h "class_name"
<BeginHMM>
  <NumStates> 3
  <State> 2 <NumMixes> 32
  <Stream> 1
  <Mixture> 1 0.0312
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
  <Mixture> 2 0.0312
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
...
  <Mixture> 32 0.0312
    <Mean> 39
      0.0 0.0 0.0 ...
    <Variance> 39
      1.0 1.0 1.0 ...
  <TransP> 3
   0.000e+0   1.000e+0   0.000e+0
   0.000e+0   6.000e-1   4.000e-1
   0.000e+0   0.000e+0   0.000e+0
<EndHMM>
```

図 4.9　HTK フォーマットの GMM 初期モデル

ヘッダ部には，音響特徴量の種類 (~o) とクラス名 (~h) が格納されており，各状態における平均と分散がそれぞれ<Mean>と<Variance>に格納される．また，状態遷移確率が<TransP>に格納されている．

4.4.3 GMM に対する学習

初期モデルの生成した後は，モデルに対する学習を行う．学習時には，音響特徴量定義や状態定義に加えて，タイムスタンプ付きのラベルファイルが必要になるため，まずはラベリングを行った．なおラベルの作成には WaveSurfer[26] を用いた．

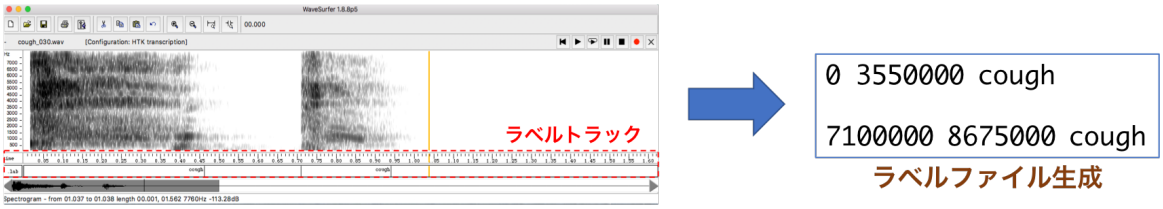

図 4.10 WaveSurfer を用いたラベリング

HTK フォーマットのラベルファイルは，各行が左から順に「開始時刻」「終了時刻」「属性名」から構成されており，時刻を 100ns 単位で表記する必要がある．WaveSurfer でのラベリングでは，HTK フォーマットでラベルファイルを作成することが可能である．

作成したラベルファイルを用いて，HTK での学習コマンド (表 4.3) を実行することで GMM を生成した．

表 4.3 学習時に使用した HTK のコマンド

コマンド名	内容
HInit	k-means 法を用いた初期学習
HRest	Baum-Welch Re-Estimation による学習

学習させたモデルを Julius でのリアルタイム認識時に使用するため，mkbinhmm[*10] コマンドを使って，Julius 用のバイナリ形式へ変換した．

[*10] https://julius.osdn.jp/juliusbook/ja/mkbinhmm.html

4.5 非言語音に対する疑似音素列観測の実装

非言語音認識用の音素列観測 (3.4 節) を実装するために，Julius 側に定義したことについて述べる．

本研究で認識対象とした計 3 種類の非言語音[*11]に対して，提案で検討した疑似音素列定義を，Julius 側に定義することで非言語音認識を実装した．なお認識時には，音響モデルと言語モデル (記述文法) を用いており，音響モデルには Julius 付属の日本語話声認識音素に対する HMM を使用した．この音響モデルで定義されている音素は，表 4.4 のとおりである．

表 4.4 Julius 付属の日本語話声認識音素に対する音響モデルで定義されている音素の一覧

母音	/a/,/e/,/i/,/o/,/u/
長母音	/a:/,/e:/,/i:/,/o:/,/u:/
子音	/b/, /by/, /ch/, /d/, /dy/, /f/, /g/, /gy/, /h/, /hy/, /j/, /k/, /ky/, /m/, /my/, /n/, /ny/, /p/, /py/, /r/, /ry/, /s/, /sh/, /t/, /ts/, /w/, /y/, /z/
促音	/q/
撥音	/N/
無音系	/sp/, /silB/, /silE/

音響モデルと言語モデルに定義した内容については，次頁の図 4.11 のとおりである．

[*11] 「笑い声」「咳」「いびき」の 3 種類

文法定義ファイル (nonverb.grammar)

```
S: NS_B NONVERB NS_E

NONVERB: COUGH
NONVERB: SNORE
NONVERB: LAUGH LAUGH_LOOP

# 左再帰を使い，笑い声特有の音素の繰り返しを表現
LAUGH_LOOP: LAUGH_LOOP LAUGH
LAUGH_LOOP: LAUGH
```

辞書定義ファイル (nonverb.voca)

```
%LAUGH
ハ      h a
ハッ    h a q
ヒ      h i
ヒッ    h i q
...
ホ      h o
ホッ    h o q
```

```
%COUGH
咳   f u q
咳   u q
咳   z u q
咳   u f u q
```

```
%SNORE
いびき   u:
いびき   o:
```

```
%NS_B
<s>    silB

%NS_E
</s>   silE
```

図 4.11　辞書定義および文法定義

Julius の文法定義では左再帰記述によって繰り返しを表現することが可能なことに注目し，文法定義ファイル内の LAUGH_LOOP に定義した．また笑い声は「最低でも 2 音節以上から構成される」と考えたため，そのことを文法定義ファイルの 3 行目に反映させた．

第 5 章

実験と評価

5.1 目的

提案した認識システムによる，生活音・非言語音認識精度の評価を行う．

実験における主目的は，「生活音や非言語音を，話声や雑音と識別する形で認識ができるか」を評価することである．そのため，リアルタイム認識時と同じ処理を行う形で，生活音・非言語音の分類精度について評価を行いたいが，その前段階として「非言語音に対する疑似音素列定義」が有効かどうかを検証する必要があるため，その事についても評価実験を行うことにする．

実験時に使用する音声データについては，様々な発声者や環境下での音声が入ったデータセットを使用することにした．その理由は，人から発する音声 (非言語音/話声) は個人差による影響があることに加えて，生活音についても環境差による影響を考慮したためである (例えばタイピングの音であればキーボードの違いから音色などが異なる)．

5.2 構成

前節で述べた実験目的について考慮した結果，下記の実験を行うことにした．

実験 1
非言語音に対する疑似音素列定義の評価実験

実験 2
生活音・非言語音に対するリアルタイム認識精度の評価実験

実験 3
リアルタイム笑い検出精度の評価実験

まず最初に，非言語音に対する疑似音素列定義が有効かどうかを検証するため，提案 (図 3.1) における「非言語音声区間検出」で正しく検出されたという仮定のもとで，非言語音に対する音素列観測の実験を行う (実験 1)．

次に，実験の主目的である「生活音や非言語音を，話声や雑音と識別する形で認識ができるか」について評価するために，提案手法 (図 3.1) 全体の処理を使って，生活音・非言語音などに対するリアルタイム認識の実験を行う (実験 2)．

続いて，実験 2 で用いた音声データは全て単独事象 (Isolated Event) であったが，実際に笑い声を発する場合は発話しながら笑い声が出る時が多いため，「発話中に笑った場合でも笑いを正確に検出できるかどうか」を評価するための実験を行う (実験 3)．

なお，実験時には音声ファイル入力による認識結果を使用しているが，リアルタイム認識を想定している実験 2 と実験 3 においては，4.2.2 節で述べたように音声ファイルに対しても「マイク入力からのリアルタイム認識時の処理」を適用していることに注意されたい．

5.3 実験 1: 非言語音に対する疑似音素列定義の評価実験

非言語音に対する疑似音素列定義を用いた，音素列観測に対する評価を行った．「笑い声」「咳」「いびき」の計 3 種類に対する疑似音素定義 (図 4.11) を用いて，非言語音のみからなる音声データ同士で分類する形で評価する．なお検証に使用した音声データは，会議音コーパスから [27] と環境音コーパス [28] から予めトリミングされたものを使用した．

5.3.1 結果

分類結果は表 5.1 のようになった．なお，表内の数値は分類された件数である．

「笑い声」に対する結果については良かったものの，「咳」と「いびき」同士での誤分類が多かった．特に「咳」に対する結果が悪く，検出率が半分を下回る結果となった．

表 5.1 実験 1 の結果 (疑似音素列定義を用いた非言語音の分類結果)

		観測されたクラス		
		笑い声	咳	いびき
実際の音声	笑い声	**70**	23	7
	咳	21	**70**	53
	いびき	3	22	**42**

次頁の考察で，今回行った非言語音に対する疑似音素列定義に対する有効性について触れることにした．

5.3.2 考察

「笑い声」「咳」「いびき」の計 3 種類に対する疑似音素定義 (図 4.11) が，各種の非言語音に対して表現できているかについて確かめるため，疑似音素列定義について検討を行うことにした．

表 5.1 より「咳」「いびき」同士での誤分類が多かったため，咳といびきに対する疑似音素列定義を変更しながら検討することにした．

まずは「咳」に対する音素列を一部削除する形で再度行った結果，「咳」から疑似音素列”/u/-/q/”を削除した結果，「咳」の音声からの結果が「いびき」に誤分類されることが多発することが分かった．しかし「笑い声」に対する結果に大きな変動は無かった．

図 5.1 「咳」に対する疑似音素列定義を一部削除した際の結果

続いて「いびき」の子音に近いと考えられる，/z/や/g/を疑似音素列に対して検討した結果，「いびき」に対する検出率こそ上昇したものの，「咳」から「いびき」に誤分類されることが多くなってしまった．しかし，図 5.1 同様「笑い声」に対する結果に大きな変動は無かった．

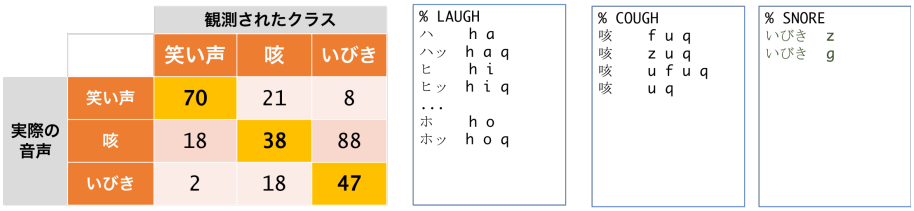

図 5.2 「いびき」に対する子音的な疑似音素列定義を検討した際の結果

以上の検討から，今回の疑似音素列定義によって，「笑い声」については表現できたが，「咳」と「いびき」に対しては互いに誤分類が頻発していることから，その 2 種類に対する定義に課題があると考えた．

5.4 実験 2: 生活音・非言語音に対するリアルタイム認識精度の評価実験

提案手法によるリアルタイム認識時に，生活音や非言語音が検出できることに加えて，話声や雑音とも識別ができるかについても評価する．

この実験では，生活音および非言語音に加えて，話声やノイズも認識対象に追加する形で評価を行うことにした．今回の認識対象とサンプル数の関係は表 5.2 のとおりである．実験 2 での音声データは，実験 1 と同様に会議音コーパス [27] と環境音コーパス [28] からの音声データを使用した．

表 5.2 実験 2 で使用した音声データ

種類	総サンプル数	出典
笑い声	100	[27]
咳	40	[28]
いびき	40	[28]
掃除機	40	[28]
タイピング	40	[28]
歯磨き	40	[28]
話声	1200	[27]
ノイズ	78	[27]

なお，同様の実験が行われた既存研究 [4][5][7] では音声データと認識結果が 1 対 1 で対応していることが多いが，本実験では図 5.3 のように 1 対多で認識結果が観測されることがあることに注意されたい．これは，提案手法における「有音区間検出」などにおいて複数の区間が検出される場合があるためことや，尤度計算における「尤度計算 (GMM)」で毎フレームごとに尤度計算が行われることによるものである．

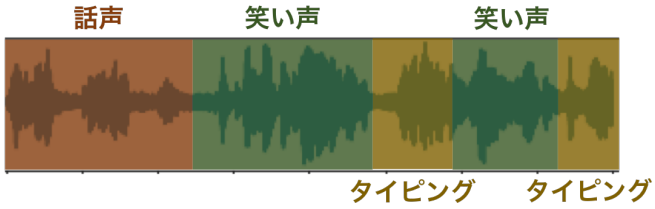

図 5.3 音声データと認識結果が 1 対多になった場合の例

5.4.1 結果

5-Fold Cross Validation で検証を行った結果は表 5.3 のようになった．なお，この表における値は (観測された音声データの数/各クラスに対する総サンプル数) でクラス毎に表している．また，図 5.3 のように 1 個の音声データから複数のクラスが観測される場合があるため，表内における分子の横方向の合計が総サンプル数 (分母) に一致するとは限らないことに注意されたい．

「話声」による生活音・非言語音の誤検出は少なかった．今回の実験で使用した「話声」の音声データは計 24 名の話者 (男女 12 名ずつ ×50 サンプル) によるものであることに加えて，発話内容の差異によって話声の音響的特徴が幅広いことを踏まえると，話声との識別結果は良好であったと考えた．

その一方で非言語音に対する認識結果が良くないものとなった．「咳」に対する認識結果が「笑い声」の方へ誤分類されることが多かったことに加えて，「いびき」の音から「ノイズ」や生活音に属するクラスが誤観測されることが多く見受けられた．また「掃除機」の音声を入力した際に非言語音が誤観測されることが多かった．これは，提案における「非言語音声区間検出」が行われてしまったためである．

表 5.3 実験 2 の結果

		観測されたクラス							
		笑い声	咳	いびき	掃除機	タイピング	歯磨き	話声	ノイズ
実際の音声	笑い声	**57/100**	24/100	7/100	0/100	1/100	0/100	3/100	6/100
	咳	20/40	**18/40**	3/40	0/40	7/40	2/40	9/40	4/40
	いびき	8/40	10/40	**16/40**	0/40	6/40	3/40	4/40	18/40
	掃除機	0/40	1/40	14/40	**22/40**	0/40	0/40	1/40	1/40
	タイピング	0/40	2/40	0/40	0/40	**35/40**	1/40	0/40	4/40
	歯磨き	2/40	2/40	1/40	0/40	1/40	**32/40**	1/40	0/40
	話声	1/1200	5/1200	5/1200	0/1200	0/1200	0/1200	**1200/1200**	2/1200
	ノイズ	3/78	1/78	0/78	0/78	4/78	2/78	9/78	**48/78**

実験 2 の認識結果が良くなかった箇所に関する原因について検討するため，次頁にて考察を行った．

5.4.2 考察

実験 2 の結果 (表 5.3) に対して，特に非言語音認識に関する認識精度に課題があったため，その原因について検討を行うことにした．

実験 1 の結果 (5.1) より，提案における「非言語音声区間検出」で正確な区間検出ができていれば，実験 2 でも非言語音認識結果に大きな差異が無いはずである．しかしながら，実験 2 において「咳」の音声から「笑い声」が誤検出されることが多くなってしまっている．そこで，「非言語音声区間検出」に課題があると考えて，次のような検討を行うことにした．

- 非言語音による認識結果が良くなかったことについて
  - 原因として次の 2 つを考えた
    1.「非言語音声区間検出」の際に，尤度差が閾値以上とならず棄却されたため
    2.「非言語音声区間検出」の途中で，非言語音に対する尤度と「ノイズ」や「話声」との尤度差が小さくなって非言語音声区間が終了したため，音声区間の始点/終点が望ましくない位置となってしまったため
  - 区間検出用の GMM から，「ノイズ」を除外した上で検証を行うことにした (表 5.4)
- 「掃除機」の音声データから非言語音の誤観測が多かったことについて
  - その原因としては，次のように考えた
    - 「掃除機」の尤度と非言語音 3 種類との尤度差が小さくなりやすく，「非言語音区間検出」が行われてしまうことが多かったため
    - 「掃除機」に対する GMM か非言語音に対する GMM の質が良くないことが原因と考えられる
  - 区間検出用の GMM から，「掃除機」を除外した上で検証を行うことにした (表 5.5)

以上の検討より，「ノイズ」や「掃除機」の GMM をクラスを除外した上で，実験 2 と同様の実験を行うことにした．結果はそれぞれ，表 5.4, 表 5.5 のようになった．

「ノイズ」を除外した場合，非言語音と同じく人から発する音である「話声」による誤検出が増えたことに加えて，非言語音の音声から生活音が誤観測されることが増加した．また，非言語音に対する検出率が大きく向上することは無かった．

表 5.4 「ノイズ」を除外した時の結果

		観測されたクラス						
		笑い声	咳	いびき	掃除機	タイピング	歯磨き	話声
実際の音声	笑い声	**61/100**	27/100	7/100	0/100	1/100	0/100	2/100
	咳	18/40	**20/40**	2/40	0/40	11/40	2/40	8/40
	いびき	13/40	8/40	**17/40**	0/40	15/40	5/40	5/40
	掃除機	0/40	2/40	13/40	**22/40**	0/40	0/40	1/40
	タイピング	1/40	4/40	2/40	0/40	**37/40**	1/40	0/40
	歯磨き	2/40	3/40	1/40	0/40	2/40	**32/40**	1/40
	話声	5/1200	61/1200	36/1200	0/1200	1/1200	0/1200	**1193/1200**

「掃除機」を除外した場合，実験 2 の結果 (表 5.3) と殆ど違いが無かったため，「掃除機」に対するモデルではなく非言語音に対するモデルが良くなかったためだと考えた．

表 5.5 「掃除機」を除外した時の結果

		観測されたクラス						
		笑い声	咳	いびき	タイピング	歯磨き	話声	ノイズ
実際の音声	笑い声	**57/100**	24/100	7/100	1/100	0/100	3/100	6/100
	咳	20/40	**18/40**	3/40	7/40	2/40	9/40	4/40
	いびき	8/40	10/40	**16/40**	6/40	3/40	4/40	18/40
	タイピング	0/40	2/40	0/40	**35/40**	1/40	0/40	4/40
	歯磨き	2/40	2/40	1/40	1/40	**32/40**	1/40	0/40
	話声	1/1200	5/1200	5/1200	0/1200	0/1200	**1200/1200**	2/1200
	ノイズ	3/78	1/78	0/78	4/78	2/78	9/78	**48/78**

以上の結果から，非言語音に対する GMM が良くなかったことによって，GMM による尤度計算が望ましくないものとなり，「非言語音声区間検出」が上手くいかなかったことが原因と考えた．非言語音に対する GMM が良くなかったのは，学習に用いたデータ量 (特に咳といびき) が少なかったためと考えられる．

実験 2 で使用した音声データは，全て単独の音声イベントである．例えば，「笑い」の音声であれば単に笑い声のみが入った音声データを実験 2 で使用した．しかし，実際には笑い声が出る時は発話しながら笑う場合も多いため，「発話中に笑った場合でも笑いを正確に検出できるか」を検証する必要があると考えた．そこで，次に実験 3 を行うことにした．

5.5 実験 3: リアルタイム笑い検出精度の評価実験

発話中に笑った場合でも，話声と識別して笑いを正しく検出できているかについて評価を行う．なお評価時に使用した音声データは，会議音コーパス [27] に付属する発話内容に対する書き起こしデータから，発話中に笑った時の音声を使用した．

> たぶん、そゆ時は旅館には行かないですね。**<フッフフ>**
> まずビジネスホテルでいいなって思っちゃいますね、ですからね。

図 5.4 発話中に笑った時の書き起こし文の例

会議音コーパスにおいて発話中に笑ったのは計 73 回あった．なお，この実験で使用した GMM の学習データは全て表 5.2 からの音声を使用し，認識時のパラメータ類は全て実験 2 の時と同じものを使用している．

結果は表 5.6 のようになった．なお表内における値は，(観測された音声データの数/実験で使用した総サンプル数) で各クラス毎に表している．発話中に笑った場合に笑いを検出できたのは，全 73 サンプル中の 65.7% にあたる 48 個であった．また，その中から話声と笑い声を同時に検出できたのは 13 個となった．なお，生活音に対するクラスが誤観測されることは無かった．

表 5.6 実験 3 の結果

笑い声	**48/73**
咳	8/73
いびき	5/73
掃除機	0/73
タイピング	0/73
歯磨き	0/73
話声	**22/73**
ノイズ	1/73

5.6 提案に対する評価

疑似音素列定義による非言語音認識について

非言語音に対する疑似音素列定義については，実験 1 の結果より分類精度が優れたものとは言い難いが，使用した HMM は既にあるモデルを使用しており図 4.11 のように文法定義を行えば同じことが可能なので再現がしやすいこと，また図 4.11 の定義を拡張することで，観測された音素列から笑い声の更なる分類や，他の非言語認識にも拡張できる可能性があるため，疑似音素列定義は今後有用になる可能性をもつアプローチと考えた．

GMM を用いた非言語音声区間検出について

実験 1 において，「非言語音区間検出」ができた仮定の下で，音素列観測を行った結果はそれなりのものであった．そこで今度は実験 2 として「非言語音区間検出」に対する実験を行った結果，実験 1 の結果から想定していたものと異なっていたことに加えて，生活音 (特に掃除機) からの非言語音が誤観測されることが多かった．

しかし実験 3 において，発話中しながら笑った音声でも約 65% の割合で笑いを検出することができるという結果を得た．

以上より，GMM による非言語音声区間検出は，笑い声検出には有効なアプローチになるが，学習データ不足などから非言語音に対する GMM の質が良くない場合，非言語音以外による誤検出が起こりやすくなる所が課題になると考えた．

GMM を用いた話声・雑音との識別について

本研究の研究目的の一つである，「生活音・非言語音を，話声や雑音と識別する形で認識すること」について述べる．

実験 2 の結果より，話声との識別については多くの話者による音声データを用いたが，話声から生活音・非言語音が誤検出されることが少なかったため，話声との識別に有効であったと考えた．これは GMM の学習に用いた音声データの量をが多かったことも大きな要因であると考えた．

しかしその一方で，ノイズが入力された際に生活音や非言語音の誤検出されることが多くあったため，雑音との識別については課題が残った．

第 6 章

おわりに

6.1 まとめ

人の生活行動や心情把握などを目的として，生活音や非言語音を，話声や雑音と識別しながらリアルタイム認識ができるシステムの開発を行った．

多種類の非言語音および生活音を対象としてリアルタイム認識を行う既存研究の多くは，「話声と非言語音が共存していないこと」や「雑音入力による誤検出対策が行われていない」という課題があり，さらにその手法が話声や非言語音の認識に向いていないという仮説を得た．そこで，認識対象の音声 (特に非言語音) がもつ音響的性質に加えてリアルタイム認識時の要件を考慮した上で，認識手法を提案した．さらに，提案手法に合った音声認識アプリケーションである Julius を用いてリアルタイム認識の実装を行った．

提案にあたって，非言語音に対して疑似音素列を定義することで認識に活かせることを期待したため，笑い声や咳音などに対する疑似音素列の定義を行った．その際，咳に対しては既存手法があったものの，それ以外の非言語音に対しては同様の手法が無かったため，音響的性質に関する既存研究を調査した上で，認識時の文法定義や辞書定義に反映していく形で定義を行った．

提案手法の認識精度を検証することを目的に，様々な話者や環境下での音声を使って 3 種類の評価を行った．その結果，疑似音素列定義による非言語音同士での分類はそれなりの結果となったものの，連続音声からのリアルタイム認識を想定した処理を含めた場合，「非言語音の検出率」や「雑音入力による非言語音の誤検出」に関して課題が残った．その一方で話声による生活音および非言語音の誤検出は抑えることができたことに加えて，生活音については「タイピング」と「歯磨き」の音声に対して比較的正確な認識ができていた．また発話中に笑った場合でも，リアルタイム認識時と同様の設定で約 65% の割合で笑いを検出することができたため，連続音声からのリアルタイム笑い声検出には本手法が有効になると考えた．

6.2 今後の課題

学習用音声データについて

実験時に使った音声データの量は多くなかったため，大量の学習用データを用意した上で同様の検証を行いたい．非言語音声区間検出に用いる GMM の学習には，十分な量の音声データを用意することができれば GMM の混合数を増やすことで性能向上が期待できる [10]．しかし，生活音や非言語音の音声データが大量に入ったデータセットが殆ど無い上に，もし用意できたとしても学習時にタイムスタンプ付きラベルデータを要するため，ラベル付けのために大きな時間コストを要してしまうという課題もある．

非言語音に対するさらなる分類

認識精度の改善ができれば，次は非言語音に対する音素列観測の活用を検討したい．提案手法によって，ただ笑い声が観測されるだけではなくそれに対する音素系列も付帯している．同じ「笑い声」でもその聞こえ方によってその意味合いが変わってくる [29] ため，観測された音素系列を活用して愛想笑いや苦笑いなどを検出することで，状況把握や心情把握に役立てることを検討したい．

音響モデルの追加

非言語音に対する音素列観測時に用いた音響モデルとして，本研究では日本語音素に対する音響モデルを使用した．笑い声やいびきなどの非言語音に対して，日本語の音素を並べたものを疑似音素列として定義することで認識を行っているが，その音素に日本語以外の音素を追加することを今後検討したい．

その理由として，フランス語の音素/r/がいびきの音に近いという独特の性質を持つことに注目したためである [30]．音響モデルは話声に対する音素の性質を表現したモデルであるため，フランス語の音声認識システムで使われる，音素/r/に対する音響モデルを今回使用した音響モデルを追加するなどして，非言語音認識への活用できるか検討を行いたい．

Julius での音響モデルは，音響特徴量などの値 (表 4.1) を合わせる形で HTK フォーマットの HMM を生成することで，今回使用した日本語話声用音響モデルに追加することができる．Julius で使用可能なフランス語認識用の音響モデルがあるかどうか調べたものの，見つけることができなかった．

音声コーパスとラベルデータを用意することで，自分で生成した音響モデルに使用することが可能であるため，今後は様々な言語での音響モデルを用いることで，非言語音認識などをより正確に行うことや，より多くの種類を認識することができるようにすることを今後検討していきたい．

参考文献

[1] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, Vol. 38, No. 3, pp. 50–60, 2005.

[2] 坂本真樹. 超スマート社会における感性 ai. 横幹連合コンファレンス予稿集 第 9 回横幹連合コンファレンス, pp. D–1. 横断型基幹科学技術研究団体連合 (横幹連合), 2018.

[3] 大石康智. あらゆる音の検出・識別を目指して: 音響イベント検出研究の現在と未来. 日本音響学会研究発表会講演論文集 日本音響学会 編, pp. 1521–1524, 2014.

[4] Mirco Rossi, Sebastian Feese, Oliver Amft, Nils Braune, Sandro Martis, and Gerhard Tröster. Ambientsense: A real-time ambient sound recognition system for smartphones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pp. 230–235. IEEE, 2013.

[5] Angelos Pillos, Khalid Alghamidi, Noura Alzamel, Veselin Pavlov, and Swetha Machanavajhala. A real-time environmental sound recognition system for the android os. *Proceedings of Detection and Classification of Acoustic Scenes and Events*, 2016.

[6] 柴田健作, 中村圭佑, 中臺一博ほか. 会話内非言語音声情報抽出のための音響特徴量の検討. 第 78 回全国大会講演論文集, Vol. 2016, No. 1, pp. 539–540, 2016.

[7] Arslan Shaukat, Muhammad Ahsan, Ali Hassan, and Farhan Riaz. Daily sound recognition for elderly people using ensemble methods. *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*, pp. 418–423, 12 2014.

[8] Mohamed A Sehili, Dan Istrate, Bernadette Dorizzi, and Jerome Boudy. Daily sound recognition using a combination of gmm and svm for home automation. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1673–1677. IEEE, 2012.

[9] 石塚健太郎, 藤本雅清, 中谷智広. 音声区間検出技術の最近の研究動向. 日本音響学会誌, Vol. 65, No. 10, pp. 537–543, 2009.

[10] Akinobu Lee, Keisuke Nakamura, Ryuichi Nisimura, Hiroshi Saruwatari, and Kiyohiro Shikano. Noice robust real world spoken dialogue system using gmm based rejection of unintended inputs. *ICSLP2004: the 8th International Conference on Spoken Language*

Processing, pp. 173–197, 2004.

[11] Shin-ya Takahashi, Tsuyoshi Morimoto, Sakashi Maeda, and Naoyuki Tsuruta. Detection of coughs from user utterances using imitated phoneme model. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[12] Thomas Drugman, Jerome Urbain, Nathalie Bauwens, Ricardo Chessini, Anne-Sophie Aubriot, Patrick Lebecque, and Thierry Dutoit. Audio and contact microphones for cough detection. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[13] 大原遼. 対話音声の笑い声や笑い方についての分析. 2005.

[14] J. Urbain, H. Çakmak, and T. Dutoit. Automatic phonetic transcription of laughter and its application to laughter synthesis. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 153–158, Sep. 2013.

[15] 寺井修. 鼾の音響学的研究. 耳鼻咽喉科臨床, Vol. 68, No. 3special1, pp. 373–397, 1975.

[16] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius — an open source real-time large vocabulary recognition engine. In *INTERSPEECH*, 2001.

[17] 李晃伸, 河原達也. Julius を用いた音声認識インタフェースの作成. ヒューマンインタフェース学会誌, Vol. 11, No. 1, pp. 31–38, 2009.

[18] Steve Young. The htk book version 3.4. 1. *http://htk. eng. cam. ac. uk*, 2009.

[19] Katunobu Itou, Mikio Yamamoto, Kazuya Takeda, Toshiyuki Takezawa, Tatsuo Matsuoka, Tetsunori Kobayashi, Kiyohiro Shikano, and Shuichi Itahashi. Jnas: Japanese speech corpus for large vocabulary continuous speech recognition research. *Journal of the Acoustical Society of Japan (E)*, Vol. 20, No. 3, pp. 199–206, 1999.

[20] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda. The hmm-based speech synthesis system (hts) version 2.0. In *SSW*, pp. 294–299. Citeseer, 2007.

[21] Keiichi Tokuda, Heiga Zen, and Alan W Black. An hmm-based speech synthesis system applied to english. In *IEEE Speech Synthesis Workshop*, pp. 227–230, 2002.

[22] 村尾和哉, 寺田努, 矢野愛, 松倉隆一, 西尾章治郎ほか. センサ内蔵型モバイル機器を用いたジェスチャ認識に関する考察. 研究報告モバイルコンピューティングとユビキタス通信 (MBL), Vol. 2010, No. 28, pp. 1–8, 2010.

[23] 須藤隆. 隠れマルコフモデルに基づくオンライン手書き文字列認識に関する研究. 2002.

[24] Md Hasnat, SM Habib, Mumit Khan, et al. Segmentation free bangla ocr using hmm: Training and recognition. 2007.

[25] Kiyoshi Asai, Tetsushi Yada, and Katunobu Itou. Finding genes by hidden markov models with a protein motif dictionary. *Genome Informatics*, Vol. 7, pp. 88–97, 1996.

[26] Kåre Sjölander and Jonas Beskow. Wavesurfer-an open source speech tool. In *Sixth International Conference on Spoken Language Processing*, 2000.

[27] Kazuyo Tanaka, Katunobu Itou, Ryuichi Oka, and Hiroshi Matsumura. Rwcp meeting speech corpus 2001. In *Proceedings of the Annual Conference of JSAI Proceedings of the 16th Annual Conserence of JSAI, 2002*, pp. 197–197. The Japanese Society for Artificial Intelligence, 2002.

[28] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018. ACM, 2015.

[29] Hiroki Tanaka and Nick Campbell. Classification of social laughter in natural conversational speech. *Computer Speech & Language*, Vol. 28, No. 1, pp. 314 – 325, 2014.

[30] 田口亜紀ほか. フランス語初学者を対象とした発音指導. 共立女子大学・共立女子短期大学総合文化研究所紀要, Vol. 24, pp. 41–51, 2018.

謝辞

本研究の遂行にあたって，認識手法の提案および実装方法に関して，数多のアドバイスを頂いた指導教員の沼尾雅之教授に心より感謝いたします．2016 年 4 月以来，足掛け 3 年間にわたってお世話になりました．また，研究室のゼミや見守りプロジェクトを有意義な機会にして頂いた，沼尾研究室の皆様と株式会社 WCL(ワイヤレスコミュニケーション研究所) の皆様に感謝いたします．

最後に，小学校から大学院までの 18 年にもわたる学生生活をサポートして頂いた家族に感謝します．以上を謝辞といたします．

付録 A

Julius のインストールについて

図 3.1 の処理を Julius で行えるようにするためには，インストールの際に configure オプションとして--enable-gmm-vad を指定する必要がある．Julius のインストールまでに必要なコマンドを以下に記載する．

```
$ mkdir Julius
$ cd Julius/
$ wget https://github.com/julius-speech/julius/archive/v4.4.2.1.
    tar.gz
$ tar xvzf v4.4.2.1.tar.gz
$ cd julius-4.4.2.1/
$ ./configure --enable-gmm-vad
$ make
$ sudo make install
```