音声言語情報処理 32−16
（2000. 7. 15）

音声補完: “TAB” on Speech

後藤 真孝　　伊藤 克亘　　速水 悟

電子技術総合研究所

{goto, kito, hayamizu}@etl.go.jp

あらまし　本稿では，ユーザがある単語を最後まで思い出せずに断片だけを発話しても，システム側がその続きを補って入力することを可能にする「音声補完」という新しい音声インタフェース機能を提案する．既にテキストインタフェースでは，ファイル名の入力等で補完の概念が広く受け入れられているが，音声では効果的な補完機能は提案されていなかった．我々は，ユーザが単語発声途中で有声休止 (母音の引き延ばし) をおこなうと，それに続く補完候補の一覧を見ることができるインタフェースを構築し，労力をかけずに自発的に補完機能を呼び出しながら音声入力することを可能にする．実際に，有声休止検出機能と補完候補作成可能な音声認識機能を備えたシステムを実装して運用し，音声補完の有用性を確認した．

Speech Completion: “TAB” on Speech

Masataka Goto　　Katunobu Itou　　Satoru Hayamizu

Electrotechnical Laboratory

1-1-4 Umezono, Tsukuba, Ibaraki 305-8568 Japan

Abstract　This paper describes a novel speech interface function called *speech completion*, which enables a user to input a word by completing a word fragment uttered by the user. Although the concept of completion has been widely accepted in text-based interfaces, effective completion for speech has not been proposed. We enable a user to invoke completion intentionally and effortlessly by building an interface that displays candidates of completion when a filled pause is uttered (a vowel is lengthened) during a word. In our experience with a system that includes a filled pause detector and a speech recognizer capable of listing completed words, the effectiveness of the speech completion was confirmed.

1　はじめに

音声対話が，我々人間にとって快適で優れた情報交換手段である一つの理由は，対話相手が様々な形で自分の発話や思考の手助けをしてくれるからである．人間同士の音声対話では，話者がある単語を最後まで思い出せず，その途中に言い淀むことがある．すると対話相手はしばしば，話者の言いたいことを推測して候補を提示することで，話者が思い出すのを手助けしてくれる．例えば，話者が「音声補完」という単語全体を思い出せず「おんせいー」と言い淀むと，対話相手が「音声補完?」と問いかけて手助けをしてくれる．これは，発話された単語の断片の続きを補うことで，話者が述べようとしている単語全体の候補を提示している，つまり，単語を補完していると見なすことができる．本研究は，このような補完による手助けという概念を音声入力の枠組みに導入することで，音声認識を中心とした音声インタフェースをより使いやすくすることを目的とする．

補完 (completion) の概念は，テキストインタフェースでは既に広く受け入れられている．例えば，tcshやbashなどのUNIXシェルや，Emacs/Muleなどのテキストエディタは，ファイル名やコマンド名の補完機能を提供している．特に，コマンド行の補完機能は，1970年代初頭には既に複数のシステムに導入されていた[1]．こうした補完機能では，ユーザが補完機能を呼び出すキー (以下，補完トリガーキー (completion-trigger key) と呼ぶ) を押したときに，途中までタイプされた単語の断片の続きが補われる．この補完トリガーキーには，慣例として **“TAB”** キーが用いられることが多い．他にも，Netscape CommunicatorやInternet ExplorerなどのWWWブラウザでは，URL等の自動補完 (autocompletion) 機能が導入されている．これは，ユーザがタイプしている最中に，システム側が補完候補一覧を次々と提示していく機能である．ここでは，以上のようなテキストを対象とした補

完を総称して，テキスト補完と呼ぶ．テキスト補完は，入力したい名前がうろ覚えで正確に思い出せない場合に有効なだけでなく，名前が長い場合や複雑な文字列の場合に，すべてをタイプしなくても入力が可能であるという点でも優れている．

補完機能は，近年，テキストインタフェース以外でも導入され始めている．例えばペン入力では，予測ペン入力インタフェース[2]やPOBox[3]などの自動補完機能を持ったインタフェースが提案されている．しかし，音声入力インタフェースでは，音声入力中に自然に補完機能を呼び出す手段がなかったこともあって，効果的な補完機能はこれまで提案されていなかった．

本稿では，「音声補完(speech completion)」という新しい音声インタフェース機能を提唱し，ユーザが音声入力中に言い淀む(有声休止をおこなう)と計算機が補完候補を提示して手助けをしてくれるインタフェースを提案する．以下の章では，まず2章で音声補完の概念を述べ，3章で我々が構築した音声補完機能を持つ音声入力インタフェースを紹介する．次に，4章で具体的な実現手法について説明し，5章でシステムの実装とその動作結果を示す．最後に，6章でまとめを述べる．

2 音声補完

「音声補完」とは，計算機システムに対する音声入力中に，ユーザが補完機能を呼び出すことができるようにするための新たな音声入力インタフェース機能の総称である．ユーザが発話した断片をシステム側が補完してくれることで，以下のような利点が得られる．

- 記憶補助
  入力したい内容がうろ覚えでも，途中まで思い出して発声することができれば，システム側の手助けにより入力することが可能になる．
- 省力化
  入力したい内容が長くて複雑なときに，その内容を特定するのに十分な部分まで発声するだけで，システム側が残りを補完して入力することができる．
- 心理的抵抗の低減
  従来の音声インタフェースの多くが，ユーザ側にすべての音を最後まで丁寧に発声することを強いていたのに対し，音声補完では，途中まで発声するだけで入力できるため，心理的抵抗が少なく使いやすい．

音声補完の方式には，テキスト補完と同様に，補完トリガーキーによる補完と，発声中に次々と補完候補を出すような自動補完の二つが考えられる．しかし，音声で自動補完をしようとしても，発話した断片を認識する時点で既に曖昧性が大きいため，テキスト補完と同じような精度で適切な候補を提示し続けることはできず，自動補完機能自体が煩わしくなってしまう可能性が高い☆1．つまり，音声の自動補完は不適切な機能となりやすい．そこで，音声補完の場合には，ユーザが候補を見たいと思うタイミングで，自発的に補完機能を呼び出せることが重要となる．使いやすい音声補完を実現するためには，どのように補完機能を呼び出すか，つまり音声における補完トリガーキー("TAB"キー)を何にするかが鍵となる．

本研究では，言い淀み現象の一つである有声休止(filled pause)に補完トリガーキーの役割を担わせることで，ユーザが自分の意志で，労力をかけずに補完機能を呼び出すことを可能にする．補完トリガーキーとして，音声入力中に有声休止をおこなうことは，人間(特に日本人)にとって極めて自然である．実際，人間同士の対話でも，有声休止は似た役割で用いられることがあり，話者は，次の単語を思い出すまでの間，時間を稼ぐために有声休止を用いたり，場合によっては対話相手の助けを期待しながら有声休止を用いたりする．

音声補完の対象には，単語や文節，文章など様々なレベルが考えられるが，本稿では以下，単語のみを取り上げて議論する．つまり，単語補完機能に論点を絞り，有声休止としては，単語の途中の母音の引き延ばしだけを対象とする．ただし，ここでの単語は，文法上の1単語ではなく，音声認識システムの単語辞書上(言語モデル上)の1単語とする．したがって，例えば「宇多田ヒカル」のような姓名が一つの単語として登録されているときに，「うただー」と「だ」の音で有声休止をおこなうと，「宇多田ヒカル」が補完候補の一つとして得られる．

3 音声補完機能付き音声入力インタフェース

我々が今回構築した音声補完機能付き音声入力インタフェースの機能を説明する．ユーザは，以下のよ

☆1 テキストの自動補完の場合には，部分文字列が確実に与えられていくため，ユーザが望まないときに候補を提示しても，比較的妥当な範囲に候補を絞り込むことが可能である．それでも，ユーザの意志とは無関係に不適切な候補が提示されると，煩わしく感じられるという問題が起きている．

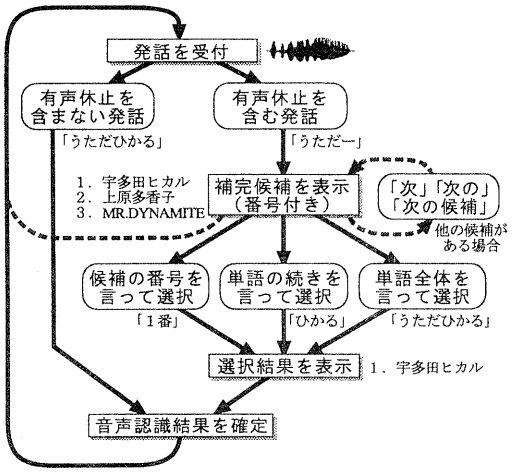


図 1: 音声補完の操作の流れ

うに有声休止を用いて音声補完しながら，単語を入力することができる (図 1).

1. 単語の発声途中で母音を引き延ばすと，既に発声された断片から始まる補完候補 (単語) の一覧が，番号付きで即座に表示される．
   (ex. 「うただー」と入力すると,「1. 宇多田ヒカル, 2. 上原多香子, 3. MR.DYNAMITE」のように補完候補が表示される．)
2. 補完候補が多くて画面に入りきらないときには，「次の候補」というマークが表示される．その場合,「次」「次の」「次の候補」のいずれかを言うことで，他の候補を見ることができる．補完候補が不適切なとき，あるいは別の単語を入力したくなったときは，次の 3. の選択をせずに別の発話に移ってもよい．
3. ユーザは候補一覧を見ながら，以下の 3 通りの方法で補完候補を選択できる．
   (a) 候補の番号を言って選択する．
   (ex. 「1番」か「1」と言う．)
   (b) 単語の続きを読み上げて選択する．
   (ex. 「ひかる」と言う．)
   (c) 単語全体を頭から読み上げて選択する．
   (ex. 「うただひかる」と言う．)

   選択すると，その候補は強調表示され，音声認識結果として確定される．

音声補完は，一つの単語を入力中に，繰り返し呼び出すことが可能である．例えば,「サザンオールスターズ」を入力するときに,「さざんー」で候補一覧を見た後,「おーるー」でさらに絞り込まれた候補一覧を見て，最後に「すたーず」と言って確定できる．なお，この例に示したように，単語中の長母音 (「おーる」の /o-/) では音声補完が呼び出されず，意図的に有声休止した箇所でのみ呼び出されるようにする必要がある．

4 実現方法

3 章のインタフェースを構築するには，有声休止開始点を決定する有声休止検出部と，単語の途中までの発声を認識して補完候補を作成する音声認識部を実現しなければならない．さらに，インタフェース全体の状態管理をするインタフェース管理部と，補完候補一覧や認識結果を提示する画面表示部も必要となる．以下，これらを順に説明する．

4.1 有声休止検出部

音声補完では，有声休止を高い精度でリアルタイムに検出することが重要である．しかも，任意の単語中の母音の引き延ばしを検出する必要があるため，トップダウン情報を使わない言語非依存な検出をしなければならない．そのような要件を満たす検出手法として，我々が文献 4),5) で提案した，有声休止箇所のリアルタイム検出手法を用いる．本手法は，自由発声音声の対話コーパスに対して，再現率 0.76，適合率 0.70 の性能を持つ[6]．再現率と適合率のトレードオフは調整でき，今回の目的に合うように適合率を高く設定することも可能である．

本手法は，有声休止が持つ二つの音響的特徴 (基本周波数の変動が小さい，スペクトル包絡の変形が小さい) をボトムアップな信号処理によって検出する．以下では，音響的特徴の推定方法を示した後に，有声休止開始点を決定する方法を述べる．この結果は，次の音声認識部へと出力される．

4.1.1 基本周波数の推定

入力信号中で最も優勢な高調波構造の基本周波数を，音声の基本周波数として推定する．そのために，時刻 t [☆2] において周波数 F が基本周波数となる可能性 $P_{F0}(F,t) = \int_{-\infty}^{\infty} p(x;F)\ \Psi_p(x,t)\ dx$ を評価する．$p(x;F)$ は基本周波数が F の高調波成分を通過させるフィルタ関数，$\Psi_p(x,t)$ は周波数成分のパワー分布関数である．$P_{F0}(F,t)$ は各高調波構造が相対的にどれくらい優勢かを表すため，基本周波数 $F_{F0}(t)$ は $F_{F0}(t) = \mathrm{argmax}_F\ P_{F0}(F,t)$ で求まる．

☆2 現在の実装では，16 kHz / 16 bit で A/D 変換し，フレームシフト 10 msec (160 点) をすべての処理の時間単位とする．

4.1.2 スペクトル包絡の推定

実環境でロバストに包絡を推定するために，$F_{F0}(t)$ の高調波構造上にある局所的な情報だけを利用する．まず，$F_{F0}(t)$ の整数倍の周波数を中心とするガウス分布で重み付けしながら，その近傍の最大パワーを検出することで，各高調波成分のパワーを求める．次に，隣接する成分のパワーの間を直線補間してスペクトル包絡を求める．有声休止を検出するためには，包絡の大局的な変形を捉えた方が良いため，包絡を粗い周波数分解能でリサンプリングし，低い方から n $(1 \leq n \leq \mathrm{N_{max}})$ 点目の周波数におけるスペクトル包絡 $Env(n,t)$ を求める．最後に，呼気によるAM変調の影響を除去するために $Env(n,t)$ を正規化する．

4.1.3 有声休止開始点の決定

有声休止を検出するための二つの特徴量として，基本周波数の変動量 $A_f(t)$ とスペクトル包絡の変形量 $A_s(t)$ を求める．これらは，$F_{F0}(t)$ と $Env(n,t)$ の過去一定期間の対数スケール上での変化を，最小自乗法で直線近似した直線の傾き $b_f(n)$, $b_s(n)$ と近似誤差 $err_f(n)$, $err_s(n)$ を用いて，$A_f(t) = |b_{F0}|$，$A_s(t) = \left(\frac{1}{\mathrm{N_{max}}}\sum_{n=1}^{\mathrm{N_{max}}} b_s(n)^2\right)\left(\frac{1}{\mathrm{N_{max}}}\sum_{n=1}^{\mathrm{N_{max}}} err_s(n)^2\right)$ のように定義される．そして，有声休止らしさ(有声休止と判定する信頼度) $P_{fp}(t)$ を，$A_f(t)$，$A_s(t)$ の短時間平均 $S_f(t)$，$S_s(t)$ に基づいて，$P_{fp}(t) = \exp\left(-\frac{(\mathrm{R}\,S_f(t) + (1-\mathrm{R})\,S_s(t))^2}{\mathrm{W}^2}\right)$ と定義する．R は特徴に対する重み付けを決める定数，W は変動・変形の考慮範囲を決める定数である．

最終的に，$P_{fp}(t)$ が十分高い値をとり続けるときに，話者が有声休止をおこなっていると判定する．具体的には，条件 $P_{fp}(t) > e^{-1}$ を満たし続ける限り $P_{fp}(t)$ を累積加算して累積値を求め，それが一定の閾値より大きくなった時刻を有声休止開始点とする．この閾値を十分高く設定することで，今回の目的に合った誤検出(false alarm)の少ない検出が実現できる．

4.2 音声認識部(補完候補作成)

音声認識部では，音声入力と有声休止検出部の結果を受け取り，有声休止開始点において音声補完候補の一覧を作成する．ここでは，連続音声認識システム niNja[7] を拡張して実現した．以下に説明するシステムは，単語発声を補完するシステムであるが，連続音声中の単語を補完することも同じ枠組で可能である．

本システムは単語辞書として，入力対象の単語辞書(人名等)以外に，インタフェース操作用語の単語辞書，つまり候補番号(数字)と他候補の表示指示(「次の候補」等)の認識用辞書を使用する．その際，単語ごとに音声補完を可能とするかどうかを指定することで，補完が不要な単語に対する誤動作を未然に防ぐことができる．

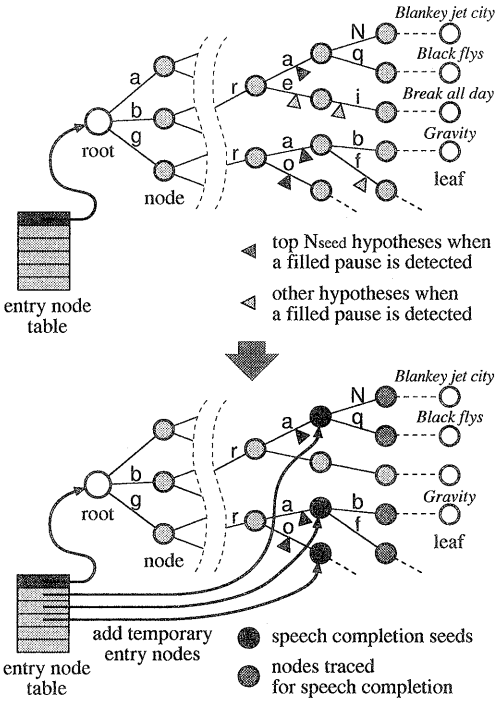


図 2: 木構造の単語辞書における有声休止開始点での仮説(上段)と音声補完候補の作成・エントリノードテーブルへの追加(下段)

音声認識部は，以下の2種類の結果をインタフェース管理部へと出力する．

- 入力対象の単語あるいはインタフェース操作用語の認識結果

  無声区間で区切られた発話単位ごとに認識処理をおこない，発話終了時点で，尤度の高いものから上位 $\mathrm{N_{result}}$ 個の認識結果を出力する．

- 音声補完候補

  有声休止開始点で生成する補完候補を尤度の高いものから番号付けし，上位 $\mathrm{N_{choice}}$ 個を出力する．この出力は発話終了を待たずにおこなう．各候補には，音素列上でどこまで発声されたかという情報も付与される．

単語辞書は，図 2のように木構造で保持される．この辞書を用いた認識処理では，辞書の根から，フレーム同期で枝別れに応じて仮説を増やして，ノードを

葉の方向へたどっていく．図中のくさび形のマークが仮説をあらわす．これらの仮説は一定数以上にならないように，フレームごとに枝刈りされる．

有声休止開始点になると，その時点で有効な仮説から補完候補を生成する．辞書が木構造で表現されているため，補完に必要な情報は，その時点での仮説に対応するノードからたどれる候補をすべて生成することで得られる．このたどり始めるノードを補完候補のシードと呼ぶ．例えば，図2下段の一番上の黒い丸がシードであるとすると，補完候補は “Blankey jet city” と “Black flys” になる．現在の実装では，尤度の高い順に上位 N_{seed} 個のシードを選び，それらの尤度の高いものから，合計 N_{choice} 個を上限として補完候補を作成する．同時に，そこまでに認識した音素列を求めることにより，各候補においてどこまで発声されたかを調べる．なお，図2上段の “Break all day” につながるパス上の /e/ と /i/ の部分の仮説のように，異なるノードから同一の候補が作成されることがある．もしそれらがシードに選ばれた場合には，重複する補完候補を削除する．

単語の続きを言っても選択できるようにするためには，単語の途中からの認識を可能にする必要がある．そのために，認識を開始する根を登録するエントリノードテーブルを導入する．通常の単語の頭からの認識では，図2上段のように，このテーブルには辞書の根だけが登録されている．単語の途中から認識を開始したい場合には，図2下段のように，補完候補のシードを根としてこのテーブルに一時的に追加する．これらの追加エントリは，有声休止を伴う発話の次の発話まで有効とする．追加エントリは，有声休止後の続きの音素列だけを言えば認識されるが，認識結果としては，その単語全体をインタフェース管理部へと出力する．

4.3 インタフェース管理部・画面表示部

音声補完候補の選択等のインタフェース全体としての機能は，インタフェース管理部と画面表示部が提供する．これらは密接に連携して動作するため，以下ではまとめて説明する．

基本的な動作として，まず，有声休止を含まない通常発話の場合には，入力対象の認識結果を受け取って，尤度の相対的な大きさに比例した棒グラフ付きで結果の一覧を表示する．一方，有声休止をおこなった場合には，音声補完候補を受け取り，ポップアップ式の補完候補ウィンドウを出現させて，ウィンドウ内に候補一覧を表示する．その際，音素列上で既に発声された部分の色を変えて各候補を表示する．候補の個数が，ウィンドウ内に表示可能な上限 (N_{window} 個) を越えた場合には，「次の候補」のマークも表示する．そして，3章の操作の流れに従いながら，次の発話の認識結果に応じて以下のように動作する．

1. 他候補の表示指示の場合

   次の N_{window} 個の候補を表示する．
2. 表示中の候補番号・単語と同一の場合

   選択候補を強調表示し，認識結果として確定する．
3. 上記以外の場合

   補完候補ウィンドウを消去し，新たな発話として処理する．

ただし，音声認識部の認識結果としては，上位 N_{result} 個を受け取るため，何位の結果を用いるべきかを検討する必要がある．通常発話では，単に1位の結果を用いれば良いが，音声補完候補が表示されているときの発話では，上記の 1. と 2. の選択操作に該当する結果を，尤度が多少低くても優先すべきである．そこで，上位 $N_{priority}$ 個以内に選択操作に該当する結果があった場合には，それを採用する．単語の続きを言って選択する場合，その発声が単語辞書上の別の単語と近いとそちらの尤度が高くなってしまうことがあるため，上記の優先処理は特に重要となる．

5 実装と結果

以上述べてきた音声補完機能付き音声入力インタフェースのプロトタイプシステムを実装した．日本のポピュラー音楽のヒットチャートから，曲名とアーティスト名のデータベースを作成し，音声補完対象の単語辞書とした．本実装では，4.2 節の各定数を $N_{result} = 5$，$N_{choice} = 20$，$N_{seed} = 5$，$N_{window} = 5$，$N_{priority} = 3$ と設定した．これらは単語辞書の内容や規模，インタフェースの用途等に応じて調整する必要がある．

効果的に負荷分散が可能で，拡張性が高くなるように，本システムを構成する図3の8つの機能を，分

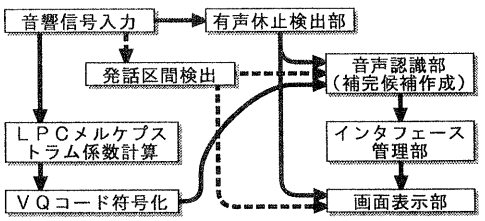
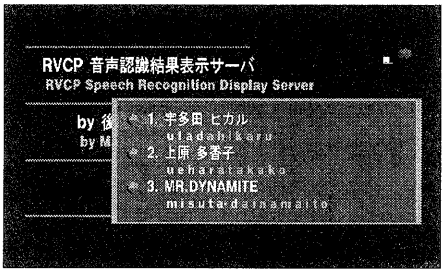


図 3: システムを構成する8つのプロセス

(1)「うただー」と入力

(2)「だー」の有声休止中



(3) 補完候補ウィンドウが回転しながら登場

(4)「1番」と入力した直後

(5) 1番の候補が輝いて跳躍

(6) 1番の候補「宇多田ヒカル」が認識結果として確定

図 4: 音声補完中の画面表示例

散環境で動作する別々のプロセスとして実装した．そのために，音声言語情報をネットワーク上で効率よく共有することを可能にするネットワークプロトコル RVCP (Remote Voice Control Protocol) を設計し，それに基づいて実装した．RVCPは，RMCP (Remote Music Control Protocol)[8] を音声言語情報の伝送用に拡張したプロトコルである．

音声補完中の画面表示例を 図 4 に示す．本システムを運用した結果，提案したインタフェースが実用的に機能し，ユーザが音声補完機能を呼び出しながら，インタラクティブに単語入力できることを確認した．音声補完機能は，使用するのが容易で訓練は不要であり，直感的で使いやすいインタフェースであることがわかった．特に，長い固有名詞を入力する際に，音声補完は非常に有効であった．

6 おわりに

本稿では，発話された単語断片の続きを補うことでユーザの音声入力を手助けする「音声補完」という新しい音声インタフェース機能を提案した．有声休止を補完トリガーキーとすることで，ユーザが自分の意志で，労力をかけずに自由に補完機能を呼び出せ，補完して欲しくないときには一切干渉することがない実用的なインタフェースを実現できた．実際に，インタラクティブに音声入力可能なシステムを実装し，曲名とアーティスト名の入力で有用性を確認したが，これは住所入力や各種固有名詞の入力等の様々な局面にもすぐに適用できる．今後，音声入力インタフェースを構築する上で，音声補完は不可欠な機能の一つになることが予想される．

最後に今後の展望を述べる．従来の音声インタフェースでは，音声認識を中心とした言語情報のモダリティーが主に利用されてきた．それに対して我々の音声補完では，有声休止のような，音声中に含まれる言語情報以外のモダリティーを積極的に利用して，より使いやすいインタフェースを実現した．これは，音声音響信号中の複数のモダリティーを活用した，一種のマルチモーダルインタフェースであると我々は捉えている．今後は，音声補完の対象を単語よりも長い単位に拡張していくと共に，音声補完を発端とした新たな音声インターフェース研究の方向性を探求していきたいと考えている．

謝 辞

連続音声認識システム niNja の改良や単語辞書作成等でご協力頂いた，秋葉 友良 氏に感謝する．

参 考 文 献

[1] 井田昌之, 亀井信義: Emacs 解剖学 入力の補完, *bit*, Vol. 29, No. 2, pp. 85–95 (1997).

[2] 福島俊一, 山田洋志: 予測ペン入力インタフェースとその手書き操作削減効果, 情処学論, Vol. 37, No. 1, pp. 23–30 (1996).

[3] Masui, T.: An Efficient Text Input Method for Pen-based Computers, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'98)*, pp. 328–335 (1998).

[4] 後藤真孝, 伊藤克亘, 速水悟: 自然発話中の言い淀み箇所のリアルタイム検出システム, 情処研報 音声言語情報処理 99-SLP-27-2, pp. 9–16 (1999).

[5] Goto, M., Itou, K. and Hayamizu, S.: A Real-time Filled Pause Detection System for Spontaneous Speech Recognition, *Proc. of Eurospeech '99*, pp. 227–230 (1999).

[6] 後藤真孝, 伊藤克亘, 速水悟: 有声休止箇所のリアルタイム検出システムの評価, 音講論集 春季 3-8-8 (2000).

[7] 伊藤克亘, 速水悟, 田中穂積: 音素文脈依存モデルと高速な探索手法を用いた連続音声認識, 信学論 (D-II), Vol. J75-D-II, No. 6, pp. 1023–1030 (1992).

[8] 後藤真孝, 根山亮, 村岡洋一: RMCP: 遠隔音楽制御用プロトコルを中心とした音楽情報処理, 情処学論, Vol. 40, No. 3, pp. 1335–1345 (1999).