修士学位論文：

韻律情報を利用した
連続音声認識に関する研究

～ピッチパタン連続整合セグメンテーションの高精度化～

東北大学大学院工学研究科
情報工学専攻
中井 満

目 次

図 目 次

表 目 次

第１章

序論

1.1 研究の背景

近年、各種のOA機器の普及、およびニューメディアの急伸に伴ない、より高度なマンマシンインターフェースが望まれるようになった。特に、人間と機械のより自然な、より知的な対話形態として、キーボード等による文字入力システムよりも、マイク等による音声入力システムへの期待が非常に高まっている。そのような対話の実現に向けて、音声認識の技術は日々進歩し続け、現在では離散的な単語の認識装置が製品化の段階に至り、簡単な(あるいは事務的な)対話を可能にした。しかし、人間側に課せられた発声の制約(例えば、発話速度、継続時間、語彙数など)はまだまだ厳しく、いまだ、知的水準には至っていない。

このような背景により、**連続音声**の、とりわけ話し言葉のような**自然発声** (spontaneous speech) を対象にした音声認識の研究に重点が置かれるようになった。しかし、連続音声の認識は明確に発声された単語を認識する場合に比べて、その構造の複雑さから非常に困難であることは明らかである。そのため、これまでの単語認識等で有効であった離散的な言語情報であるホルマント等の音韻的特徴(**音韻情報**)に加えて、話者の感情的な側面を表わすイントネーション、アクセントなどのような韻律的特徴(**韻律情報**)についての検討が必要とされ、近年、数多くの研究が報告されている。2、3の例を挙げると文献[1]では韻律情報の中でも特にアクセントやポーズが音声の了解性に大きく貢献していることを報告している。また、文献[2]では発声時の感情(怒り、歓喜、悲哀)が音声の**ピッチ**(音声波形の基本周波数)、時間、振幅の各構造によってその相違を表現することができることを報告している。このようにあらゆる環境下で発声された会話音声は、その違いを韻律的構造により区別することが可能とされている。したがって、韻律情報の研究は、機械に感情を理解

させたり、あらゆる環境下の音声を認識させたりすることを可能にする上で必要不可欠とされている。

さて、先に連続音声の認識が単語音声等の認識に比べて明らかに困難であると述べたが、その主な要因として、

1. 単語や文節の境界 (gt 句境界) が明確でない、
2. 単語境界付近の音が前後の単語の影響により変形する (**調音結合**がおこる)、
3. 自然な発声速度は、離散発声に比べてかなり速く、個々の単語を構成する各音素の継続時間も短くなり、その発声自体もあいまいになりやすい、

ということが挙げられる。特に 1 の問題は**構文推定** [3] による意味的な解釈、単語や文節などによる小規模単位での認識のための前処理として重要視され、単語や文節、さらに小さいものでは**音素**を単位とした境界検出に関する研究が数多く報告されている [4][5]。句境界検出のアプローチにはさまざまな手法が提案されているが、中でも韻律情報を利用した研究例では、アクセントに関する特徴が**ピッチパタン** (音声の基本周波数の時間変化) として明確に現れる [6] ことに着目して、**アクセントパタン** (1 個のアクセント型) を単位とした句境界の検出が試みられている [7][8][9]。

1.2 研究の目的

連続音声の理解の第一歩は文音声の構造を理解することである。そのためには、まず発声文章を認識の基本単位に切り分ける操作 (**セグメンテーション**) が必要であることは前節で述べた通りである。これまでに発表された韻律情報を用いた句境界の自動検出法は、大きく分けて 2 通りある。ひとつは直接的な検出法で、鈴木ら [7] の手法がその代表的な例である。ここではアクセントパタンの局所的な特徴 (ピッチパタンの谷間、ピッチパタンの局所変化、境界の時間的な間隔) に着目し、総合的な推定を行なっている。もうひとつは間接的な検出法で、下平ら [8] の手法が挙げられる。ここでは予め準備された擬似アクセントパタンを参照して全発話区間のピッチパタンからアクセントパタンを抽出し、その結果、抽出境界位置から句境界を推定している。この手法は前者の手法に比べて厳密なパラメータ推定等の必要がなく学習が容易で、また音響分析等の異なるデータセットの影響を受けにくい。

そのような理由から、本研究では句境界検出の高精度化にあたり、下平らによって提案されているピッチパタンの連続整合による手法を用いる。また、本論文では、その句境界検出法の問題点を指摘、改善することにより、その高精度化を図ることを目的とする。

ここで、高精度化の具体的な内容として、

1. 句境界検出率の向上

2. 句境界挿入誤りの減少

3. 実時間処理に適した高速化

等が挙げられる。1 の項目と 2 の項目は互いに相関があって、本検出法では、自動検出によって得られる境界の数が増えれば句境界検出率は上昇し、同時に句境界挿入誤りも上昇する。したがって両方を同時に満たすのは至難であり、本論文では、 1 の項目に重点を置く。その理由は、挿入誤りは事後処理で訂正可能であるが、未検出句境界は検出されないままとり残されるからである。また 3 の項目については、**One-Stage DP 法** (第2章後述) により達成されるので、以後の高精度化はこのアルゴリズムの実現範囲内で図られることにする。

1.3 本研究に用いる音声資料

本論文中における全ての句境界検出実験は、ATR の提供する日本語音声データベース (連続音声データベース) を用いて行なわれる。本データベースは、新聞、雑誌、小説、手紙、教科書の文献から無作為に抽出した約一万の文をもとに、音素環境のバランスを考慮して作成した 503 文から構成されている [10]。男性、女性それぞれ 5 名計 10 名のナレータの発話について、音声セグメント・ラベル及び韻律情報、言語情報が作成されていて [11]、本研究では男性話者 1 名 (MYI) のみ入手している。韻律情報としては、韻律の研究を進める上で必要な声帯の基本周波数値 (ピッチ) を発話中の全有声音区間に対し、2.5 ミリ秒ご

表 1.1: ATR 連続音声資料

音声データ	12kHz サンプリング、16bit
分析窓	ブラックマンウインドウ
分析窓長	384 ポイント (32.0msec)
分析シフト	30 ポイント (2.5msec)
FFT ポイント	512 ポイント (42.7msec)
ピッチ探索範囲	男性話者 50～350 ポイント (34Hz～240Hz) 女性話者 24～120 ポイント (100Hz～500Hz)

とにケプストラム法で自動抽出し、誤抽出個所については視察によって修正したものが用意されている。また、聴取により、アクセント句の区切りとアクセント核の位置も用意されている。分析条件の詳細は表 1.1のとおりである。

1.4 本論文の構成

本論文は 6 章で構成され、**第 1 章**は序論である。

第 2 章は句境界検出に関する研究の現状、および、従来のパタン連続整合法のアルゴリズム、特徴等についてまとめ、その問題点を指摘する。

第 3〜5 章はその問題点について解決策を与え、高精度な句境界検出を実現させていく。まず、**第 3 章**では句境界検出の対象となるピッチパタンについて考察し、問題点を除去する。続いて、**第 4 章**ではアクセントパタンの類似性について考察し、パタンの整合アルゴリズムを改良する。そして、最後の**第 5 章**ではこれまで固定時間長 (単一時間長) であった参照アクセントパタンを、複数時間長に拡張して句境界検出法をまとめる。

第 6 章は結論である。

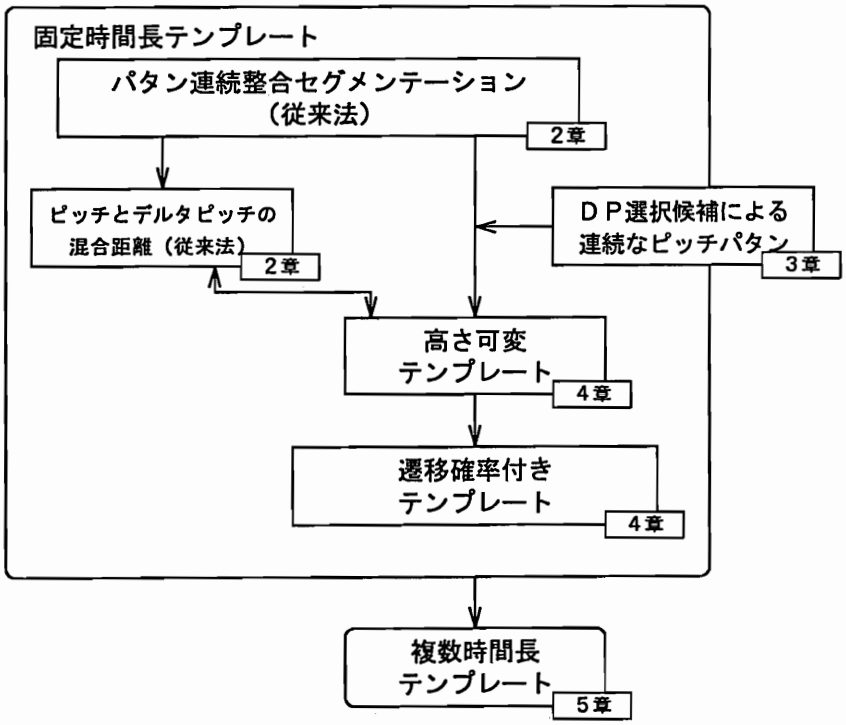


図 1.1: 本論文に提案される句境界検出法

第 2 章
パタン連続整合法を用いた句境界検出とその問題点

2.1 はじめに ～句境界検出の現状について～

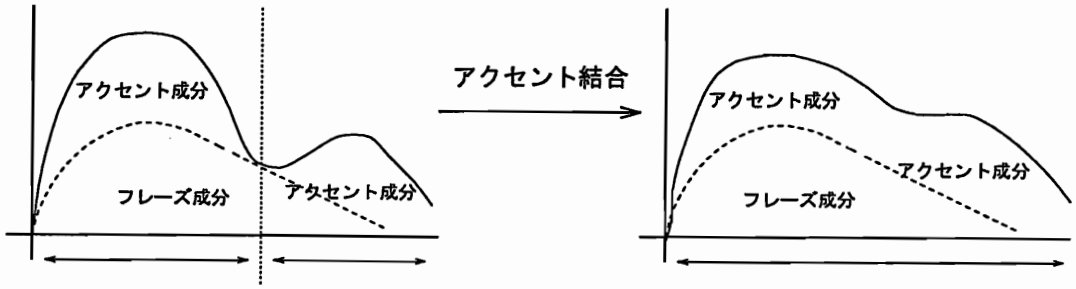
連続音声の発話全区間を直接モデル化してパタン認識することは非常に困難であり、認識性能上でも、また演算処理上でも多くの問題がある。そのため、文音声を文法的にあるいは意味的にまとまりのある小さな単位に分割 (セグメンテーション) し、重要語に対して単語レベルの認識処理を行うことが有効であると考えられ、これにより、文音声も単語音声と同程度の認識性能を達成できると考えられている。

このような理由から文音声の意味的な境界 (句境界) の検出はその重要性を帯び、数多くの研究成果が報告されている。その検出法はさまざまであり、それぞれの目的もまた異なっている。例えば、認識精度が十分高く、文脈等の解析に重点が置かれている研究では、予めフレーム単位で音韻 (あるいは音素) の認識を行い、その出力シンボルから境界位置を推定する手法も提案されている [12]。また、音素や単語レベルの認識に境界情報を付加することを目的として、前処理的に韻律情報 (アクセント、イントネーション、ポーズ等) から句境界を推定するという研究も試みられている [7][8]。

文献 [7] では句境界候補の抽出基準として、基本周波数の谷間、基本周波数の局所変化、振幅包絡の谷間の勾配、無音区間の長さ、擬似点ピッチパターンの極小点、前後の句境界からの距離の 6 つを挙げ、成人男性に対して個人性とは無関係に約 80%の句境界を確実に検出できることを報告している。

また、文献 [8] では、平叙文のアクセントパタンが “へ” の字型を成していることを利用して、クラスタリング分類により代表的なアクセントパタンを作成し、パタン認識の手法を用いてピッチパタンと整合することにより句境界を検出している。

本章では、後者の手法について追実験を行い、その問題点を指摘することを目的とする。



例、「兄の」＋「雨具を」　　例、「姉の」＋「絵具を」

図 2.1: ピッチパタンのモデル

2.2　連続音声の韻律的単位

境界を定める基本単位としては、古くから単語や文節が用いられているが、具体的な発話の単位として、これが必ずしも妥当ではないことはすでに指摘されている [13]。これは文の**統語構造**と**韻律構造**が異なる次元のものであるためである。そこで、藤崎らは文献 [14] の中で、韻律構造について次のようにまとめている。

- **韻律語** (prosodic word)

  “1 個のアクセント型を構成する発話の一部又は全体”

- **韻律句** (prosodic phrase)

  “アクセント成分の結合により形成される 1 個のアクセント型”

- **韻律節** (prosodic clause)

  “長い文において、休止によって区切られる区間”

- **文音声**

  “発話された文全体”

一般に 1 個のアクセント核を持ったパタン (**アクセント型、アクセントパタン**) は韻律語または韻律語のアクセント結合によって形成される韻律句の形となりピッチパタン上に明確に現れる。図 2.1のようにピッチパタンは文頭から文末に向かう緩やかな下降に対応する**フレーズ成分**と、局所的な起伏に対応する**アクセント成分**との和で表現され、そのモデル

表 2.1: 統語構造上と韻律構造上の句境界

		プロペラ	機	で	ふわり	ふわり	と	地球	を	一周	した
統語構造	形態素境界										
	文節境界										
韻律構造	アクセント境界										

$\ln F_0$(対数基本周波数) は時刻 t の関数として

$$\ln F_0 = \ln F_{\min} + \sum_{i=1}^{I} A_{p_i} G_{p_i}(t - T_{0i}) + \sum_{j=1}^{J} A_{a_j} \left\{ G_{a_j}(t - T_{1j}) - G_{a_j}(t - T_{2j}) \right\} \qquad (2.1)$$

により与えられる。ここで $F_{\min}$は声帯振動が可能な最低周波数、I, Jは一文中でのフレーズ数およびアクセント数、A_{p_i}, A_{a_j}は i 番目のフレーズおよび j番目のアクセントの大きさ、T_{0i}は i 番目のフレーズの開始点、T_{1j}, T_{2j}は j番目のアクセントの開始点び終了点である。また $G_{p_i}(t), G_{a_j}(t)$ はそれぞれフレーズ制御機構のインパルス応答関数、アクセント制御機構のステップ応答関数であり、α_i, β_jをそれぞれの固有角周波数とすれば

$$G_{p_i}(t) = \alpha_i t e^{-\alpha_i t} \qquad (2.2)$$

$$G_{a_j}(t) = \min[1 - (1 + \beta_j t) e^{-\beta_j t}, \theta] \qquad (2.3)$$

である。ただし、$t \leq 0$ ではともに 0 であり、θ は $G_{a_j}(t)$ の上限値 (およそ 0.9) である。このように、ピッチパタンは定式化されており、境界抽出後のピッチパタンのフレーズ成分、アクセント成分の分離は可能であると考えられているが、未知入力パタンに対しては不可能である。しかし (2.1) 式の第 2 項が比較的緩やかな下降として考えられるので、実際には両成分の和をもってしても、1 つのアクセント型は平叙文において “へ” の字型を成すことが多い。それ故にアクセント境界が句境界の基本単位としてよく用いられるのである。表 2.1に統語構造上と韻律構造上の境界位置の違いについてまとめておく。韻律構造は統語構造上の文節と良く似ているが、「ふわりふわりと」のように一文節に 2 つのアクセント核が存在する場合にその違いが見られる。

2.3　パタン連続整合法による句境界検出

2.3.1　テンプレートパタンの学習

前節で述べたように、アクセントパタンはピッチパタン上で “へ” の字型を成すことが多く、セグメンテーションの基本単位としてしばしば用いられる。つまり、アクセントパタ

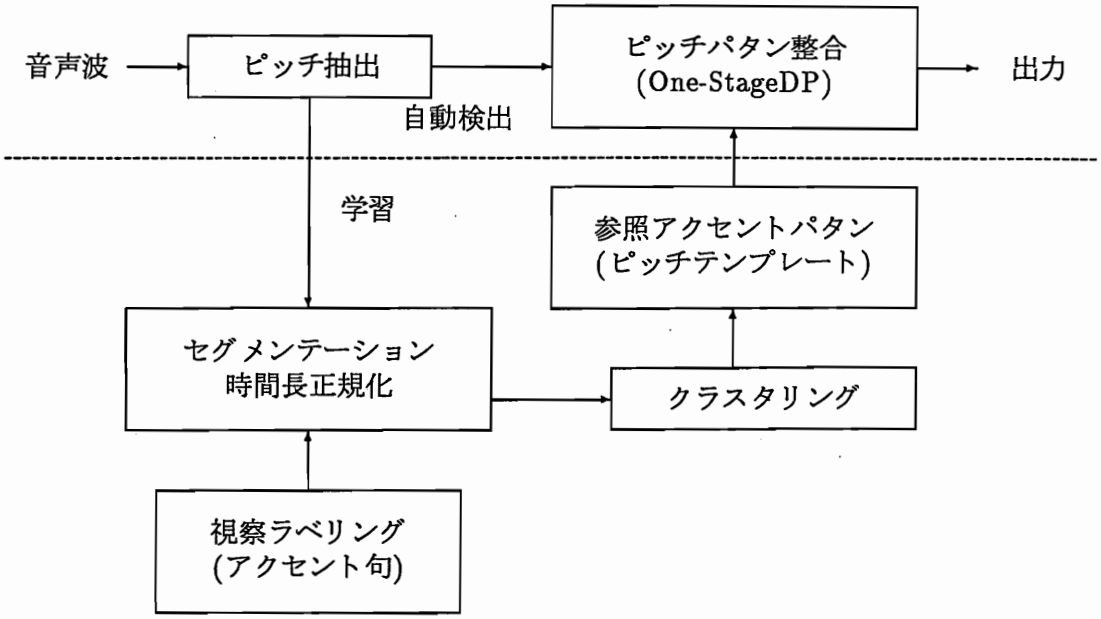


図 2.2: 句境界検出システムの概略

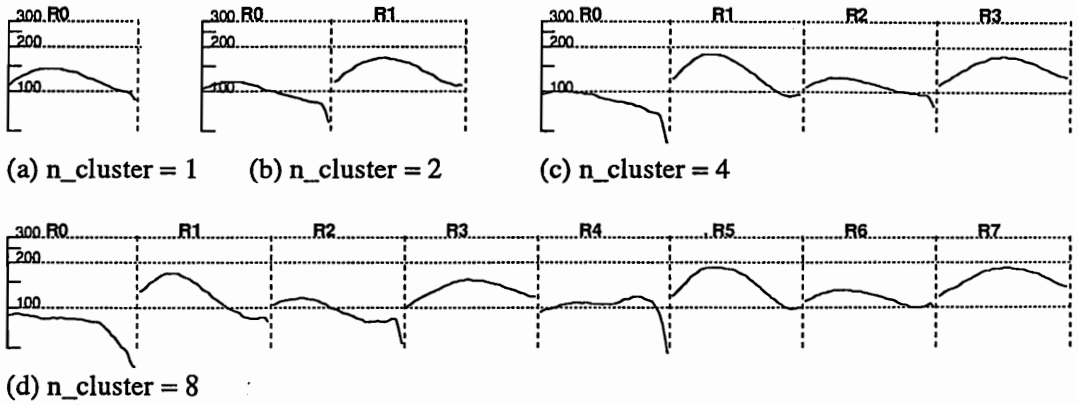
ン抽出のための参照用パタン (以降、**ピッチテプレート**、もしくは単に**テンプレート**) として、アクセントの特徴を表わす“へ” の字型のパタンを作成する必要があり、このパタンは学習サンプルのクラスタングによって、容易に得られる。

学習サンプルは連続音声中から視察、あるいは聴取によって切り出されたアクセントパタンを使用する。クラスタリング分類の基準として、アクセントパタンの形、アクセントパタンの継続時間長など、その類似性について幾つかの要因が挙げられるが、この問題は第5章で触れることにしてここではアクセントパタンの形による分類を行なう。今、2 つのパタン P_1、P_2を例に考える。異なる長さを持つパタン $P_1 = (p_{11}, \cdots, p_{1L_1}), P_2 = (p_{21}, \cdots, p_{2L_2})$ の間の距離をを簡単に定義するため、まずそれぞれを同じ長さ L フレームに線型変換する。このときパタン長 L はDP 整合の際の時間伸縮の制限範囲にもよるが、全アクセントパタンの平均長にするのが妥当である。この変換されたパタン$\hat{P}_1 = (\hat{p}_{11}, \cdots, \hat{p}_{1L}), \hat{P}_2 = (\hat{p}_{21}, \cdots, \hat{p}_{2L})$ のユークリッド距離

$$D(\hat{P}_1, \hat{P}_2) = \sum_{i=1}^{L} (\hat{p}_{1i} - \hat{p}_{2i})^2 \tag{2.4}$$

を 2 つのパタン間の距離として定義する。またクラスタリング分類の方法はクラスタ数を階層的に増加させていく手法である **LBG 法** [15] を用いる。この結果、各クラスタに属するアクセントパタンの平均を計算してピッチテンプレートとする。

図 2.3に見られるように、クラスタ数が 1 のとき、つまりテンプレート数が 1 のとき



(a) n_cluster = 1　(b) n_cluster = 2　(c) n_cluster = 4

(d) n_cluster = 8

図 2.3: アクセントパタンの分類結果

(n_cluster=1) は全アクセントパタンの平均ということになり、最も典型的な形を表す。またクラスタ数 (テンプレート数) を増して行くにつれてそれぞれ最も異なるパタンが得られて行く。いずれもひとつのアクセント核とも言える山を持ち、さまざまなアクセント型 (頭高型、起伏型、平板型など) を表現できるようになることが分かる。

2.3.2　連続整合法

句境界の自動検出の概要は次のようになる。まず、未知の入力連続音声に対してピッチを抽出し、ピッチパタンを作成する。このとき、同時に音声パワーの閾値およびその継続時間より無音区間を検出し、連続整合の対象区間、つまり発声の始点から終点を限定する。パタンの連続整合のイメージは図 2.4のようになる。図中の上段は未知入力音声のピッチパタンであり、下段は前節の学習によって得られたピッチテンプレート (テンプレート数 8) である。また、中段はピッチテンプレートを連続に並べたものである。この中段のピッチテンプレート系列を入力ピッチパタンの時間長に合わせて DP(Dynamic Programming) で非線形に時間軸伸縮することにより、テンプレート系列による擬似的なピッチパタンが生成される。このときの伸縮された個々のテンプレートに相当する区間がアクセントパタンとして認識され、テンプレートの接続境界に相当する時間上の位置がアクセント句境界として検出される。ここであらゆるテンプレートの組み合わせ (同じテンプレートが繰り返し出現する場合も可) のうち DP 整合による距離歪みが最も小さいものを最適なピッチテンプレート系列とする。

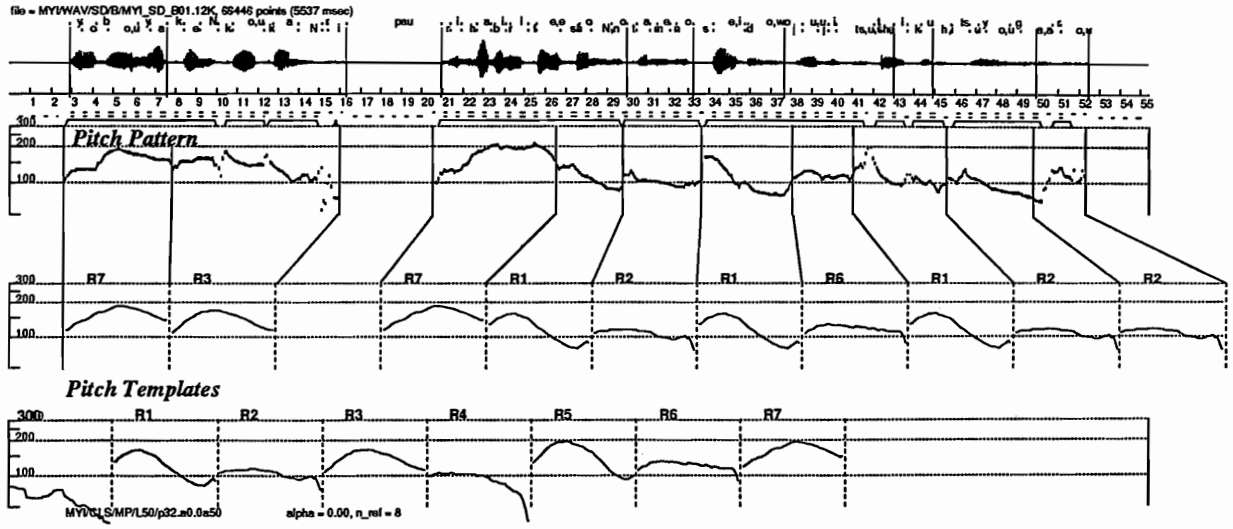


図 2.4: ピッチパタンとテンプレート系列の DP 整合

2.3.3　One-Stage DP 法による高速処理

整合のためのテンプレート系列の組み合わせは数限り無く、全ての組み合わせを試行することは不可能である。この処理を高速に行なうアルゴリズムとして **One-Stage DP 法** [16] があり、本システムではこの手法を用いることにする。これによりピッチテンプレートの最適な系列を実時間に沿って、逐次的に求めることが可能になり、テンプレート数 n に対し、$O(n)$ の時間で処理することが出来る。

以下に DP パスの傾斜制限が 0〜2 の場合、つまり、テンプレートの伸縮を 1/2 から∞まで許可する場合を例に、そのアルゴリズムを記す。

[アルゴリズム]

未知入力パタンのフレーム：$i = 1, \cdots, N$

テンプート番号：$k = 1, \cdots, K$

テンプレート kのフレーム：$j = 1, \cdots, J(k)$

(i, j, k) における累積距離：$D(i, j, k)$

(i, j, k) におけるフレーム間距離：$d(i, j, k)$

対数ピッチ値を $F_i(i)$

テンプレート番号 kにおけるフレーム jの対数ピッチ値を $F_{tk}(j)$

とするとき、次のように定義する。

$$d(i, j, k) = (F_i(i) - F_{tk}(j))^2 \tag{2.5}$$

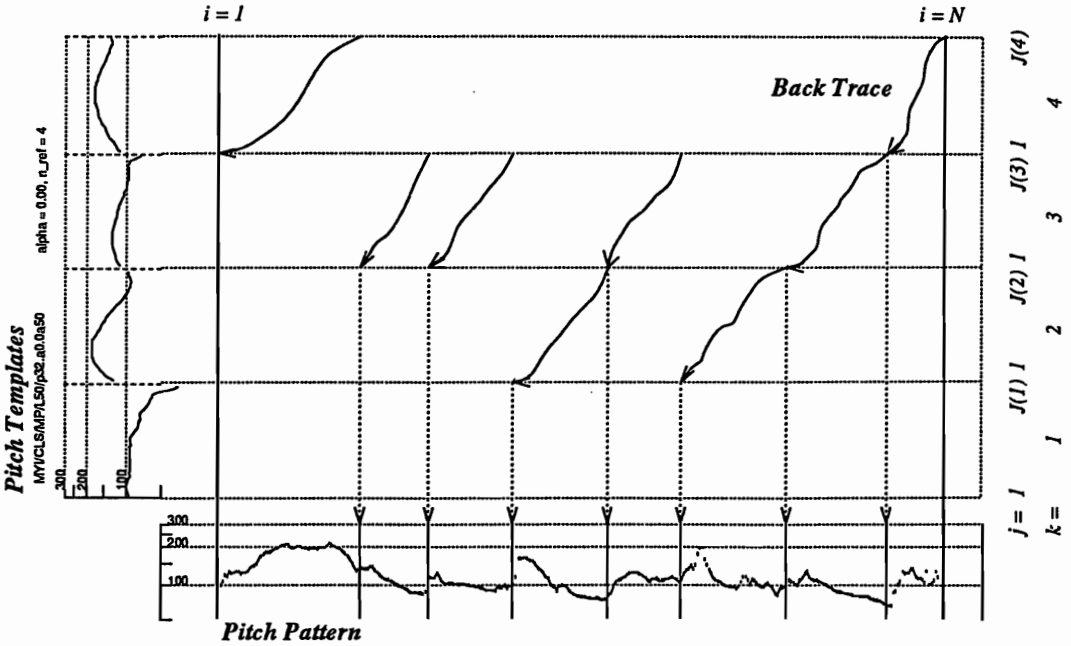


図 2.5: One-Stage DP による高速処理

Step1 initialize $D(1,j,k) = \sum_{n=1}^{j} d(1,n,k)$

Step2 (a) for $i := 2$ to N do steps (b) - (e)

(b) for $k := 1$ to K do steps (c) - (e)

(c)
$$D(i,1,k) = d(i,1,k) + \min \left\{ \begin{array}{l} D(i-1,1,k) \\ D(i-1,J(k^*),k^*), \quad k^* = 1,\cdots,K \end{array} \right\}$$

(d) for $j := 2$ to $J(k)$ do step (e)

(e)
$$D(i,j,k) = d(i,j,k) + \min \left\{ \begin{array}{l} D(i-1,j,k) \\ D(i-1,j-1,k) \\ D(i-1,j-2,k) \end{array} \right\}$$

Step3 Trace back the best path

図 2.5中の下側のパタンはフレーム $i = 1,\cdots,N$ の入力ピッチパタンであり、左側に横付けされた 4 個のパタンはピッチテンプレートである。各テンプレートの始端では **Step 2**

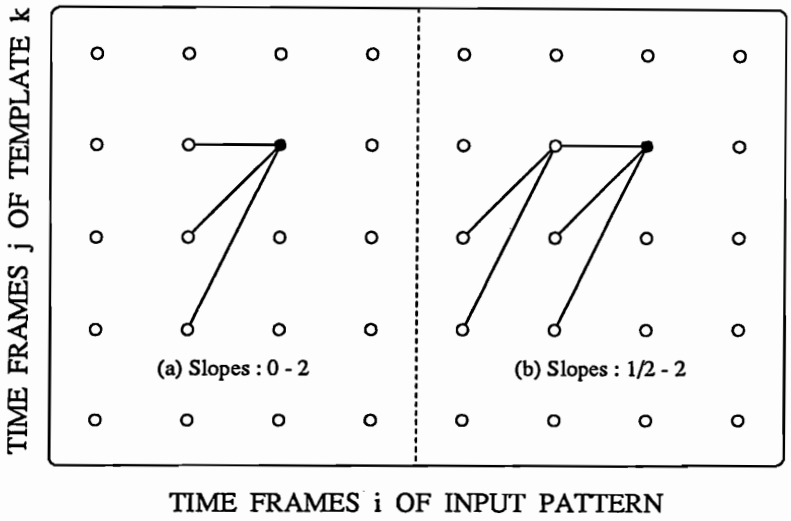


図 2.6: DP パスの傾斜制限

の (c)、それ以外では (e) により累積距離が最小になる経路を求め、最終的に $i = N$で最も累積距離が小さく終端したテンプレートの終点より逆方向に経路をトレースした結果が図中に示されている。この経路に相当するピッチテンプレート系列が最小コストのテンプレート系列であり、経路がテンプレートの接続境界と交差する位置が句境界の位置である。

図 2.6に見られるようなDP **パスの傾斜制限**が、0〜2 の場合は過去 1 フレームの値を記憶するだけでよく、計算量やメモリ量の上で望ましいのであるが、実際には図 2.6の (b) のような 1/2〜2 のものを使うことが多い。これは傾斜を 0 まで許した場合、テンプレートの伸縮が無限まで可能になり、アクセントパタンの形状が失なわれてしまうので、それを抑制するためである。このときアルゴリズム中の **Step 2**(c)(e) の処理が次のように置き変わる。

Step2 (c)

$$D(i,1,k) = d(i,1,k) + \min\left\{ D(i-1,J(k^*),k^*),\ \ k^*=1,\cdots,K \right\}$$

(e)

$$D(i,j,k) = d(i,j,k) + \min\left\{ \begin{array}{l} d(i-1,j,k)+D(i-2,j-1,k) \\ d(i-1,j,k)+D(i-2,j-2,k) \\ D(i-1,j-1,k) \\ D(i-1,j-2,k) \end{array} \right\}$$

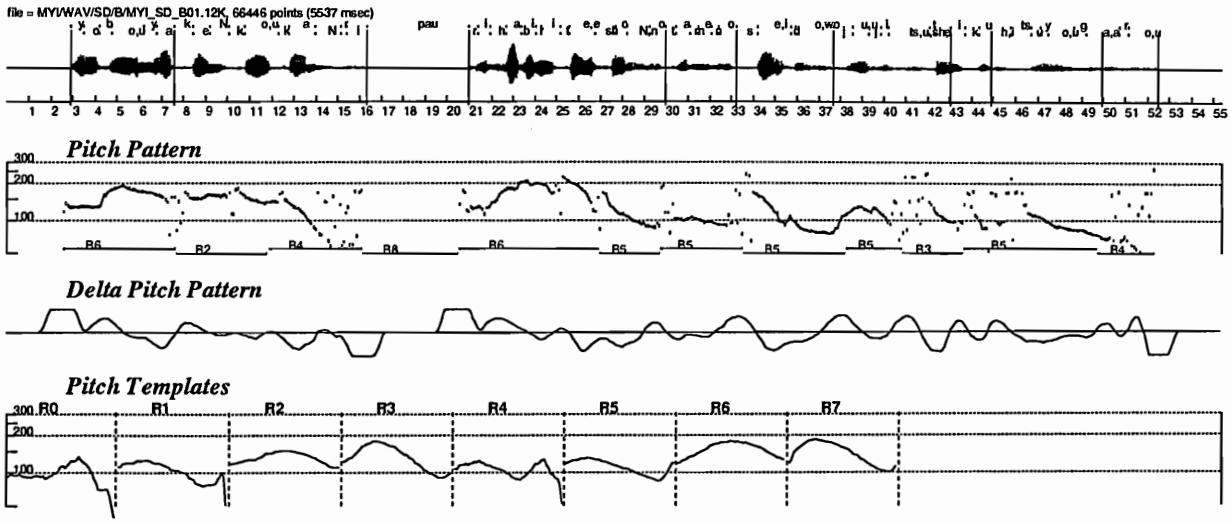


図 2.7: ピッチパタンとデルタピッチパタン

2.3.4　ピッチとデルタピッチの混合距離尺度

前節までが句境界検出の概要であるが、下平ら [8] はパタンマッチングの距離尺度にピッチとピッチの変化度 (**デルタピッチ**) を用い、2 次元のマトリクスパタンによる整合を行っている。これはピッチが 1 点 (分析フレーム) の絶対的な情報しか表さないのに対して、デルタピッチはその近傍の形状を表し、この両者を組み合わせることによって、パタンの全体的な形状をより的確にとらえることができるからである。

分析フレーム i において抽出された対数ピッチ周波数値を p_i とし、そのときの安定度を r_i とするとデルタピッチ Δp_i は以下の式で示すように、p_i に対する重み付けされた最小二乗誤差直線の傾きとして定義される。

$$\min_{\Delta p_i, b} \sum_{k=-M/2}^{M/2} r_{i+k} w_k (p_{i+k} - (\Delta p_i k + b))^2 \tag{2.6}$$

ここで、w_k は分析窓による重み付けで、M はその窓幅を示す。本実験では分析窓として三角窓を用いる。

デルタピッチの導入により、(2.4) 式で定義されたパタン間の距離を以下のように再定義する。いま、視察によって切り出された 2 つのピッチパタン P_1, P_2 のフレーム i における対数ピッチ周波数をそれぞれ p_{1i}, p_{2i} とする。これらのパタンは継続時間長が異なるので、予め定められている長さ L フレームに線型伸縮変換を行なう。変換されたパタンをそれぞれ、$\hat{P}_1 = (\hat{p}_{11}, \cdots, \hat{p}_{1L}), \hat{P}_2 = (\hat{p}_{21}, \cdots, \hat{p}_{2L})$ と記すと、2 つのパタン間の距離は以下のよう

表 2.2: 句境界検出実験条件

音声資料		
	ATR 日本語音声データベース連続音声 (自由発声) 音韻バランス 503 文章 (A ～ I、各 50 文章、J、53 文章) 男性 1 名 (MYI)	
句境界検出実験条件		
学習	データ	C ～ J (403 文章、2695 セグメント)
	テンプレート長	$L = 50$ (500ms)
実験	データ	A ～ B (100 文章、804 セグメント)
	DP パス	1/2～2 の傾斜制限
	デルタピッチ寄与率	$\alpha = 0.0$、0.5、1.0
	ピッチ抽出法	lag-window 法

に定義される。

$$D(\hat{P}_1, \hat{P}_2) = \sum_{i=1}^{L}((1-\alpha)(\hat{p}_{1i} - \hat{p}_{2i})^2 + \alpha(\Delta\hat{p}_{1i} - \Delta\hat{p}_{2i})^2) \tag{2.7}$$

ここで、αはパタン間の距離におけるデルタピッチ距離の寄与率で、$0 \leq \alpha \leq 1$ の値を取る。これに伴ない、(2.5) 式の入力パタンのフレームとテンプレートのフレームとの距離も以下のように再定義する。

$$d(i,j,k) = (1-\alpha)(F_i(i) - F_{tk}(j))^2 + \alpha(\Delta F_i(i) - \Delta F_{tk}(j))^2 \tag{2.8}$$

$F_i(i)$: 入力パタンフレーム i におけるピッチ周波数
$F_{tk}(j)$: テンプレート kのフレーム jにおけるピッチ周波数
$\Delta F_i(i)$: 入力パタンフレーム i におけるデルタピッチ周波数
$\Delta F_{tk}(j)$: テンプレート kのフレーム jにおけるデルタピッチ周波数

2.4　パタン連続整合法による句境界検出実験

2.4.1　実験

句境界検出の実験は表 2.2の条件で行なう。ピッチの抽出は lag-window 法 (第 3章後述) を用い、視察修正を行なわないことにする。音声資料は ATR 日本語音声データベースの

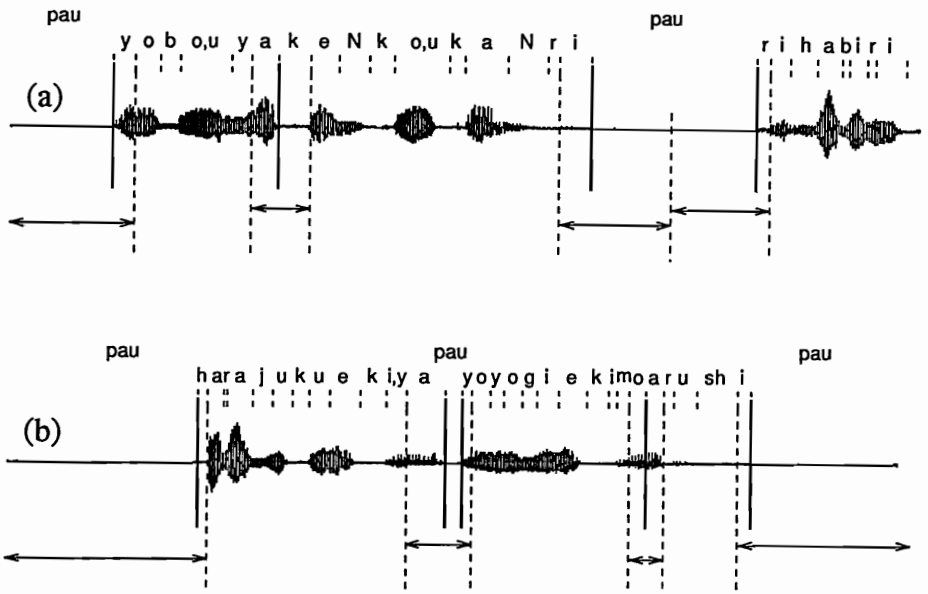


図 2.8: 1 音素基準による正解句境界範囲

男性話者 1 名分、503 文章を用いる。(実験資料の詳細は 1.3節、もしくは文献 [10][11] を参照。) ピッチテンプレートの学習には C グループから J グループまでの 403 文章を用いてテンプレート長 500ms(アクセントパタンの平均長) でクラスタリングを行ない、自動検出実験には残りの 100 文章を用いた。また、DP 整合には時間伸縮によるテンプレートの形状の変化を抑制するため傾斜制限を 1/2〜2 の範囲とした。句境界検出結果は**句境界検出率**と**句境界挿入誤り率**によって評価する。

$$\text{句境界検出率} = \frac{\text{正解検出数}}{\text{視察による句境界の総数}} \tag{2.9}$$

$$\text{句境界挿入誤り率} = \frac{\text{不正解検出数}}{\text{自動検出による句境界の総数}} \tag{2.10}$$

ここで句境界の正解範囲として次の 2 つを考える。

- 視察による句境界の前後 100ms 以内 (**付録 A**)
- 視察による句境界の前後 1 音素以内 (平均：前 89.51ms 後 75.17ms)

前者の規範は文献 [8] に倣ったものである。しかし境界から 100ms 以内のずれを許した場合、実際にはその範囲内に音素が 2〜3 個含まれる可能性があり、語の切り出しとしては不十分である。また、自動検出した境界をもとに確実なワードスポットを行なうために

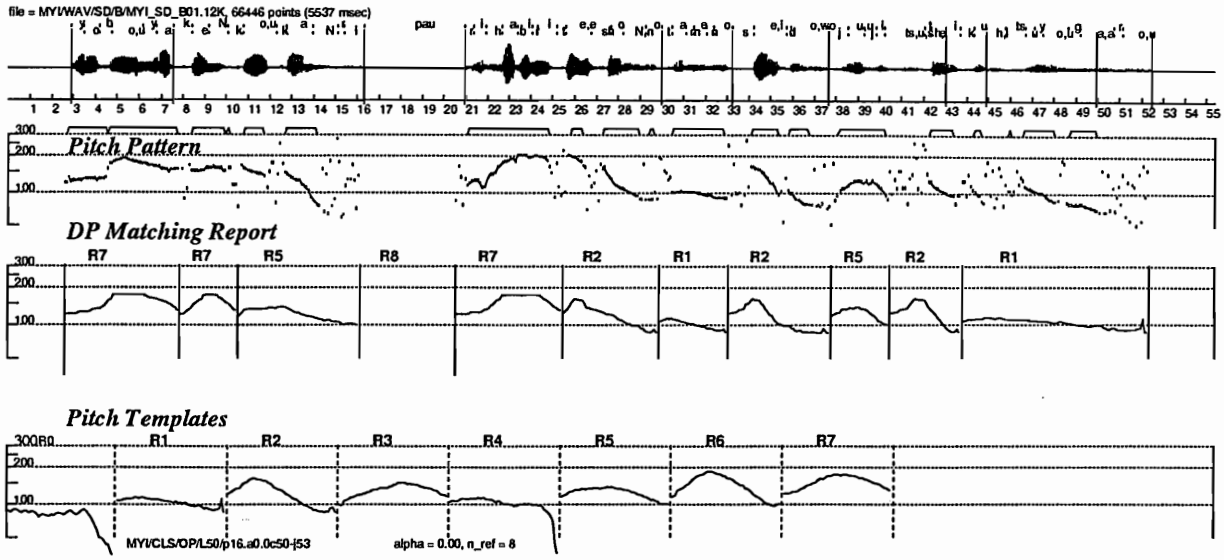


図 2.9: 連続整合セグメンテーションによる句境界検出例

は、100ms 冗長に切り出す必要があり、最悪の場合、片側で 200ms 余計に、両側で 400ms 余計に切り出すことになる。それに比べ、後者の規範における正解領域はその心配が全く無く、自動検出した句境界から数フレーム余分に広く切り出せば十分 1 つの語を包含することができる。したがって、以降の実験では 1 音素規範の検出について考察を行う。また、文献 [8] との比較用に 100ms 規範の句境界検出率を付録として巻末に収録した。(いずれも句境界検出率の傾向は同じであり、1 音素規範の方が条件が厳しくい分だけ、5%程度低くなる。)

図 2.8は 1 音素を基準とした場合の正解領域の例である。矢印で示される範囲が正解領域で、発声の始端、終端も境界としてカウントする。またポーズも入力音声パワーの閾値等により予め検出しておくので、発声の始端、終端と同様に境界として扱う。(a) のようにポーズが長く、明確に検出できる場合は、ポーズ区間の中間で分けて、2 つの境界として扱うが、(b) のようにポーズ区間が非常に短く、前処理でも検出できないような場合には 1 つの句境界として扱うことにする。

2.4.2　実験結果

句境界検出の例を図 2.9に示す。統計的な結果は章末の図 2.10～2.11、表 2.3～2.4 のようになった。一般にデルタピッチを適度に寄与した場合 ($\alpha = 0.5$) の方が検出率が高い。また、テンプレート数を増やすにつれて検出率が上がるが、テンプレート数が 8 を超えると

頭打ちになって下がり始める。デルタピッチのみのときではテンプレート数が少ない場合において他のものより良好な検出率を挙げているが、テンプレート数を増やしてもそれほど変化が現れない。これは典型的なデルタピッチテンプレートが正から始まり、負で終わるパタンとなるので、少ないテンプレート数でも変化を捉え易く、逆にその分ピッチパタンに比べてパタンの種類が乏しく、テンプレート数を増やしてもそれほど効果が現れないものであると考えられる。しかし、それ以前にこの実験ではシステムや条件にいくつかの問題点が見られ、現段階では正当に評価ができないことも指摘される。次節ではこの問題点について考察する。

2.5　パタン連続整合セグメンテーションの問題点

ここで、パタン連続整合法による句境界検出の問題点を 2、3 挙げておく。

まず、自動抽出によるピッチパタンが句境界検出のための韻律的な特徴パタンとして不適当なことである。実験に用いたパタンでは、無声音や微弱な発声のために無声化された音声により、ピッチパタンに不連続区間が生じている。これを修正しない限り、パタン連続整合法では値がとぶことによる距離歪みを抑制しようと、不連続個所が整合境界として選ばれることになる。もし不連続個所の近隣 (DP で整合可能な 1 パタン以内) に正解句境界がある場合、その境界は検出不可能になる。このため、検出法の性能の良し悪しに関わらず、ほぼ同じ個所が句境界として検出されるので、システムの評価が難しくなる。

次に緩やかな傾斜が続く区間ではピッチテンプレートの両端の高さに等しい個所で整合する場合がある。これはピッチテンプレートの各値がそれを代表とするクラスタに属するアクセントパタンの平均値ということもあり、ピッチの高さに対する絶対的な意味を持っているからである。従来法では絶対的な値に加えて、その近傍の形状の変化を表わすデルタピッチを用いたが、いずれも単体では不十分で、2 つの混合距離尺度によって解消している。

最後に、アクセントパタンの形状を保つため DP パスの傾斜制限を 1/2〜2 とすると、テンプレート長 500ms では、250ms 以下のセグメントは検出不可能であるし、又、1000ms 以上のセグメントに対しては挿入誤りが生じる。今回の実験データ中 (グループ A〜B) に、検出不可能と思われるアクセントパタンが約 65 あるので、検出率は最大でも約 93%程度であると考えられる。

2.6　まとめ

この章では文献 [8] で提案されているパタン連続整合セグメンテーションについてその概念、アルゴリズム等をまとめ、追実験を行なった。その結果、システムの性能を下げる原因ともなる 3 つの問題点を指摘した。ひとつは不連続なピッチパタンによる悪影響であり、もうひとつはアクセントパタンの絶対的な高さに関する問題であり、最後のひとつははテンプレートパタン固有の時間長による整合の限界である。以降の章ではこれらの問題に対して対策を立て、それによって句境界検出の高精度化を図っていくことにする。

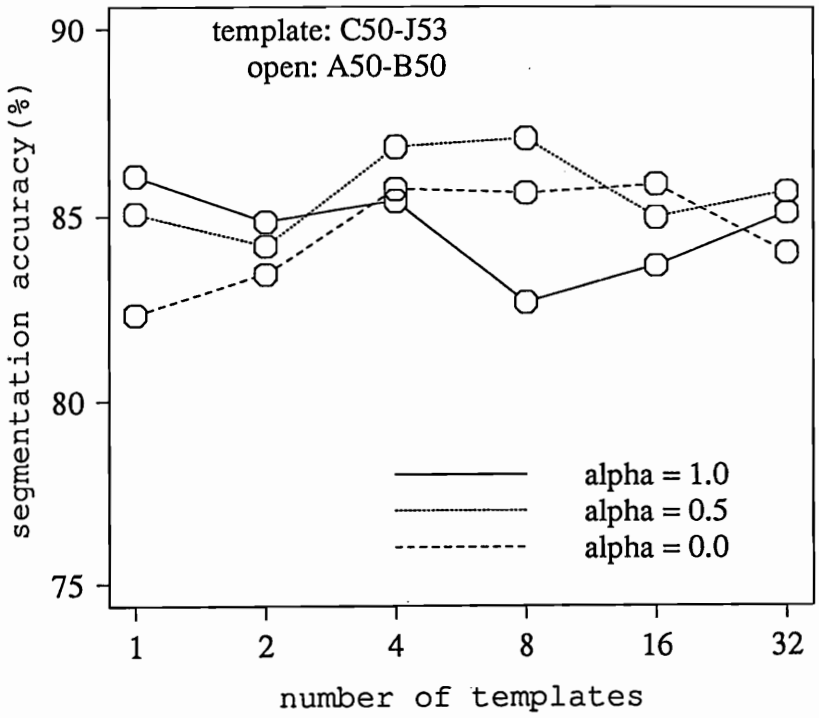


図 2.10: 連続整合セグメンテーションによる句境界検出率

表 2.3: 連続整合セグメンテーションによる句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	82.301	83.407	85.730	85.619	85.841	83.960
0.5	85.066	84.181	86.836	**87.058**	84.956	85.619
1.0	86.062	84.845	85.398	82.633	83.628	85.066

(単位：%)

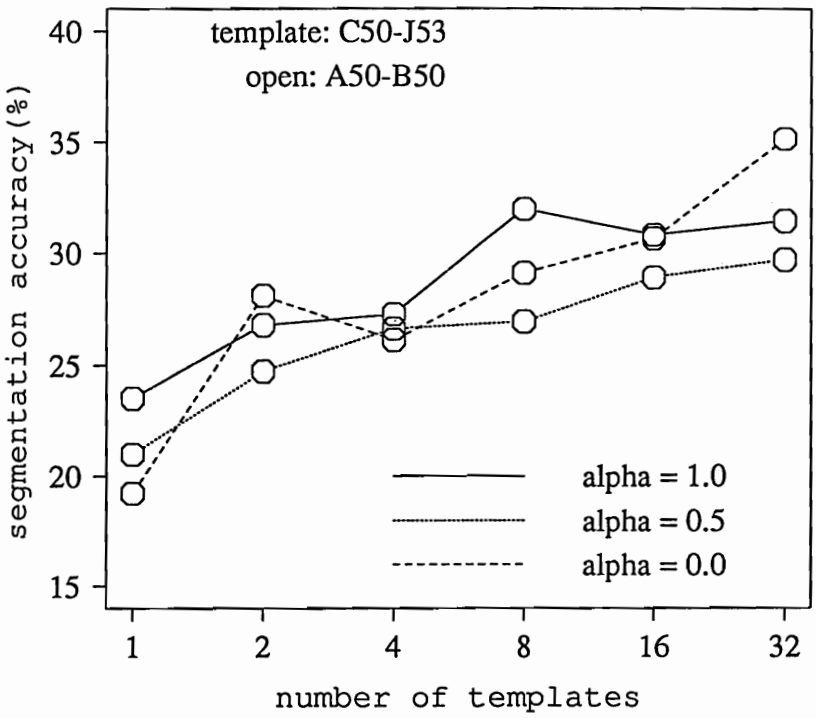


図 2.11: 連続整合セグメンテーションによる句境界挿入誤り率

表 2.4: 連続整合セグメンテーションによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	19.217	28.097	26.131	29.110	30.624	35.081
0.5	20.990	24.713	26.641	26.953	28.893	29.626
1.0	23.511	26.800	27.282	31.962	30.799	31.362

(単位：%)

第 3 章

ピッチパタン検出の高精度化

3.1 はじめに ～不連続なピッチパタンが及ぼす影響～

一般にピッチは音声波形の周期的に変化する部分の最低周波数成分のことであり、音声の生成過程では声帯の振動数に相当する。つまり/b,d,g,z,m,n,j,w,…/などの**有声音**にはピッチが存在し、/p,t,k,h,…/のような声帯の振動を伴なわない**無声音**にはピッチが存在しないと考えられる。これは音声の**有声・無声判別**等における重要な性質である。しかし、2.5節で述べたようにパタン連続整合法による句境界検出おいてはこの性質は不都合であり、無声音によるピッチの不在、声量的にあいまいな発話によるピッチの不確かさによって生じたピッチパタンの不連続性は句境界検出結果に大きな影響を及ぼす。そのためシステムの性能が検出アルゴリズムに依存するのか、あるいはピッチパタンの精度に依存するのかを議論する必要が生じてくる。

韻律情報に関する多くの研究ではこの問題に対して、不連続区間を直線近似するなどの補間処理を施しているが、本システムにおいて、同種の処理を施したピッチパタンが有効であるかどうか疑問である。パタン連続整合セグメンテーションは他のシステムに比べて、個々の分析点 (時間フレーム) のピッチの値およびその値の変化に大きく依存し、距離歪みとして累積されていくので、ピッチパタンの特徴を損なうような近似・補間処理はあまり好ましくない。そこで、本章ではピッチパタンが全発声区間で連続になるように可能な限り波形分析からピッチの値を抽出する手法について提案し、そのパタンの精度について検討することを目的とする。

3.2　狭周波数帯域窓によるピッチの抽出

3.2.1　lag-window 法と cepstrum 法

ピッチの自動抽出に関する研究はこれまでに数多く報告されているが、いまだに完全な自動抽出法は実現しておらず、いずれの手法を用いても抽出誤りを避けられない。また、どの手法にも一長一短があり、全ての面で他に優れている手法も見いだされていない。比較的簡単に良好なピッチを抽出する手法に自己相関を用いた抽出法があり、その代表的なものとして **cepstrum 法** [17] と **lag-window 法** [18] が挙げられる。

3.2.1.1　cepstrum 法

音声波形をフーリエ変換し、実部と虚部の自乗和とって**パワースペクトル**を求め、対数をとった後に逆フーリエ変換して**ケプストラム** [17] を求める。次に、基本周波数の存在範囲に対応する探索区間でケプストラムの最大値を求め、最大値を与える**ケフレンシー** (quefrency) から基本周波数を決定する。

3.2.1.2　lag-window 法

lag-window 法の分析過程を図 3.1に示す。cepstrum 法と同様に音声波形 $x(t)$ よりパワースペクトル $S(f)$ を求め、逆フーリエ変換して自己相関関数 $C(\tau)$ を求める。次に、自己相関関数の遅延時間軸方向に減衰する**ラグ窓**

$$\omega_\tau = \frac{(L!)^2}{(L+\tau)!(L-\tau)!} \qquad (L \text{ は窓長}) \tag{3.1}$$

をかけて、再びフーリエ変換することによってパワースペクトルを平滑化 ($S'(f)$) し、パワースペクトル $S(f)$ との比をとってピッチ構造を分離する。その逆フーリエ変換 $\bar{C}(\tau)$ を求めると、基本周波数の探索範囲に著しいピークが現われるので、その位置からピッチが求められる。

この手法は cepstrum 法に比べて、ピッチ構造が明確に分離でき、またラグ窓長を選択することで低いピッチ、あるいは高いピッチに重点を置いた抽出が可能である。

3.2.1.3　比較実験

ここで、cepstrum 法と lag-window 法の比較実験を行う。本章におけるピッチ抽出実験では ATR の連続音声データベース中の A グループ 50 文章を用いることにする。また、ピッチの抽出は表 3.1 と同様の条件で行う。正解ピッチとして ATR の提供する視察修正による

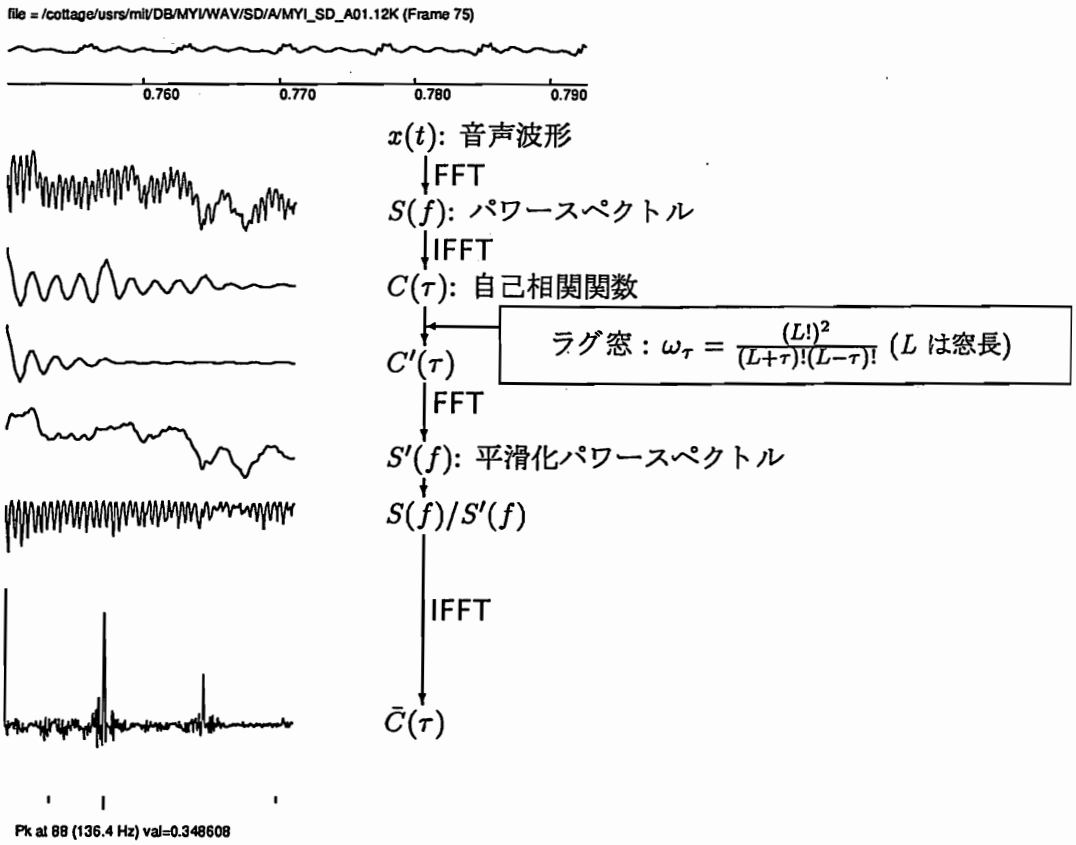


図 3.1: lag-window 法によるピッチ抽出過程

表 3.1: ピッチ分析条件

	ATR データベース	実験
FFT ポイント数	512 ポイント (42.7ms)	512 ポイント (42.7ms)
分析シフト	120 ポイント (10ms)	120 ポイント (10ms)
ピッチ探索範囲	50～350 ポイント (34～240Hz)	40～240 ポイント (50～300Hz)
ピッチ抽出法	ケプストラム法 (視察修正)	lag-window 法 (自動抽出) (ラグ窓長 $L = 100$)

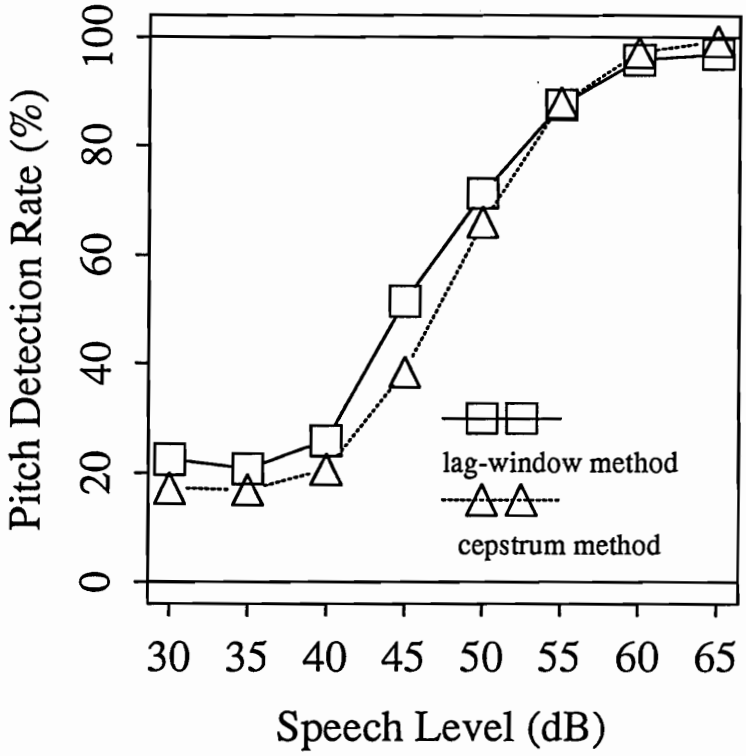


図 3.2: lag-window 法と cepstrum 法のピッチ検出率

ピッチの値を使用し、無声音区間は除外した。ここで、抽出したピッチが正解ピッチの前後 10 %以内に入った場合 (例えば、正解が 100Hz ならば 90Hz〜110Hz が正解区間) を正解検出されたとみなし、正解検出された割合を検出率として定義する。

比較実験の結果は図 3.2のようになった。横軸は 1 フレームの分析に切り出された 512 ポイントの入力音声の平均振幅を対数にしたものである。(ここでは音声レベルと呼ぶことにする。) いずれの手法も高レベルでは高い検出率を挙げているが、低レベル (発声が微弱な部分) では非常に検出率が低い。また抽出法を比較してみると、50dB 以下の領域では lag-window 法、50dB 以上の領域では cepstrum 法の方が優れていることが分かる。これは、レベルの低くなる発声領域 (文末など) ではピッチが低くなることが多く、自動検出において倍ピッチ (基本周波数値の倍の値) 誤りが多発するが、lag-window 法ではラグ窓をかけることにより、高域周波数成分を抑制できるので、このような結果になる。(ラグ窓長の

選択にもよる。)

高精度なピッチパタンとして望ましいことは全区間のピッチの値が正しいことである。しかし、自動抽出におけるピッチ誤りは避けられず、基本的には誤り訂正によって近づけて行くしかない。したがって、基本となる抽出法は、自動修正に適したものを選ぶ必要がある。自動修正の基本的手法は、前後数フレームのピッチの値より推定することであるので、発話区間内でピッチの良好な部分とそうでない部分とが著しく偏った抽出法は不都合である。cepstrum 法と lag-window 法の場合、高レベルの時間領域に関してはどちらも検出率が高く、誤りの自動修正は容易であるが、低レベルに関しては、検出率が低いので非常に難しい。2 つの手法を比較した場合、lag-window 法の方が低ピッチの検出に向いているので、以下の実験では lag-window 法を用いることにする。

3.2.2　パワースペクトルの低周波数領域におけるピッチ抽出

今、入力波形として基本周波数 f_0の正弦波の場合を考える。このときの入力信号は

$$x(t) = a \sin 2\pi f_0 t \tag{3.2}$$

であり、パワースペクトルは

$$P(f) = \frac{a^2}{4}\delta(f - f_0) \tag{3.3}$$

で表されるような f_0のインパルスとなる。逆 FFT を行って、$x(t)$ の自己相関関数を求めると、

$$C(\tau) = \frac{a^2}{2}\cos 2\pi f_0 \tau \tag{3.4}$$

となり、ピークが遅延時間 $1/f_0$の定数倍で現れる余弦波が得られる。$x(t)$ のスペクトル成分は f_0近傍に集中しているので、$P(f)$ 上で f_0の近傍を切り出すような周波数帯域窓をかけても得られる結果は同じである。

実際の音声波形は正弦波のような単純な波形ではないが、片側 6kHz あるパワースペクトルの全周波数成分を必要としなくとも、ピッチの探索範囲を十分に含む狭周波数成分だけで十分にピッチの抽出は可能である。

実験

ここで、パワースペクトルの低周波数成分のみによるピッチ抽出実験を行う。周波数窓は 512 ポイント (6kHz) の $1/2^n$の窓長の方形窓を用いることにする。つまり、窓長 64 のときは−750Hz〜750Hz の方形窓と定義する。実験は表 3.1の条件で行なう。実験評価に関し

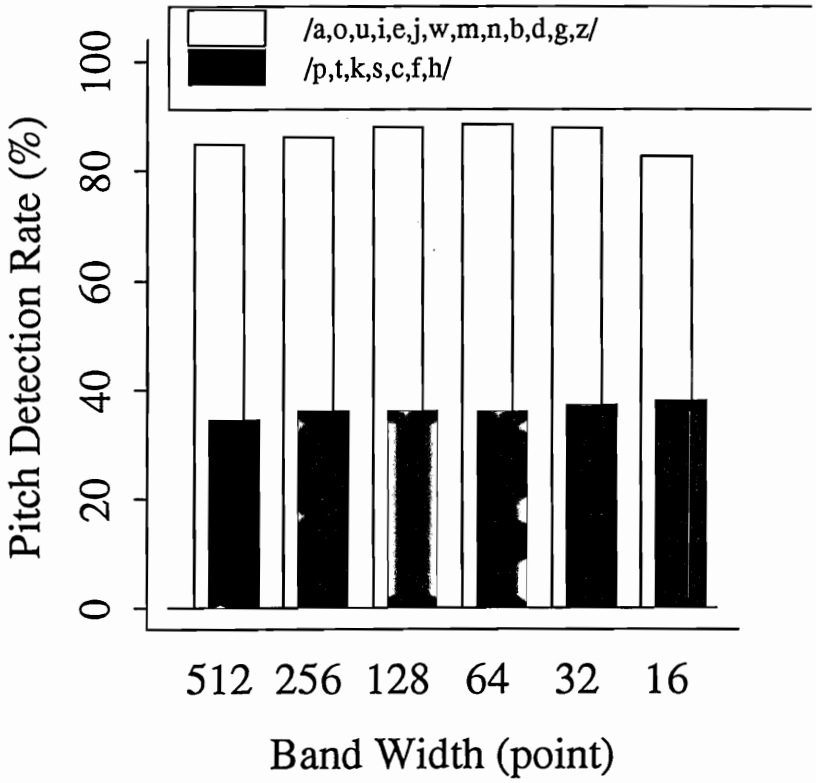


図 3.3: 低周波数帯域窓によるピッチ検出率

ては3.2.1節とは異なり、有声音と無声音について分けて評価する。ここで問題になることは、ATR のデータベースでは cepstrum 法で自動抽出した後、視察によって修正、および無声音の区間のピッチを削除しているため、無声音区間に対する基本周波数値は用意されていない。そこで ATR で視察修正されたピッチパタンをもとに若干の修正を加え、「ピッチの時間的な変動は比較的緩やかである。」という性質を前提にして、3 次元スプラインで補間近似したものを正解ピッチパタンとして予め用意した。正解ピッチ範囲については前実験同様±10%である。

図 3.3は低周波数領域に 512(6kHz)、256(3kHz)、128(1.5kHz)、 64(750Hz)、32(375Hz)、16(187.5Hz) ポイントの 6 種類の方形窓をかけてピッチ抽出実験を行った結果である。白棒が有声音、黒棒が無声音を表している。結果は窓幅が 64 ポイントのとき、わずかの差でははあるが最も検出率が高くなった。32 ポイント、16 ポイントまで小さくするとピッチの探索範囲の低い位置にしか自己相関のピークが現れず、64 ポイントを境にピッチ検出率は

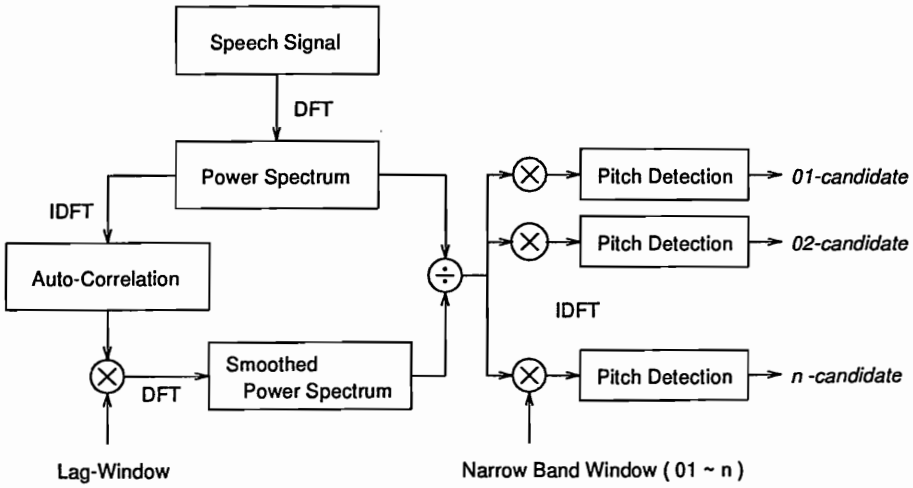


図 3.4: 複数周波数帯域のピッチ抽出過程

低下する。

3.2.3　パワースペクトルの高周波数領域におけるピッチ抽出

3.2.2節の実験よりパワースペクトルの低周波数成分のみを用いて、良好なピッチが得られることが分かった。しかし、無声音区間のピッチ検出率は上がったとはいえ、いまだ十分とは言えず、40%にも満たない。なぜなら、本来無声音は声帯の振動を伴なわないので、音声波形に周期性が見られず、ピッチが存在しないからである。しかし、**ささやき声**のように声帯を振動させずとも、呼気音を声帯波に代用して十分会話することが可能であり、ささやき声にもピッチアクセントが成立することが報告されている [19]。無声音のピッチの知覚の原因はまだ推測の段階ではあるが、**フォルマント** (formant) の移動、音声波形のピーク間距離の変化から、**バンドノイズ**、**周期性ピッチ**などが考えられている。そのような理由から無声音のピッチ推定は決して無意味なこととは言えず、ピッチパタンの連続性を考慮する上で必要不可欠である。

そこで狭周波数帯域窓による無声音のピッチ抽出について考えることにする。有声音のパワースペクトルでは低周波数成分 (およそ 800Hz 以下) が約 8 割を占めているのに対し、無声音は白色ランダム雑音で近似される。したがって、無声音の場合は低周波数領域のみによるピッチ抽出は不十分であり、パワースペクトルの高周波数領域についても検討する必要があると考えられる。

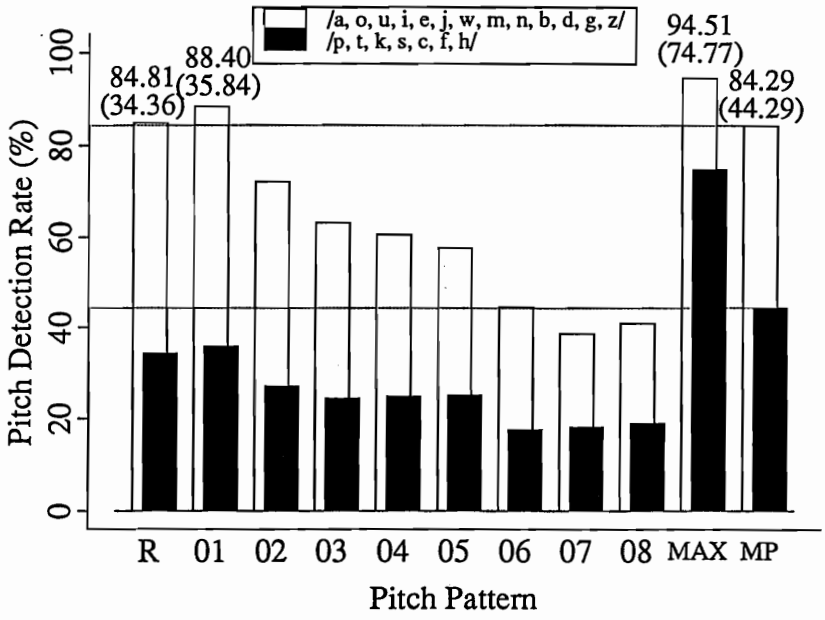


図 3.5: 帯域幅 64 ポイントのときの各帯域のピッチ検出率

実験

この実験では周波数帯域幅として 3.2.2節の実験で最も良好だった 64 ポイントを用いることにする。周波数軸上をオーバーラップさせることなく窓をシフトして全周波数成分を分割できるように窓の数を 8 に設定する。ここで合計 8 つの窓を低周波数帯域から順に 01〜08 窓と呼ぶことにする。(例えば 02 窓は−1.5kHz〜−0.75kHz、0.75kHz〜1.5kHz の計 64 ポイント。)

結果は図 3.5の 01 から 08 で表わされる棒グラフのようになった。また R で表されているグラフは従来の全周波数成分を使用した場合のピッチ検出率である。01 から 08 まで有声音、無声音ともに高周波数帯域になるほど検出率は下がっている。しかし、高域になるほど低域での正解ピッチが一方的に誤りに転じているというわけではなく、高域で分析することによって、これまで得られなかったピッチを抽出することも可能になった。図 3.5中の MAX のグラフは 01 領域で不正解であったピッチを 02 領域のピッチで置換し、さらに不正解であったものを 03 領域のピッチで置換して、最終的に全領域のピッチを用いて検出率が最も高くなるように計算した結果である。これを見ると、01 から 08 領域まで最適なピッチを選択することができれば、有声音で 94.51%、無声音でも 74.77%まで検出率を上げることができることが分かる。

3.3　複数周波数帯域のピッチ候補による連続なピッチパタンの作成

3.3.1　複数ピッチ候補の抽出

3.2.3節の結果からも分かるように、複数のピッチ候補を用いて誤り訂正を行えば、かなり検出率の高いピッチパタンが得られるはずである。複数のピッチ候補として、前節の 01 領域から 08 領域までのピッチを採用することが可能であるが、古くからの手法として文献 [20] の中で用いられている手法もある。その手法は短区間自己相関関数が局所的にピークを迎える点のうち、その値が最も大きなものから順にいくつかのピーク点を選んで候補とし、そのときの自己相関の値を確信度として与えるものである。しかしこのような確信度付きピッチを候補とした場合、ピッチ推定にさまざまなパラメータの結合などが考えられ、試行錯誤に落ち入り易く、一般化が困難である。そこで、本節では前節の狭周波数帯域窓による分析で得られるピッチを候補にすることの意義と利点についてまとめる。

この手法の基本的な考えは以下の通りである。$\bar{C}(\tau)$ は $S(f)/S'(f)$ を逆 FFT したものであるから、パワースペクトルの全周波数帯域を逆 FFT して得られる自己相関関数$\bar{C}(\tau)$は帯域幅 64 ポイントの窓をかけて得られる 01 領域から 08 領域の自己相関関数$\bar{C}_{01}(\tau)$〜$\bar{C}_{08}(\tau)$ の和である。つまり、適当な周波数帯域窓をかけることによって自己相関関数を分解することができると考えられる。ここで、従来の手法で抽出したピッチの値がパワースペクトルのいずれかの周波数成分に依存するものであるならば、他の周波数領域からは第 2、第 3 の異なるピッチ候補が得られるであろうし、どの領域からも同じ値が抽出されるようであれば、確からしいピッチとしてその候補を選択する確率が増すことになる。つまり、確信度を兼ね備えたピッチ候補群として扱うことができる。

3.3.2　DP によるピッチの選択

複数候補からのピッチの選択には連続性を最も保証した DP(Dynamic Programming) 法が有効であると考えられる [21][22]。これまでに述べたように、このピッチ候補群の特徴として、

1. パワースペクトルの全周波数成分を用いるよりも低周波数成分のみの方がピッチの抽出精度が高いこと、
2. 確からしい (いわゆる確信度の高い) ピッチは、いずれの周波数領域からもほぼ同じ値が得られ、実質的な候補数が絞られていること、

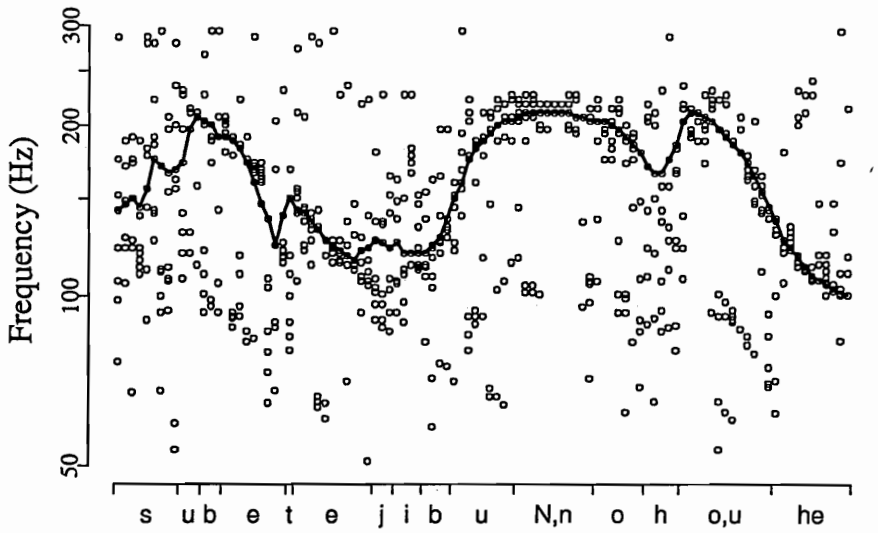


図 3.6: DP によるピッチ選択

が挙げられる。このうち、DP 法が有効な理由として、特に 2 の性質が挙げられる。DP によるピッチ選択の基準はピッチの変化量 (隣接する時間フレーム間の対数ピッチの差の自乗) の累積を最小にすることである。候補数が多過ぎるとピッチパタンの短絡が生じる危険があるが、確信度の高いピッチの時間フレームでは異なる値の候補が少数であるので、ほぼ確実に正確な値を選択することができ、全体的にピッチパタンの概形を損なうことがない。図 3.6は窓幅 64 による候補数 8 の場合を時間—周波数軸上にプロットしたものである。音素/N/の近辺ではほぼ安定したピッチが得られ、ピッチパタンの形を損なっていないことが分かる。また音素/j/付近でわずかながら短絡した個所が見られるが、この問題は検出精度が高い低域ピッチ候補の重み付け等によって解消することにする。

この手法で作成されたピッチの検出率は図 3.5中の MP で示されている。この結果、有声音で 84.29%、無声音で 44.29%の検出率を挙げることができた。MAX の値には残念ながら及ばなっかたが、無声音はこれまでの単一の領域のピッチパタンに比べて最良のものであることが分かる。また、有声音に関しては 01 のものより低い検出率となったが、01 のピッチに重みをかけて、02 以降のピッチで置換するという実験 [23] では 89.12%の検出率を挙げ、01 のパタンよりも正確な値のピッチパタンを作成できることを証明した。

3.4　句境界検出による連続なピッチパタンの評価

3.4.1　まえがき

前節まででピッチ検出率によるピッチパタンの評価を行なったが、本研究においては個々のピッチの値の確からしさよりも、認識システムにおいて果たす役割の方がより重要である。そこでこの節ではピッチパタンの不連続部分を除去したことによる句境界検出結果への影響について検討する。連続なピッチパタンとして、複数候補から選択作成されたパタンと、不連続区間を補間したパタンを作成し、これまでの未処理ピッチパタンと比較検討する。

3.4.2　不連続区間を補間したピッチパタンの評価

3.4.2.1　スプライン補間パタン

図 3.7の上から 3 段目のパタンは補間区間の始点と終点を 3 次元スプラインで近似したものである。滑らかに補間することができ、ピッチの谷間の構造など特徴をよく捉えているようにも見えるが、補間の対象となる区間の選定 (不連続性の判断) や補間の始点、終点の外側のピッチの微妙な変化によってパタンが大きく異ることがある。(S 言語の spline 関数による処理。[24])

3.4.2.2　平滑化パタン

図 3.7の最下段のパタンは全区間を局所重みつき最小二乗法 (locally weighted regression) によって平滑化したパタンである。重みを大きくすれば図のようにパタンの特徴をかなり損なう。また重みを小さくすれば不連続性を除去することができず、その見極めが困難である。(S 言語の lowess 関数による処理。[25])

3.4.2.3　線型補間パタン

図 3.7の上から 2 段目は不連続区間を線型補間したパタンである。スプライン補間に比べて危険が少なく、安定したパタンが得られるので、比較的直線近似ピッチパタンとしてよく用いられる。

ここで不連続区間を補間したパタンの代表として、線型補間パタンによる句境界検出実験を行なった。実験条件は 2.4節の実験と全く同じである。章末の図 3.9～図 3.10はその結果である。図中の “○” は未処理パタン “△” は線型補間パタンである。挿入誤りに関して

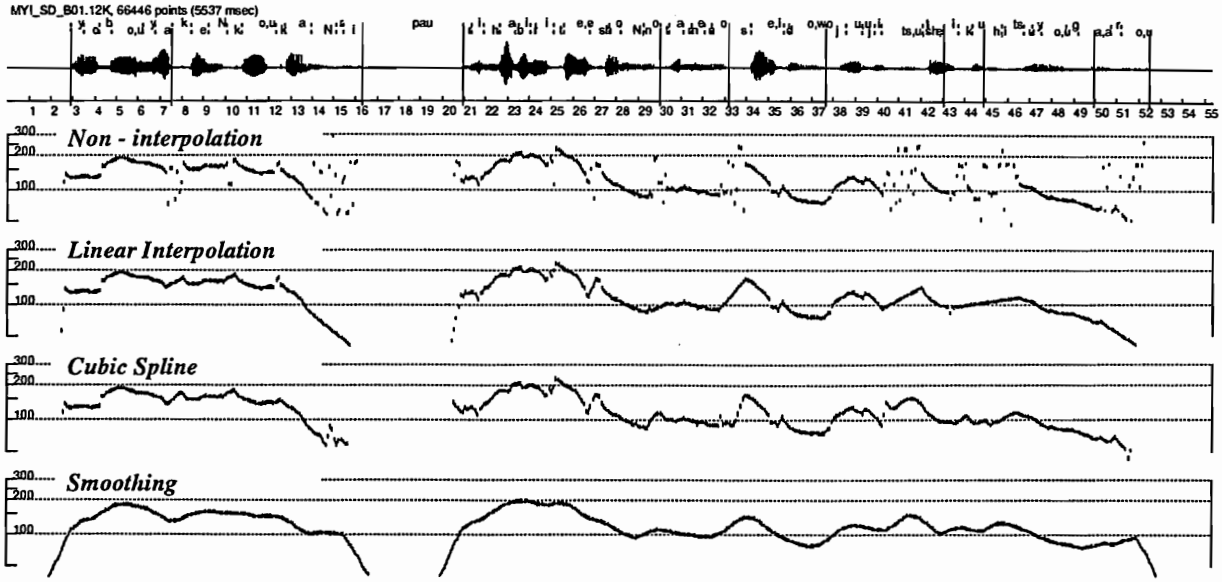


図 3.7: 未処理・線型補間・プライン補間・平滑化によるピッチパタン

は未処理パタンよりも低いので一見性能の良さを思わせるが、同時に句境界検出率もかなり低くなっていることに気付く。これは、線型補間処理によって句境界検出総数そのものが減少したことを意味している。実際、語と語の境界の発声があいまいになり易いので不連続部分も句境界周辺に集中し、その部分を補間によって直線近似したため、ピッチパタンの句境界構造が失なわれてしまったと考えられる。

3.4.3　複数ピッチ候補から作成された連続なパタンの評価

線型補間パタンでは句境界構造が損なわれてしまって、結果的に句境界検出率を下げてしまった。そこで補間による近似を避け、分析過程から得られる複数のピッチ候補から選択作成されたピッチについて評価する。

3.4.3.1　履歴参照マトリクスを用いて候補を選択したパタン

ピッチの変化は緩やかであるという性質から、時間的な連続性を導入してピッチの推定を試みた研究は多く、文献 [20] もそのひとつである。簡単にまとめると、既存のピッチ抽出法を用いて得られる複数のピッチ候補に対して、過去数フレームの履歴を重み付きマトリクスを通じて参照し、ピッチの確信度 (自己相関値)、ピッチの値からスコアを計算して最も妥当と思われる候補を現在の第 1 候補として選択する方法である。この手法の特徴と

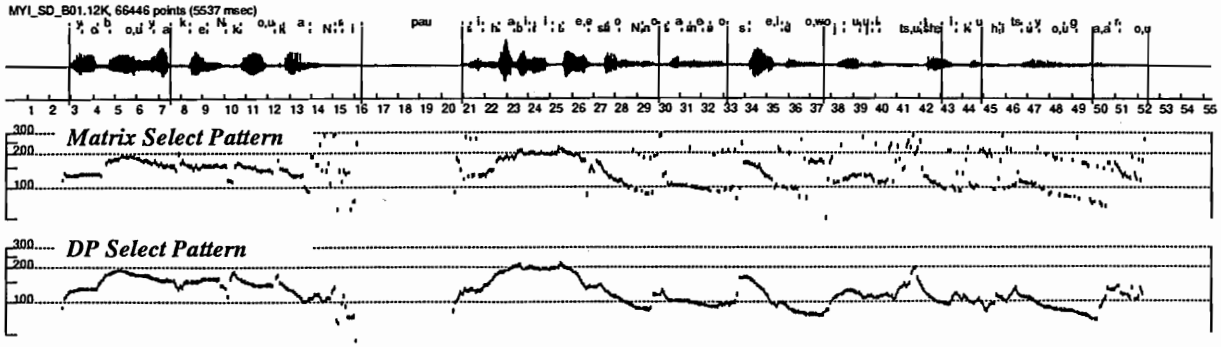


図 3.8: 複数候補からのピッチ選択によるピッチパタン

して、過去の各フレームのピッチの値が決定されている必要が無く、最終的なピッチの推定はフレーム毎に独立に行われるため、必ずしも時系列順に処理を進める必要が無いことである。

この手法を用いて作成されたピッチパタンが図 3.8の上段のパタンであるが、明らかに全発声区間に対しての連続性が保証されていないことが分かる。実際、この手法が有効に働くのは明瞭に発声された短区間のみであり、単一候補による検出率がある程度高いことが必要である。

3.4.3.2　DP を用いて候補を選択したパタン

図 3.8の下段のパタンは 3.3節で述べた DP を用いて候補を選択したパタンである。履歴マトリクスを用いて作成したパタンに比べて連続性が保証され、線型補間パタンに比べてピッチの谷間などの句境界構造が良く表われていることが分かる。

このパタンを用いた句境界検出結果は章末の図 3.11～図 3.12 に示す通りである。図中の“○” は単一候補によるパタン (従来のパタン)、“◇” は周波数帯域窓 64 の 8 候補ピッチによる DP 選択パタンである。デルタピッチの寄与率αが 0.0 でテンプレート数が少ないときの数例を除けば、ほぼ全ての場合において DP 選択パタンの検出率の方が高く、最大検出率もこれまでの 87.058%に対し、88.164%まで上げることができた。これは不連続による影響が排除されたことを表わし、これによりデルタピッチの寄与による性能、テンプレートの数による性能が純粋に評価できるようになった。おおよそ、テンプレート数が増えるにつれて検出率が高くなり、ある程度の検出率 (おおよそ 88%) で頭打ちになる。デルタピッチの寄与率についても 2.4.2節で述べた通りの結果となった。また句境界挿入誤り率も単一候補パタンよりも低い。つまり、単一候補によるパタンにおいても複数候補から DP 作成

されたパタンにおいてもほぼ同数の句境界を検出することができるが、その正解と誤りの比率において、後者のパタン方が優れていることを示している。

3.5　まとめ

本章では不連続なピッチパタンによる句境界検出への悪影響を指摘し、連続性を重視したピッチパタンの高精度化を図った。

3.2節では弱パワー音声のピッチ抽出に有効な lag-window 法をベースにして、狭周波数帯域窓を用いた複数ピッチ候補の抽出法について提案した。3.2.3節の実験より、本手法によるピッチ候補群は第 1 候補 (低域の候補) が非常に良好で、第 2 候補以下のピッチも誤り訂正に有用であることが示された。また、複数の周波数帯域で同一の値の候補を抽出する場合も許可しているので、確からしいピッチ候補は自ずと選択される確率が増し、確信度のいらないピッチ候補群としての特徴も持っている。

DP 選択によるピッチパタン作成ではこの性質が有効で、3.3.2節では概形を損なわず、ピッチの谷間などの句境界構造を備えた良好なピッチパタンを作成した。

3.4節では、このピッチパタンを用いて句境界検出実験を行い、従来の不連続なピッチパタンによる検出率を上回る、88.164%の検出率を達成した。

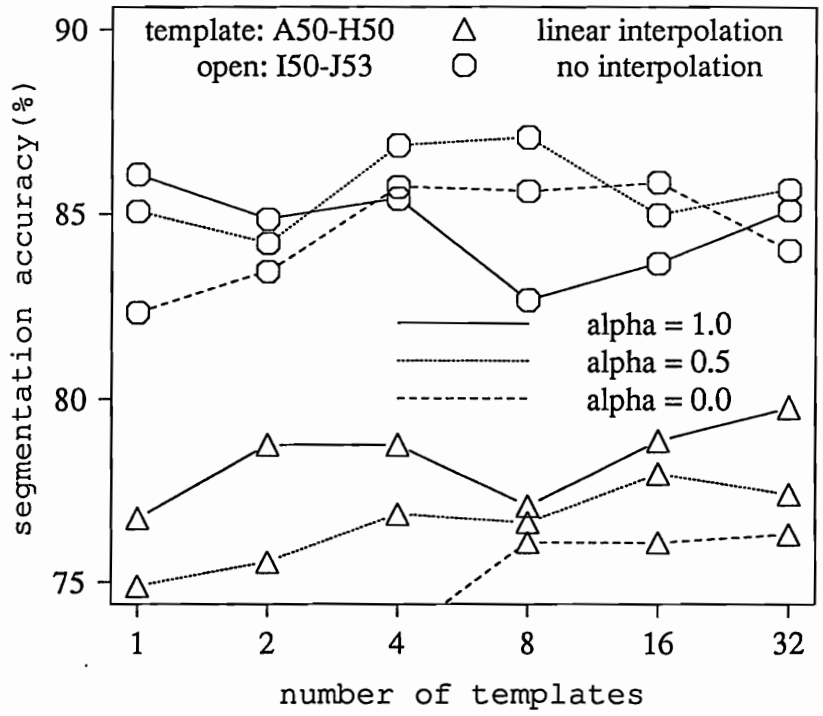


図 3.9: 線型補間パタンの句境界検出率

表 3.2: 線型補間パタンの句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	72.235	71.681	73.451	76.106	76.106	76.327
0.5	74.889	75.553	76.881	76.659	77.987	77.434
1.0	76.770	78.761	78.761	77.102	78.872	**79.757**

(単位：%)

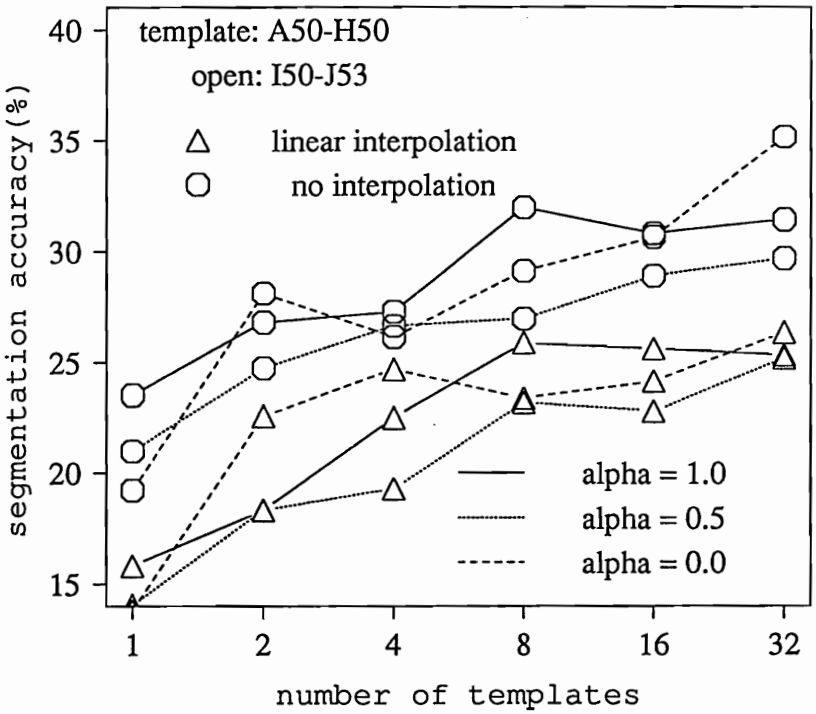


図 3.10: 線型補間パタンの句境界挿入誤り率

表 3.3: 線型補間パタンの句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	13.803	22.548	24.667	23.349	24.126	26.298
0.5	14.054	18.321	19.287	23.149	22.775	25.142
1.0	15.830	18.303	22.477	25.868	25.604	25.274

(単位：%)

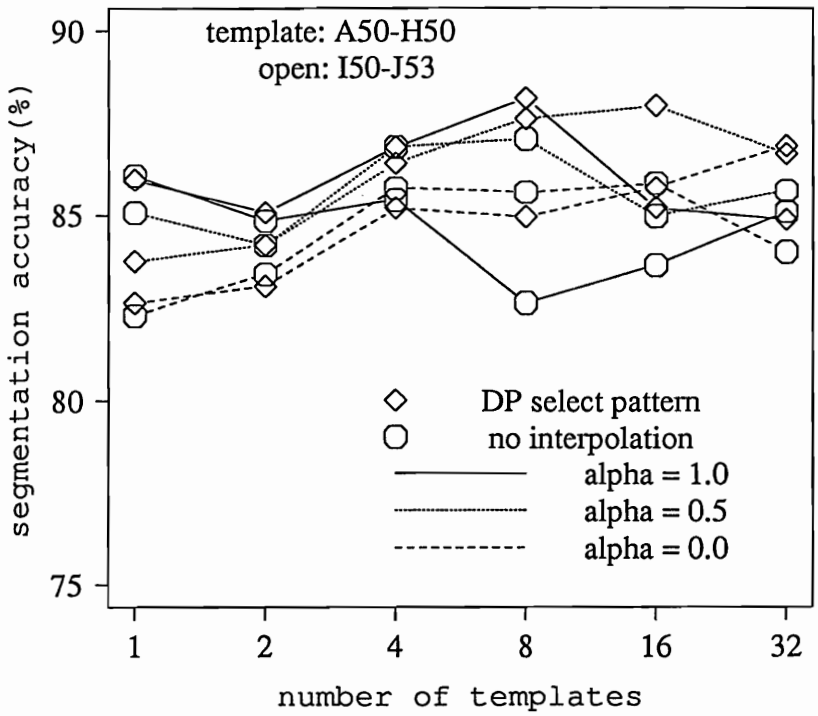


図 3.11: DP 連続パタンの句境界検出率

表 3.4: DP 連続パタンの句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	82.633	83.075	85.177	84.956	85.730	86.836
0.5	83.739	84.181	86.394	87.611	87.942	86.615
1.0	85.951	85.066	86.836	**88.164**	85.177	84.845

(単位：%)

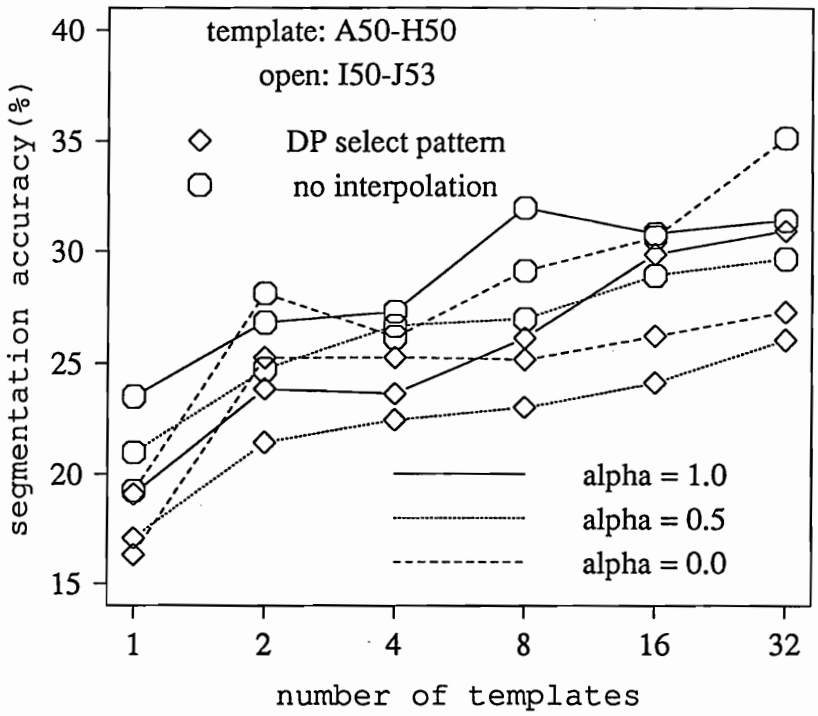


図 3.12: DP 連続パタンの句境界挿入誤り率

表 3.5: DP 連続パタンの句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	16.312	25.237	25.230	25.128	26.174	27.229
0.5	17.068	21.429	22.441	22.999	24.104	25.988
1.0	19.063	23.834	23.613	26.087	29.846	30.878

(単位：%)

第 4 章

句境界検出の高精度化

(固定長テンプレート)

4.1 はじめに ～ピッチの高さに関する問題～

2.5節において、従来のパタン整合法の 2 番目の問題点としてテンプレートパタン固有の絶対的なピッチの高さを挙げた。いずれのテンプレートパタンも各クラスタに属するアクセントパタンの平均的な高さとして定義されているため、ピッチパタンの著しく高い個所、あるいは著しく低い個所では整合が追随しきれずに、その結果テンプレートの両端の高さを基準とした句境界が検出される恐れがある。従来の方法ではデルタピッチパタンを 2 次的なパラメータとして導入して分析フレームの近傍の形状の変化を表わした。これにより特徴的な句境界構造にあたるピッチパタンの谷間が、いかにテンプレートの高さからかけ離れていようとも、単純に負から正に変わるパタンとして認識することができるようになった。

ここでパタンの類似性について考察する。図 4.1はいずれも同じパタンをもとに作図したものであり、長さか高さ、あるいはその両方が異なるだけであるから、ここでは同一のパタンとして認識したい。仮に (a) を基準のテンプレートパタンとして考えることにする。(b) は長さのみが異なるパタンであり、発声上では発話速度などの影響として考えられるパタンである。この場合、パタン長が DP による時間軸伸縮の許される範囲の長さである限り、全く同じパタンとして認められる。(c) は高さのみが異なり、これは発話文中の位置、つまり 2.2節で述べたようなフレーズ成分の影響や、話者による特定の高さによる違いとして考えられる。この場合、時間軸上でどのように伸縮しても一致することはなく、二乗誤差はかなり大きくなる。しかし、傾きを表わすデルタパタンで整合をすることにより同じパタンとして認識できる。しかし、(b)(c) の両方のケースが同時に生じた場合として (d) を考えた場合、パタン間の距離は大きく、またその傾きの違いからデルタパタンをもってし

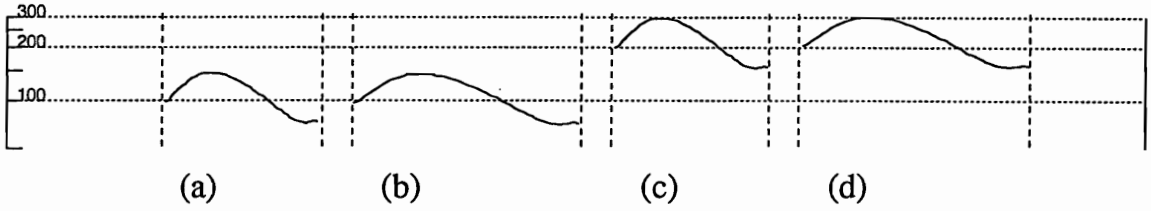


図 4.1: 類似なパタン

ても誤差は避けられない。(a) と (d) を一致させる最も簡単な方法はテンプレート (a) に高さ方向への自由度を与え、高さを揃えた後、時間軸方向へ DP 伸縮させることである。

本章では、この原理を用いた句境界検出法について検討する。また自由度が増えたことにより、テンプレートのアクセントパタンに対する表現能力が上がり、句境界検出数が増加することが予想される。結果として正解検出数が増加することになるが、それに伴なって挿入誤りも増加することになる。そこで、アクセントパタンの遷移情報によってテンプレート系列の接続に関する最適制御についても検討する。

4.2　テンプレートの高さを可変にした句境界検出

4.2.1　まえがき

これまで、個々のフレームの絶対的な値のみを表すピッチパタンと近傍の形状を表わすデルタピッチパタンによる 2 次元のマトリクスパタンによる整合を行なってきた。しかし、前節で述べたように、このマトリクスパタンでは整合できない類似パタンが存在する。そこでパタンの始点からの相対的な変化を基準に学習アクセントパタンを分類することで、直接テンプレートに形状の変化を導入することにする。これに伴ない、テンプレートに高さ方向への自由度を与え、相対的な値によるパタン整合を試みる。

4.2.2　アクセントパタンの相対的変化 (高さ可変テンプレート)

2.3.1節でのテンプレートの学習はアクセントパタンの絶対的な値によるクラスタリングであり、いわばピッチの高さに関する情報を含んだ形によるクラスタリングであった。しかし、この学習サンプルはピッチパタンを視察によって切り出しただけのものであるから、2.2節で述べたようにフレーズ成分とアクセント成分の和の形で表現されている。したがって、同一のアクセント成分でもフレーズ成分の勾配と高さによって、異なるものとして分

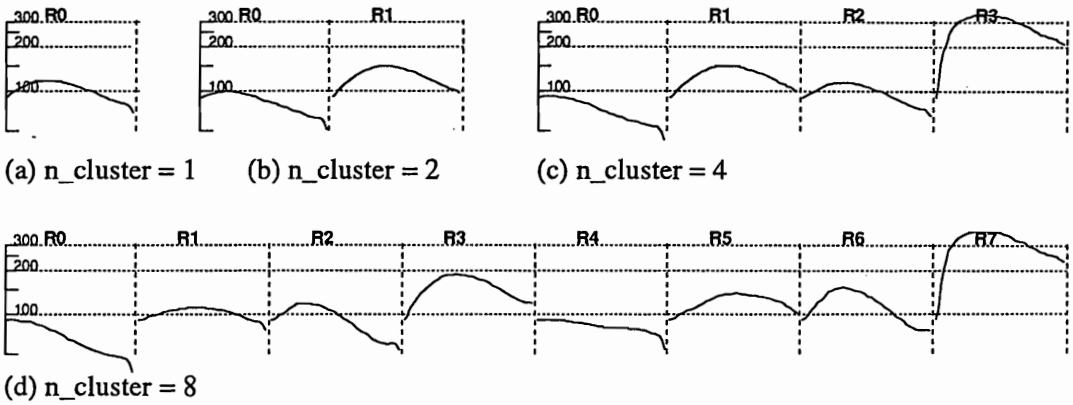


図 4.2: 高さ可変テンプレート (相対的なアクセントパタン)

類されてしまう。視察でアクセント境界を与えたピッチパタンよりアクセント成分のみを抽出して学習することは可能であるが、自動抽出においてアクセント成分のみのパタンを作成するのは困難であるから、今のところアクセント成分を基準としたテンプレートは効果的ではない。しかし、フレーズ成分が文頭から文末に向かう緩やかな下降で近似されることより、フレーズの立ち上がり (文頭、ポーズ後) を除く位置のフレーズ成分の勾配をほぼ一定と仮定すると、アクセントパタンの相対的な変化はフレーズ成分の立ち上がりを加えたアクセント成分と一定の下り勾配を加えたアクセント成分の 2 種類として分類できる。

図 4.2はアクセントパタンの相対的な変化によって分類した結果である。視察で切り出されたアクセントパタンを始点の高さが対数表示で同じになるように平行移動したこと以外は 2.3.1節の学習条件と全く同じである。ここでは図示し易いように 90Hz を仮の始点の高さとしたが、相対的なパタンではピッチの高さは実質的な意味を持たない。図に見られるように、テンプレートの形は基本的には 2.3.1節のものと同様にアクセント語尾が下降する“へ” の字型になるが、テンプレート数が増すにつれてアクセント語頭から下降する一方のパタンも生成されるようになる。これは、フレーズ成分の下り勾配とアクセント成分の立ち上がりの勾配がちょうど相反するときのパタンとしてや、アクセント結合によって立ち上がりが平坦になったパタンとして分類される。また、クラスタ数が 4、8 ともなれば、立ち上がりの急なパタンが得られるが、これは文頭、あるいはポーズ後の発声開始時のもので、フレーズ成分の立ち上がりを含んだパタンである。(実際には無音区間の対数ピッチの下限の値を 1.70(50Hz) としてスケールの制限をしている。)

4.2.3　高さ可変テンプレートによる句境界検出

図 4.2のテンプレートは学習側から見れば「**相対的なアクセントパタン**」というべきパタンであるが、句境界検出法から見れば、高さに関する自由度を与えた「**高さ可変テンプレート**」ともいうべきパタンである。以後、観点の違いからそのように呼び分けることもあるが、実質的には同じパタンを指していると思ってもさしつかえない。したがって、このテンプレートを用いた手法を「**高さ可変テンプレートによる句境界検出法**」と呼び、テンプレートにオフセットを与えた整合法という意味で「**OS-Matching 法**」と略称することにする。さて、ここで高さに関する自由度というのはテンプレートの上下への並行移動を意味するのであるが、この移動幅を決定する方法は次の 2 通り考えられる。

1. 分析フレームのピッチの高さにテンプレートの始点の高さを合わせる。
2. ピッチテンプレート系列の接続境界を連続にする。

以降、1 の場合を「**高さ可変 (A)**」、2 の場合を「**高さ可変 (B)**」と呼ぶことにする。

4.2.3.1　ピッチの高さ揃え (高さ可変 A) による手法

高さ可変 (A) の場合、テンプレートのどのフレームをピッチパタンのどのフレームに合わせるかが問題である。最良の合わせ方はテンプレートの平均の高さと未知のアクセントパタンの平均の高さを揃えることであるが、これは句境界検出結果を用いて反復計算する必要があり、One-Stage DP 上では実現できない。仮に One-Stage DP を繰り返し実行したとしても収束に時間がかかり、とても実時間システムとして成り立たない。また、テンプレート長を L とし、ピッチパタンの L フレーム平均で高さを合わせることも試みたが、良好な結果は得られなかった。ただし、それぞれの条件に合わせたテンプレートの学習を行なっているので、前節で述べたような相対的なアクセントパタンを用いた句境界検出とは言えない。そのような理由からも、分析フレームのピッチの値とテンプレートの始端の高さを合わせるのが最も妥当で、One-Stage DP 法においても容易に実現できる。

[アルリズム]

未知入力パタンのフレーム：$i = 1, \cdots, N$
テンプート番号：$k = 1, \cdots, K$
テンプレート kのフレーム：$j = 1, \cdots, J_k$
(i, j, k) におけるオフセット (高さ方向の対数移動距離)：$O(i, j, k)$
(i, j, k) における累積距離：$D(i, j, k)$

(i,j,k) におけるフレーム間距離 : $d(i,j,k,O)$

対数ピッチ値を $F_i(i)$

テンプレート番号 kにおけるフレーム jの対数ピッチ値を $F_{tk}(j)$

とするとき、次のように定義する。

$$d(i,j,k,O) = (F_i(i) - (F_{tk}(j) + O))^2$$

Step1 initialize $D(1,j,k) = \sum_{n=1}^{j} d(1,n,k)$

Step2 (a) for $i := 2$ to N do steps (b) - (e)

(b) for $k := 1$ to K do steps (c) - (e)

(c)
$$\begin{aligned} k^* &= \arg \min_{k'=1,\cdots,K} [D(i-1, J_{k'}, k')] \\ O(i,1,k) &= F_i(i) - F_{tk}(1) \\ D(i,1,k) &= d(i,1,k,O(i,1,k)) + D(i-1, J_{k^*}, k^*) \\ &= 0 + D(i-1, J_{k^*}, k^*) \end{aligned}$$

(d) for $j := 2$ to J_k do step (e)

(e)
$$\begin{aligned} O_{11} &= O(i-1, j-1, k) \\ O_{12} &= O(i-1, j-2, k) \\ O_{21} &= O(i-2, j-1, k) \\ O_{22} &= O(i-2, j-2, k) \\ D_{11} &= d(i,j,k,O_{11}) + D(i-1, j-1, k) \\ D_{12} &= d(i,j,k,O_{12}) + D(i-1, j-2, k) \\ D_{21} &= d(i,j,k,O_{21}) + d(i,j,k,O_{21}) + D(i-2, j-1, k) \\ D_{22} &= d(i,j,k,O_{22}) + d(i,j,k,O_{22}) + D(i-2, j-2, k) \\ (x^*, y^*) &= \arg \min_{x=1,2,y=1,2} [D_{xy}] \\ D(i,j,k) &= D_{x^*y^*} \\ O(i,j,k) &= O_{x^*y^*} \end{aligned}$$

Step3 Trace back the best path

Step2 の (c) は 2 つのテンプレートの接続を決定する。この分析フレーム i でのピッチパタンとテンプレートの距離は、高さを始点で合わせるため常に $d(i,1,k,O(i,1,k)) = 0$ と

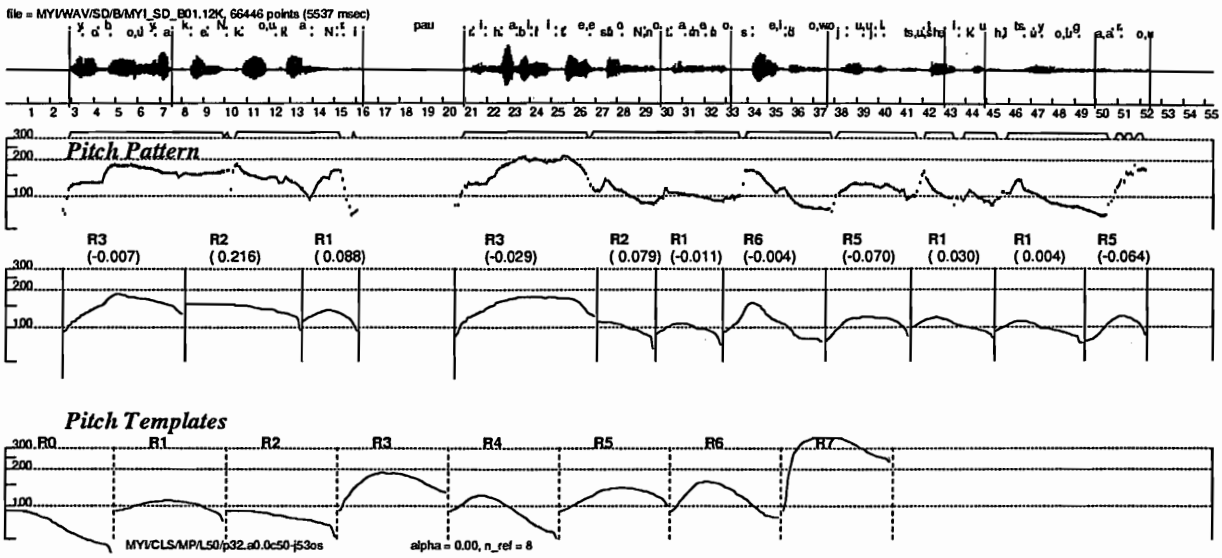


図 4.3: 高さ可変テンプレート (A) による句境界検出例

なるので、接続フレームは直前の分析フレームまでで最も累積距離 $D(i-1, J_{k^*}, k^*)$ が小さく終端したテンプレートに定まる。また各々のテンプレート kで始まったときのオフセット値は $O(i, 1, k)$ に格納され、処理 (e) に受け渡される。(e) はテンプレートの始点以外での処理であり、距離計算では常にオフセット値 O_{xy}が呼び出され、累積距離を最小にする $O_{x^*y^*}$がそのフレームのオフセット値として保存される。

図 4.3はテンプレートの始点の高さを分析フレームのピッチの高さで揃えた場合の句境界検出例である。中段のパタンはピッチパタンとピッチテンプレート系列の整合結果であるが、R で始まる数値は整合テンプレートの番号 (ここでは 0〜7) を、括弧内の数値は高さ方向への移動距離 (対数尺度) を示している。テンプレートの自由度が高まっているので従来の結果 (例えば図 2.9) に比べて、テンプレート系列による擬似ピッチパタンが、かなりピッチパタンに近似していることが分かる。

この手法の問題点として次の 2 つが挙げられる。第 1 にアクセントパタンの中の更に微少な部分パタンと整合するケースが多く、挿入誤りを生じ易いという点である。挿入誤り自体は深刻な問題ではないが、時として正解句境界の検出を不可能にする場合があるので、いずれは解消すべき問題である。第 2 点は発声開始直後や、ポーズのあとの発声再開直後に現れる最初の句境界が抽出しにくいことである。これは発声開始時のピッチの立ち上がりを表現するパタンが少ないためであり、テンプレートの始点の高さを無音区間の便宜上のピッチの高さ 1.70(50Hz) に合わせた場合には、テンプレート長 L の収縮限界にあたる

$L/2$ フレームが最初のパタンとして検出され易くなる。この問題の解消法として、そのような発声開始直後のパタンに対しては分析フレームとテンプレートの始点で合わせるのではなく、互いに数フレーム後 (後述の実験では 3 フレーム) に出現する高さで揃えることにする。

4.2.3.2　接続テンプレートの境界連続 (高さ可変 B) による手法

テンプレートの高さを決定するもうひとつの方法はピッチテンプレート系列の接続境界を連続にする方法である。これは高さ可変テンプレート、つまり相対的なアクセントパタンがピッチパタンのある点 (アクセントパタンの始点) からの変化を表わすのと同時に、先行するアクセントパタンの終端からの変化も表わしていることに起因する。またピッチパタンの連続性からも妥当な方法であると言える。

[アルリズム]

未知入力パタンのフレーム : $i = 1, \cdots, N$

テンプート番号 : $k = 1, \cdots, K$

テンプレート kのフレーム : $j = 1, \cdots, J_k$

(i, j, k) におけるオフセット (高さ方向の対数移動距離) : $O(i, j, k)$

(i, j, k) における累積距離 : $D(i, j, k)$

(i, j, k) におけるフレーム間距離 : $d(i, j, k, O)$

対数ピッチ値を $F_i(i)$

テンプレート番号 kにおけるフレーム jの対数ピッチ値を $F_{tk}(j)$

とするとき、次のように定義する。

$$d(i, j, k, O) = (F_i(i) - (F_{tk}(j) + O))^2$$

Step1 initialize $D(1, j, k) = \sum_{n=1}^{j} d(1, n, k)$

Step2 (a) for $i := 2$ to N do steps (b) - (e)

(b) for $k := 1$ to K do steps (c) - (e)

(c)
$$k^* = \arg \min_{k'=1,\cdots,K} [d(i, 1, k, O(i, 1, k')) + D(i-1, J_{k'}, k')]$$
$$(O(i, 1, k') = F_{tk'}(J_{k'}) + O(i-1, J_{k'}, k') - F_{tk}(1))$$
$$O(i, 1, k) = F_{tk^*}(J_{k^*}) + O(i-1, J_{k^*}, k^*) - F_{tk}(1)$$
$$D(i, 1, k) = d(i, 1, k, O(i, 1, k)) + D(i-1, J_{k^*}, k^*)$$

(d) for $j := 2$ to J_k do step (e)

(e)
$$
\begin{aligned}
O_{11} &= O(i-1, j-1, k) \\
O_{12} &= O(i-1, j-2, k) \\
O_{21} &= O(i-2, j-1, k) \\
O_{22} &= O(i-2, j-2, k) \\
D_{11} &= d(i, j, k, O_{11}) + D(i-1, j-1, k) \\
D_{12} &= d(i, j, k, O_{12}) + D(i-1, j-2, k) \\
D_{21} &= d(i, j, k, O_{21}) + d(i, j, k, O_{21}) + D(i-2, j-1, k) \\
D_{22} &= d(i, j, k, O_{22}) + d(i, j, k, O_{22}) + D(i-2, j-2, k) \\
(x^*, y^*) &= \arg \min_{x=1,2, y=1,2} [D_{xy}] \\
D(i, j, k) &= D_{x^* y^*} \\
O(i, j, k) &= O_{x^* y^*}
\end{aligned}
$$

Step3 Trace back the best path

前節の**高さ可変 (A)** と比較して、**Step2** の (c) が多少異なる。この手法では $i-1$ フレームで終端した各々のテンプレートの終端点の高さが、次のテンプレートの始点の高さとして受け継がれていくので、$i = N$ でない限り、$i-1$ フレームまでのテンプレート系列の決定にはテンプレートの接続による i フレームでの誤差を考慮しなければならない。

図 4.4はテンプレートの始点の高さを先行するテンプレートの終端に連続になるように接続した場合の句境界検出の結果である。この方法ではテンプレートとテンプレートとの接続が滑らかになり、その接続境界はピッチパタンの局所的な谷間を近似することができる。したがって、**高さ可変 (A)** による句境界検出と異なり、アクセントパタンの部分パタンと整合する危険も少なく、また、発声開始直後などで挿入誤りが頻繁に検出されるということもない。しかし、この手法の問題点として、検出率を上げるためにはより多くのテンプレートが必要なことである。仮にテンプレート数 1 のときを例に挙げれば、始端より終端の方が低くなる“へ”の字型のパタンのみであるから、複数個接続するとテンプレート系列の終端の高さが始端の高さに比べてずいぶん低くなってしまう。結果として、高さを維持するために少ない個数のテンプレート系列で整合し、検出される句境界の数も少なくなる。逆に、始端よりも終端の方が高いテンプレートのみという場合も、稀なケースではあるが同様に不都合である。つまり、少なくとも始端より終端が上がるパタンと逆に下が

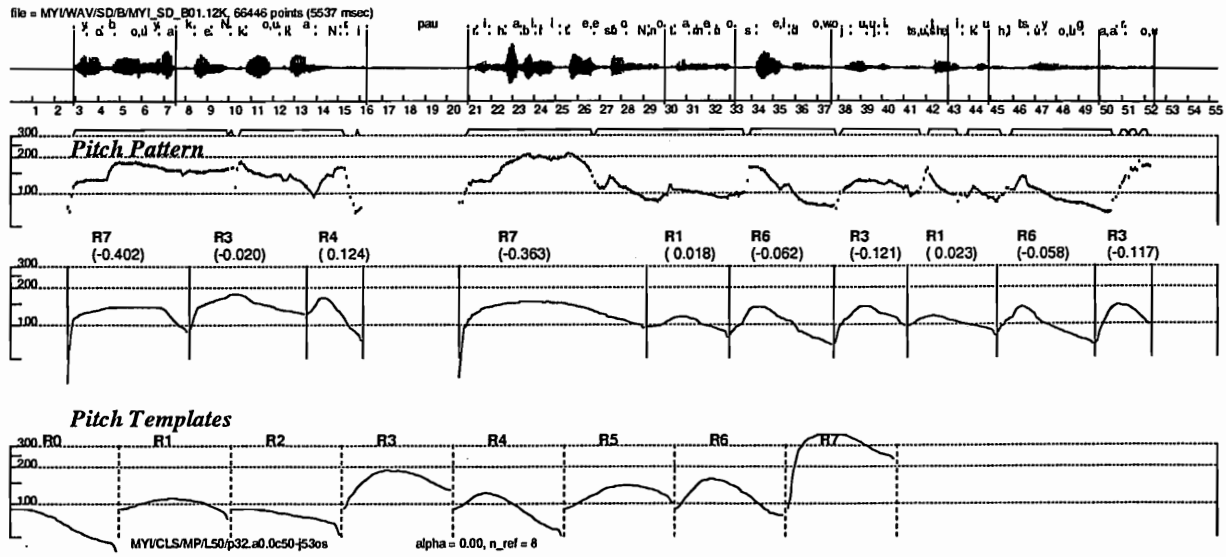


図 4.4: 高さ可変テンプレート (B) による句境界検出例

るパタンの 2 種類を備えていなければならず、図 4.2の場合だと、テンプレート数 4 以上になってようやく評価できるようになる。

4.2.4　高さ可変テンプレートによる句境界検出実験

ここで、前節の 2 つの高さ可変テンプレートによる句境界検出実験を行なう。句境界実験条件は 2.4 と同様である。ただし、デルタピッチの寄与は考えないことにする。(仮にデルタピッチの寄与を考えるならば、デルタピッチの距離尺度は従来通りの絶対的な値によって行なう。) また、本手法ではピッチパタンの連続性が非常に重要であるので、未知入力ピッチパタンについては第 3章で作成し、最も精度の高かった**複数周波数帯域候補による DP 連続パタン**を用いる。

実験結果は句境界検出率を図 4.5および表 4.2に、句境界挿入誤りを図 4.6および表 4.3に示した。図中の “◇” は従来の句境界検出の結果、“◆” は**高さ可変 (A)** による句境界検出の結果である。また、**高さ可変 (B)** の結果については表にのみ記してある。前節で指摘したように相対的なアクセントパタンは少数のテンプレートではピッチパタン全体を表現することは出来ず、テンプレート数が 1 や 2 のときは検出率が低い。**高さ可変 (B)** においてはテンプレート数 1 ではわずか 53.650% である。しかし、テンプレート数を増すにつれて高さ固定テンプレートの句境界検出率が頭打ちになるのに対し、高さ可変テンプレートによる句境界検出率は著しく増加する。同じ条件 ($\alpha = 0.0$) で比較すればテンプレート数 8

表 4.1: 高さ可変テンプレートによって自動検出された句境界総数

テンプレート数	1	2	4	8	16	32
視察による検出	904					
高さ固定 (α=0.0)	846	951	979	979	1001	1032
(0.5)	873	924	967	987	1004	1012
(1.0)	918	965	991	1035	1042	1059
高さ可変 (A)	951	1014	1029	1045	1070	1072
高さ可変 (B)	651	943	989	1017	1105	1092

以上では高さ可変の方が優れていることが分かるであろうし、デルタピッチを寄与した場合 ($\alpha = 0.5, 1.0$) と比較しても従来のものが 2 次元のマトリクスパタンであることを考慮すれば、本手法の方が約 1/2 の計算時間で済むわけであるし、単位時間あたりの効率では決して劣るものではないことが分かる。また句境界挿入誤りを見れば、高さ可変テンプレートの方はほぼ一定に 30%前後の挿入誤りがあることが分かる。これは、良い結果とは言えないがテンプレート数の少ないときに検出率が低い原因が**句境界検出能力** (正解、不正解に関わらない検出) の低さによるものでは無く、挿入誤りによるものであることが分かる。実際、正解、不正解を含めた句境界検出総数では表 4.1のように高さ可変の方が多く、句境界検出能力の高さを表わしている。

4.2.5　まとめ

この節では、アクセントパタンの相対的な変化に着目してクラスタリング分類し、絶対的なアクセントパタンでは表わせなかったアクセント成分的なパタンを表現した。また相対的なパタンが高さ方向に自由度を与えたテンプレートとして句境界検出に有効であることを示した。高さ方向への移動距離を決定する基準として、分析フレームのピッチの高さとテンプレートの接続境界における連続性の 2 つの手法を提案し、それぞれに一長一短があることも示した。また共通の問題として、検出能力が高いにも関わらず、その多くが挿入誤りとして検出され、句境界検出率の低下の原因となっていることも挙げられる。

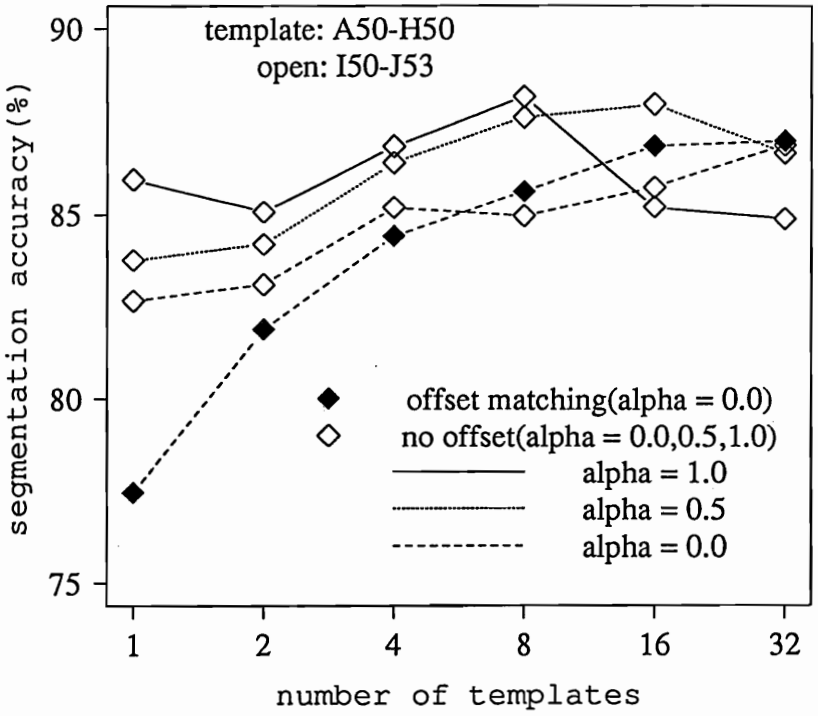


図 4.5: 高さ可変テンプレートによる句境界検出率

表 4.2: 高さ可変テンプレートによる句境界検出率

テンプレート数	1	2	4	8	16	32
高さ固定 ($\alpha = 0.0$)	82.633	83.075	85.177	84.956	85.730	86.836
(0.5)	83.739	84.181	86.394	87.611	87.942	86.615
(1.0)	85.951	85.066	86.836	88.164	85.177	84.845
高さ可変 (A)	77.434	81.858	84.403	85.619	86.836	**86.947**
(B)	53.650	83.850	83.407	85.066	86.283	84.292

(単位：%)

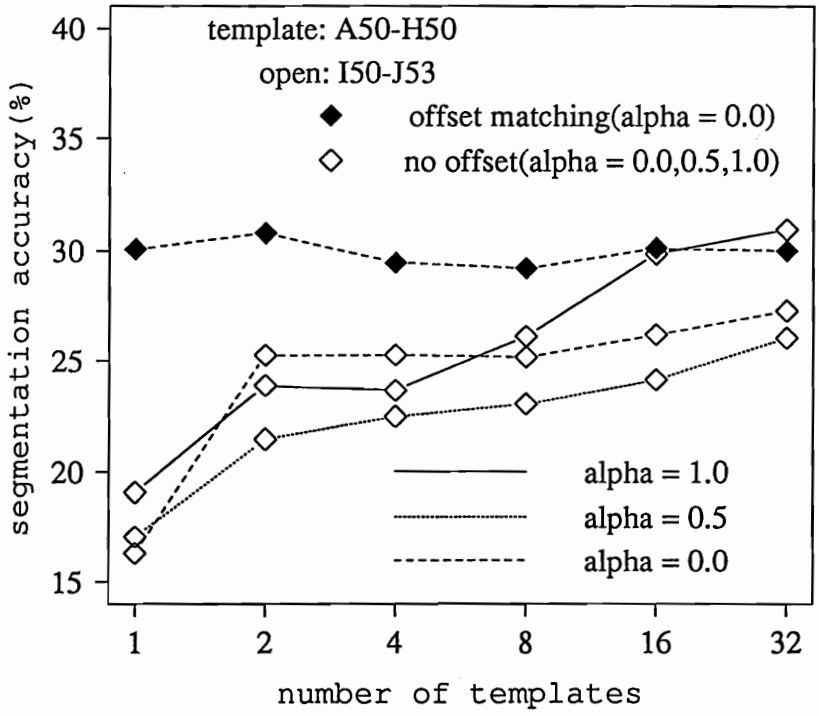


図 4.6: 高さ可変テンプレートによる句境界挿入誤り率

表 4.3: 高さ可変テンプレートによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
高さ固定 ($\alpha = 0.0$)	16.312	25.237	25.230	25.128	26.174	27.229
(0.5)	17.068	21.429	22.441	22.999	24.104	25.988
(1.0)	19.063	23.834	23.613	26.087	29.846	30.878
高さ可変 (A)	30.074	30.769	29.446	29.187	30.093	29.944
(B)	28.725	22.694	27.300	27.827	32.489	33.425

(単位：%)

4.3　ピッチテンプレートの遷移確率を考慮した句境界検出

4.3.1　まえがき

前節ではテンプレートの高さを可変にすることにより個々のテンプレートの表現力を上げ、従来法を凌ぐ個数の句境界を検出することが可能になった。その結果、検出率を向上させることができたが、反面、句境界挿入誤りをも増加させることになった。高さ可変テンプレートによる句境界検出の場合には常に 3 割程度の挿入誤りが生じるため、自動検出句境界数が少なくなるような低いレベルのテンプレート (テンプレート数 1、2 など) では必然的に句境界検出率が低下する。そこで、この節ではテンプレートの遷移情報を導入して、テンプレート系列の接続に関する制限を与えることにより、挿入誤りを回避することを試みる。

4.3.2　ピッチテンプレートの遷移確率

表 4.4: ピッチテンプレートの遷移確率

	R0	R1	R2	R3	R4	R5	R6	R7
初期状態	0.02	0.14	0.06	0.16	0.15	0.19	0.23	0.06
R0	0.07	0.19	0.14	0.12	0.14	0.18	0.14	0.02
R1	0.12	0.19	0.24	0.04	0.16	0.14	0.10	0.01
R2	0.09	0.22	0.13	0.07	0.13	0.17	0.17	0.01
R3	0.20	0.20	0.26	0.02	0.13	0.10	0.09	0.00
R4	0.03	0.24	0.11	0.08	0.14	0.20	0.17	0.03
R5	0.21	0.16	0.30	0.02	0.15	0.07	0.08	0.00
R6	0.04	0.20	0.20	0.05	0.15	0.19	0.16	0.00
R7	0.10	0.24	0.20	0.06	0.10	0.22	0.06	0.04

表 4.4は 4.2.2節で作成した高さ可変テンプレートをもとに、学習データ C～J403 文章の 2695 個のアクセントパタンを R0～R8 に量子化し、2 つのパタン間の遷移確率についてまとめたものである。各行はパタンの遷移前の状態、各列は遷移後の状態である。それぞれの行から各列への遷移確率で表されているので、各行の合計は 1 である。また、初期状態は無音状態のことであり、各文の発声開始前やポーズの状態である。

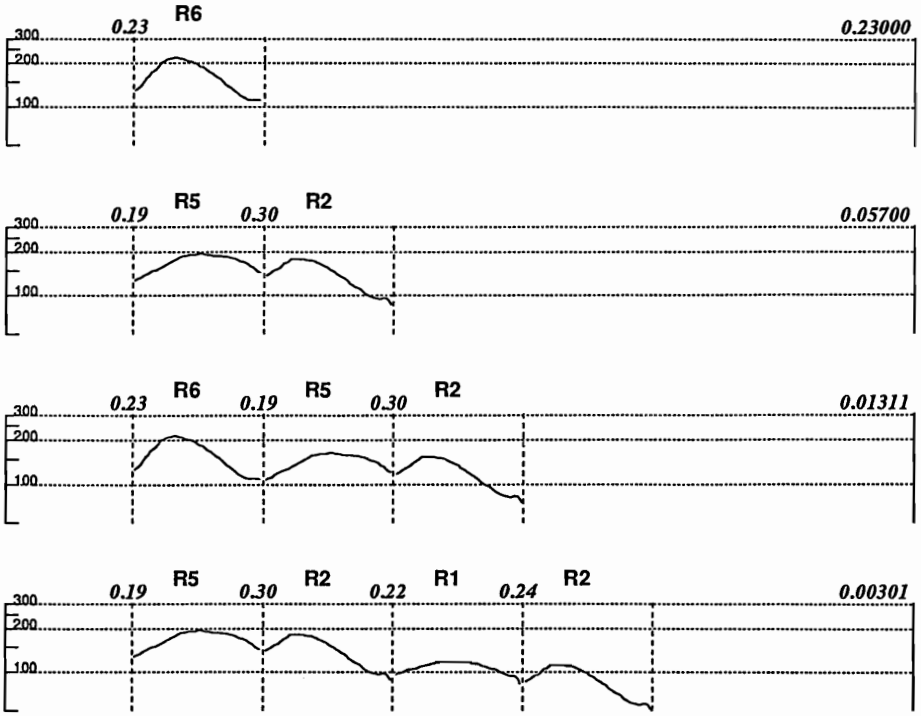


図 4.7: 遷移確率によるテンプレートパタンの接続

表を見ても分かるようにパタン間の結合には偏りがあり確率 3 割で遷移する例もあれば、全く 0 のものもある。例えば、R7 は発声開始直後に稀に現れる特殊なパタンであり、いずれの状態からも遷移しにくいことが分かる。また比較的よく現れるパタンでは R2、R6 のパタンがあり、他のパタンから遷移し易いことが分かる。

図 4.7はテンプレート数 8 個の例において 1 から 4 個でテンプレート系列を作成する場合の確率の高い組み合わせについて作図したものである。(パタンの高さは任意であるが、接続の高さは境界連続を基準にしてある。) 1 個の場合、初期状態からの遷移確率の最も高い R6 になる。このパタンは比較的立ち上がりが急で、約 1/3 でピークを迎えて、あとは緩やかに傾斜していく典型的なパタンである。2 個で構成した場合、前半のパタンは始端より終端の方がわずかに上がり、後半のパタンは緩やかに下がるパタンである。このとき、1 つ目のピークの方が 2 つ目のピークよりも高くなるパタンになる。このようにして 4 個の系列を求めた場合、全体的に始まりが高く、時間方向に経過するにしたがい下降していくような擬似的なピッチパタンとなる。

4.3.3　遷移確率付きテンプレートによる句境界検出

この節では、テンプレートの遷移確率をもとに、生起確率の低いテンプレート間の接続を抑制することで、テンプレート系列の最適化を試みる。ただし、この句境界検出法も One-Stage DP 上で実現可能なことを前提にする。したがって、遷移確率を DP 距離計算の重みとして乗ずる場合は、時間軸方向に向かって正方向の確率を考え、2 状態のパタンの遷移で行なうのが比較的簡単である。

今、未知入力ピッチパタンを構成する部分パタン (アクセント候補パタン) の系列を $S(S_1, S_2, S_3, \cdots, S_N)$ とし、これに整合するテンプレートパタンの系列を $T(T_1, T_2, T_3, \cdots, T_N)$ とする。ここで、Rをテンプレートの集合、$D(S_n, R_k)$ を部分パタン S_nとテンプレートパタン R_kの距離とすれば、S_nと整合するテンプレートは

$$T_n = \arg \min_{R_k \in R} [D(S_n, R_k)] \tag{4.1}$$

で定義される。$n-1$ 番目のテンプレートパタンT_{n-1}から n 番目のパタンT_nに接続するときの重み係数を $W_{n-1,n}$とし、S_nと T_nの距離を D_nとすれば、2 つのパタン系列間の合計距離は

$$D(S,T) = \sum_{n=1}^{N} \{W_{n-1,n} \times D_n\} \tag{4.2}$$

である。ただし、$W_{0,1}$は初期状態から T_1への重み係数である。したがって最適なピッチテンプレート系列は

$$T^* = \arg \min_{T} [D(S,T)] \tag{4.3}$$

であり、これに対応するアクセント候補パタンの系列

$$S^* = \arg \min_{S} [D(S,T)] \tag{4.4}$$

の境界が自動検出句境界である。以後、この手法を「**遷移確率付きテンプレートによる句境界検出法**」と呼び、「**TR-Matching 法**」と略称する。以下に**高さ可変 (A)** を例に、テンプレートの結合の重み係数をかけた句境界検出「**遷移確率付き (A)**」のアルゴリズムを示す。

[アルリズム]

未知入力パタンのフレーム：$i = 1, \cdots, N$

テンプート番号：$k = 1, \cdots, K$

テンプレート kのフレーム：$j = 1, \cdots, J_k$

(i, j, k) におけるオフセット (高さ方向の対数移動距離)：$O(i, j, k)$

(i,j,k) における累積距離 : $D(i,j,k)$

(i,j,k) におけるフレーム間距離 : $d(i,j,k,O)$

テンプレート k^*から kへの重み係数 : $W(k^*,k)$

対数ピッチ値を $F_i(i)$

テンプレート番号 kにおけるフレーム jの対数ピッチ値を $F_{tk}(j)$

とするとき、次のように定義する。

$$d(i,j,k,O) = (F_i(i) - (F_{tk}(j) + O))^2$$

Step1 initialize $D(1,j,k) = \sum_{n=1}^{j} d(1,n,k)$

Step2 (a) for $i := 2$ to N do steps (b) - (e)

(b) for $k := 1$ to K do steps (c) - (e)

(c)
$$\begin{aligned} k^* &= \arg \min_{k'=1,\cdots,K} [D(i-1,J_{k'},k')] \\ O(i,1,k) &= F_i(i) - F_{tk}(1) \\ D(i,1,k) &= W(k^*,k) \times d(i,1,k,O(i,1,k)) + D(i-1,J_{k^*},k^*) \\ &= 0 + D(i-1,J_{k^*},k^*) \end{aligned}$$

(d) for $j := 2$ to J_k do step (e)

(e)
$$\begin{aligned} O_{11} &= O(i-1,j-1,k) \\ O_{12} &= O(i-1,j-2,k) \\ O_{21} &= O(i-2,j-1,k) \\ O_{22} &= O(i-2,j-2,k) \\ D_{11} &= W(k^*,k) \times d(i,j,k,O_{11}) + D(i-1,j-1,k) \\ D_{12} &= W(k^*,k) \times d(i,j,k,O_{12}) + D(i-1,j-2,k) \\ D_{21} &= W(k^*,k) \times d(i,j,k,O_{21}) + W(k^*,k) \times d(i,j,k,O_{21}) \\ &\quad + D(i-2,j-1,k) \\ D_{22} &= W(k^*,k) \times d(i,j,k,O_{22}) + W(k^*,k) \times d(i,j,k,O_{22}) \\ &\quad + D(i-2,j-2,k) \\ (x^*,y^*) &= \arg \min_{x=1,2,y=1,2} [D_{xy}] \\ D(i,j,k) &= D_{x^*y^*} \\ O(i,j,k) &= O_{x^*y^*} \end{aligned}$$

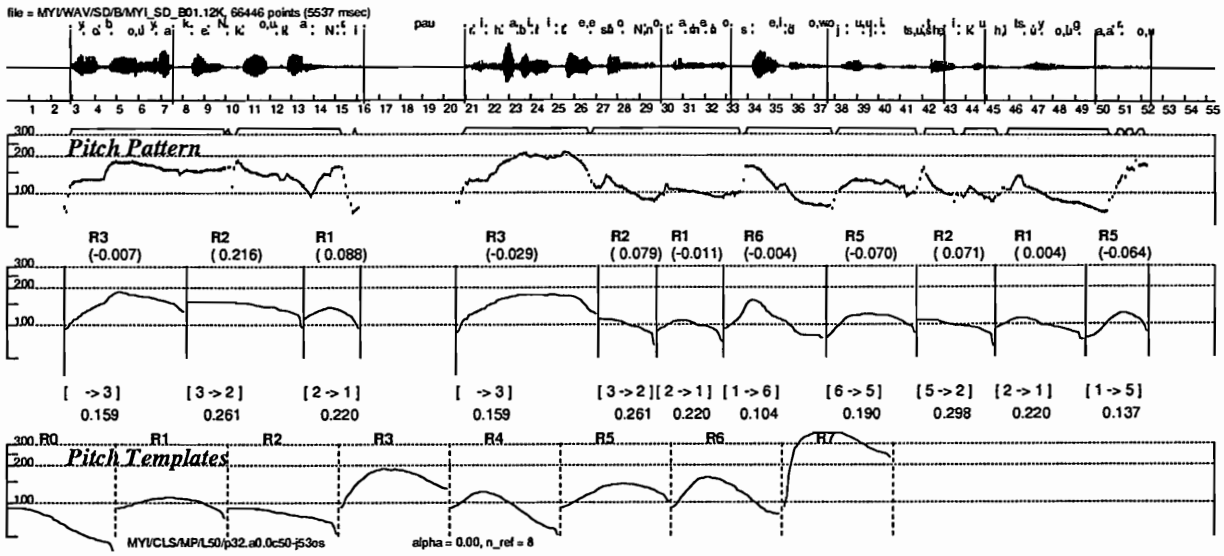


図 4.8: 遷移確率付きテンプレート (A) による句境界検出例

Step3 Trace back the best path

同様に**高さ可変 (B)** に遷移確率を導入した句境界検出法「**遷移確率付き (B)**」においては (c) の過程が次のように変更される。

Step2 (c)

$$
\begin{aligned}
k^* &= \arg \min_{k'=1,\cdots,K} [W(k',k) \times d(i,1,k,O(i,1,k')) \\
&\qquad + D(i-1,J_{k'},k')] \\
&\qquad (O(i,1,k') = F_{tk'}(J_{k'}) + O(i-1,J_{k'},k') - F_{tk}(1)) \\
O(i,1,k) &= F_{tk^*}(J_{k^*}) + O(i-1,J_{k^*},k^*) - F_{tk}(1) \\
D(i,1,k) &= W(k^*,k) \times d(i,1,k,O(i,1,k)) + D(i-1,J_{k^*},k^*)
\end{aligned}
$$

図 4.8は遷移確率付き (A) テンプレートによる句境界検出例、図 4.9は遷移確率付き (B) テンプレートによる句境界検出例である。自動検出境界の下に記された $[i \to j]$ の表示とその下の数値はテンプレート i から jへの遷移確率を示している。それぞれ、図 4.3、図 4.4 と比較することにより、遷移情報の効果が分かる. この例では、重み係数の値として遷移確率の対数の絶対値 (ただし確率 0 や 1 の場合は危険なので全体的に補正したもの) を用

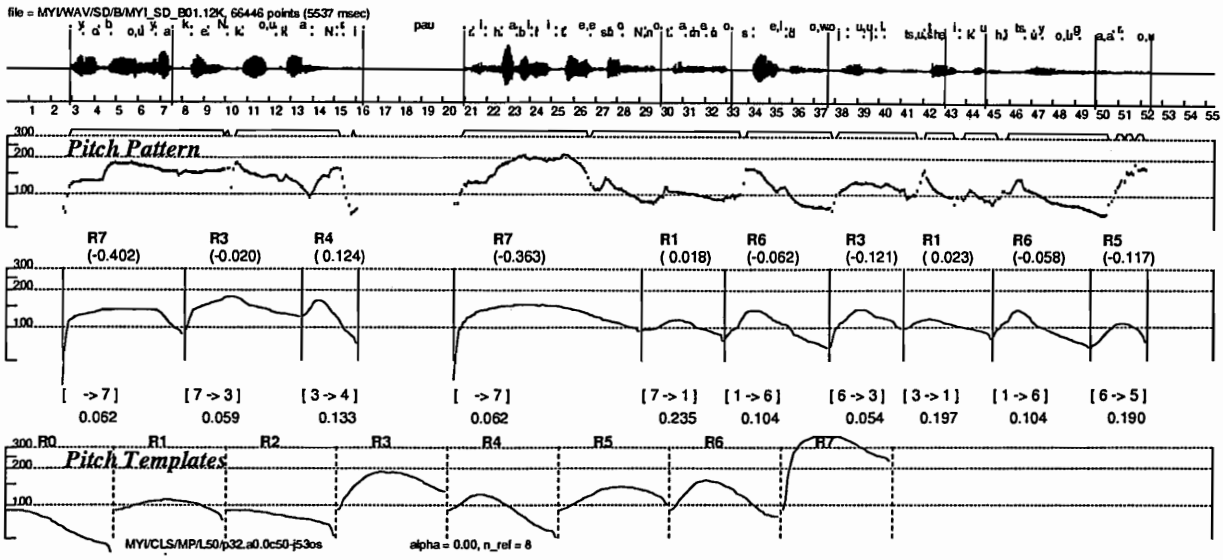


図 4.9: 遷移確率付きテンプレート (B) による句境界検出例

いたので、重みの隔差が小さく、あまり大きな違いは現れなかった。しかし、**遷移確率付き (A)** の場合は R5 → R1 → R1(確率 0.16 × 0.19) であったものが、R5 → R2 → R1(確率 0.30 × 0.22) に修正された。また、**遷移確率付き (B)** でも文末で R6 → R3(確率 0.05) であったものが R6 → R5(確率 0.19) に修正された。しかし、**遷移確率付き (B)** のテンプレートの接合では、確率 0.06 前後の遷移が抑制されずに頻繁に生じているのでまだまだ重み係数について検討が必要である。

4.3.4　遷移確率付きテンプレートによる句境界検出実験

ここで、**遷移確率付き (A)**、**遷移確率付き (B)** の両手法による句境界検出実験を行なう。句境界実験の詳細は 4.2.4節と同様である。テンプレート k^*から kへの遷移確率 $P(k^*, k)$ より、重み係数として次のものを用いる。

$$W(k^*, k) = \left|\log\left\{(1-\gamma)P(k^*, k) + \gamma\right\}\right|, \qquad (\gamma = 10^{-6}) \tag{4.5}$$

ただし、学習データが 2695 個しかないので、テンプレート数が 16、32 になるにつれて、遷移確率が 0 の状態が増えるので、個々の遷移確率に影響を及ぼさない範囲の微少な値 γ を用いて補正した。

実験結果は句境界検出率を図 4.10 および表 4.7に、句境界挿入誤りを図 4.11 および表 4.8に示した。図中の "◆" は**高さ可変 (A)**(遷移確率なし) の句境界検出、"□" は**遷移確率**

表 4.5: 遷移確率付きテンプレートによって自動検出された句境界総数

テンプレート数	1	2	4	8	16	32
視察による検出	904					
高さ可変 (A)	951	1014	1029	1045	1070	1072
遷移確率付き (A)	951	1012	1030	1046	1069	1075
高さ可変 (B)	651	943	989	1017	1105	1092
遷移確率付き (B)	651	940	987	1017	1102	1092

表 4.6: 遷移確率付きテンプレートによって正解検出に転じた句境界数

テンプレート数	1	2	4	8	16	32
遷移確率付き (A)	0(0)	2(2)	3(2)	4(0)	3(3)	4(1)
遷移確率付き (B)	0(0)	3(7)	4(5)	1(2)	4(4)	2(6)

(括弧内は不正解に転じた数)

付き (A) の句境界検出の結果である。遷移確率付き (B) については表に記した通りである。テンプレート数 1 のときは当然同じ結果であり、テンプレート数を増すにつれて句境界検出率に差が見られる。全般的に、句境界検出率がわずかに上がり、句境界挿入誤りが少し下がる傾向にある。しかし、表 4.6のように不正解から正解に転じたものと正解から不正解に転じたものが存在し、わずかに正解の方が増えただけである。また、遷移確率付き (B) については不正解の方が上回った。これは、先に述べたように確率重み係数の値が遷移を抑制できるほど強い値に設定されていなかったためであると考えられ、正解から不正解へ、あるいはその逆に転じた個数自体が比較できるほど多くなかったためである。

4.3.5　まとめ

この節では句境界検出にテンプレートパタンの遷移情報を導入した。この手法の概要は 2 状態の遷移について正方向に確率重み係数を乗じるものであり、比較的簡単に One-Stage DP 上で構成可能である。これにより、句境界検出の挿入誤りを抑制し、生成確率的に最適なテンプレート系列を作成できるという可能性を示した。ただし、本実験では確率重み係数を 1 通りしか試みていないので、その最適化が課題として残されている。

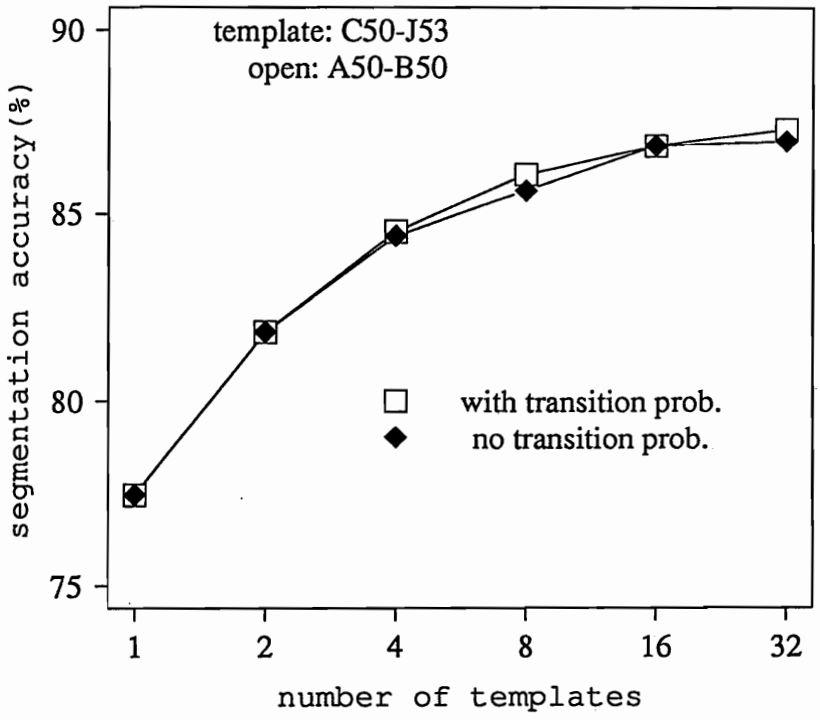


図 4.10: 遷移確率付きテンプレートによる句境界検出率

表 4.7: 遷移確率付きテンプレートによる句境界検出率

テンプレート数	1	2	4	8	16	32
高さ可変 (A)	77.434	81.858	84.403	85.619	86.836	86.947
遷移確率付き (A)	77.434	81.858	84.513	86.062	86.836	**87.279**
高さ可変 (B)	53.650	83.850	83.407	85.066	86.283	84.292
遷移確率付き (B)	53.650	83.407	83.296	84.956	86.283	83.850

(単位：%)

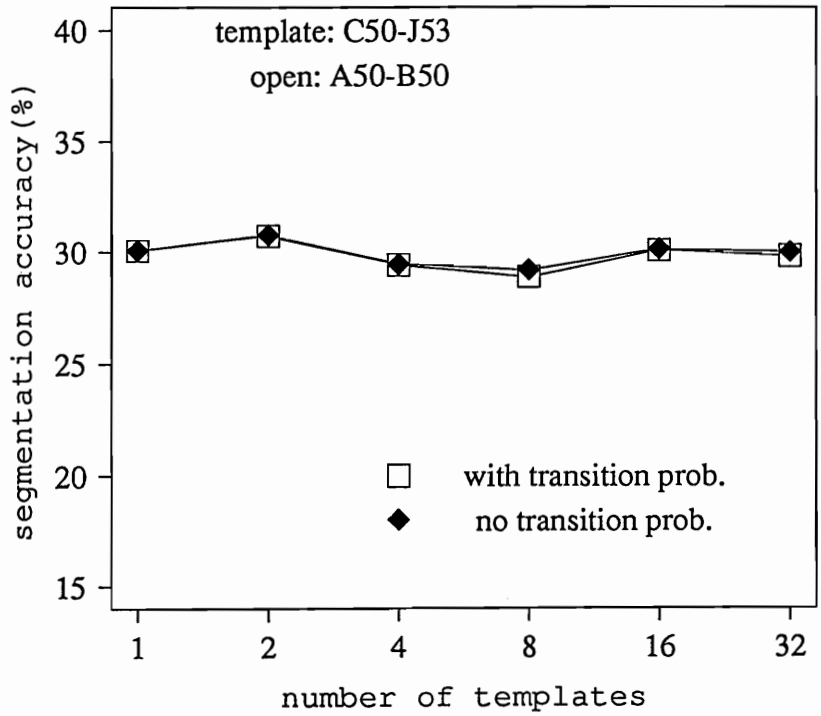


図 4.11: 遷移確率付きテンプレートによる句境界挿入誤り率

表 4.8: 遷移確率付きテンプレートによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
高さ可変 (A)	30.074	30.769	29.446	29.187	30.093	29.944
遷移確率付き (A)	30.074	30.731	29.417	28.872	30.028	29.767
高さ可変 (B)	28.725	22.694	27.300	27.827	32.489	33.425
遷移確率付き (B)	28.725	22.766	26.950	28.024	32.305	33.700

(単位：%)

4.4　まとめ

本章では次の 2 つの句境界検出による高精度化を図った。

1. アクセントパタンを始点からの相対的な変化で分類することにより、絶対的なアクセントパタンに比べてより多くのパタンの表現を可能にした。これを高さ可変テンプレートとしてパタンの変化に着目した整合を行うことで、句境界検出の高精度化 (句境界検出率の向上) を実現した。課題としては、挿入誤りの抑制が上げられる。

2. また、遷移確率付きテンプレートによってパタンの結合を制御することにより、生成確率の高いテンプレート系列を求め、句境界挿入誤りの訂正による句境界検出率の向上の可能性を示した。ただし、確率重み係数の最適化などの問題が残されている。

第 5 章

複数時間長テンプレートへの拡張

5.1 はじめに ～固定長テンプレートの限界～

2.5節で挙げた 3 番目の問題点として、固定長テンプレート (単一時間長テンプレート) での句境界検出の限界がある。これまでのように、テンプレート長をアクセントパタンの平均長 500ms でクラスタリング、かつ One-Stage DP の傾斜制限を 1/2～2 とした場合、1000ms 以上の長いパタンには必然的に句境界挿入誤りが生じ、逆に正解句境界領域を含めても 250ms に満たないような短いパタンでは、少なくとも始まりか終わりのどちらか一方の句境界は検出不可能になる。これを可能にする手段として、まず制限を緩めることが考えられる。具体的にはテンプレート長を短く (250ms) して、DP の傾斜制限を広く (0～2) すればパタンの長短に関係無く、整合が可能となるはずである。このときの句境界検出例が図 5.1であり、それぞれのテンプレートが 125ms～∞の範囲で時間伸縮可能である。サンプリング周期を 10ms としているのでテンプレート長は 25 フレームしかなく、図から見てとれるように、テンプレートの作成においてすでに量子化の粗さが目立つ。また、傾斜が 0 のため無限大までの伸縮が可能であるので、整合のイメージはステップ状になり、ピッチの高さの等しいところはどこまでもテンプレートの一点で整合する。例えば図中の 1 番目の句境界は、その前後がほぼ平坦であるのでほとんど検出され得ない状態である。

このように、

1. テンプレート長 500ms で DP 傾斜制限 1/2～2
2. テンプレート長 250ms で DP 傾斜制限 0～2

のいずれの場合でも全ての句境界を検出するのは不可能である。そこで、本章では複数の異なる長さのテンプレートを準備することでこれまで検出されなかった句境界の検出を試

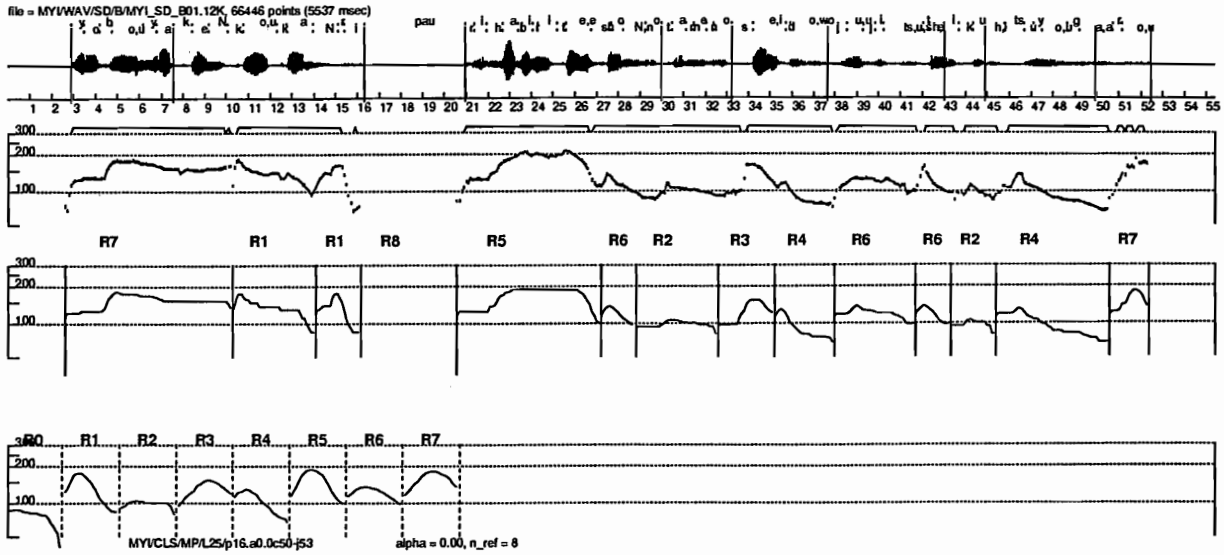


図 5.1: テンプレート長 250ms、DP 傾斜制限 0～2 による句境界検出

みる。また、複数時間長のテンプレートを準備するにあたって、これまでに作成した 500ms のテンプレートと 250ms のテンプレートを混合するのではなく、ひとつのクラスタリング分類の過程で複数の長さのテンプレートを導出する方法についても検討する。これは異なる時間長のテンプレート間の遷移確率を推定し易くするためであり、第 4 章で提案した遷移確率付きテンプレートによる句境界検出を可能にするためである。

5.2　複数時間長テンプレートの学習

ここでは、**複数時間長テンプレート**の学習法について述べる。これまではアクセントパタンの形状についてのクラスタリングであったが、これにパタンの時間長に関する距離尺度を結合すれば従来と同じく LBG 法で容易に分類することができる。ただし、異なる長さでのパタンの形状距離を定義するのは難しいので、これまでと同様に各アクセントパタンを L フレームに線型伸縮変換した後、クラスタリングすることにする。

以下、2 つのアクセントパタン P_1、P_2の距離を例に考えることにする。長さ L_1のパタン $P_1=(p_{1i}, \cdots, p_{1L_1})$ と長さ L_2のパタン $P_2=(p_{2i}, \cdots, p_{2L_2})$ の変換されたパタンをそれぞれ $\hat{P}_1=(\hat{p}_{1i}, \cdots, \hat{p}_{1L}, L_1)$、$\hat{P}_2=(\hat{p}_{2i}, \cdots, \hat{p}_{2L}, L_2)$ とする。変換後のアクセントパタンの形状は L フレームのベクトルで表現されるが、もとの長さに関する情報を保存するために、$L+1$

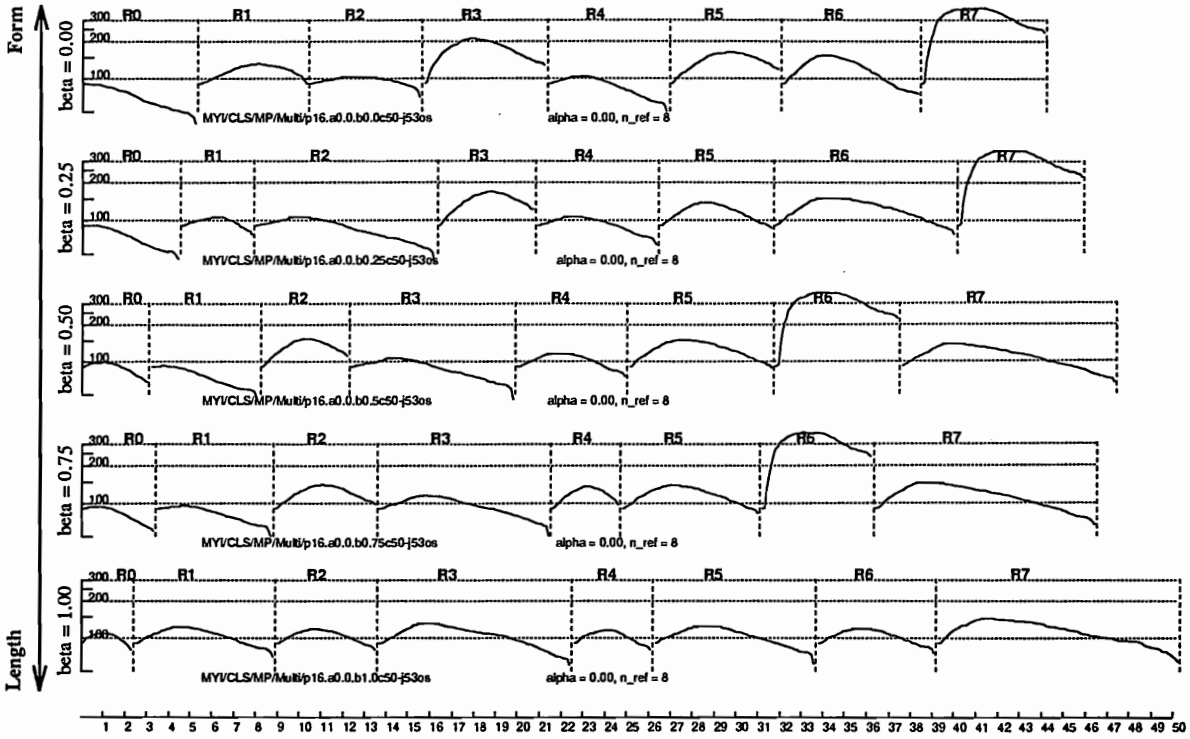


図 5.2: 複数時間長テンプレート

番目の要素として加える。このとき、形状に関する距離 (**パタン形状距離**) は

$$D_F(\hat{P}_1, \hat{P}_2) = \sum_{i=1}^{L} (\hat{p}_{1i} - \hat{p}_{2i})^2 \tag{5.1}$$

であるし、また長さに関する距離 (**パタン長距離**) は

$$D_L(\hat{P}_1, \hat{P}_2) = (L_1 - L_2)^2 \tag{5.2}$$

である。これにより、パタン間の総合距離 (**パタン間距離**) をパタン形状距離とパタン長距離の結合により次のように定義する。

$$\begin{aligned} D_\beta(\hat{P}_1, \hat{P}_2) &= (1-\beta) D_F(\hat{P}_1, \hat{P}_2) + \beta C_L D_L(\hat{P}_1, \hat{P}_2) \\ &= (1-\beta) \sum_{i=1}^{L} (\hat{p}_{1i} - \hat{p}_{2i})^2 + \beta C_L (L_1 - L_2)^2 \end{aligned} \tag{5.3}$$

ここでβはパタン間距離全体に対する**パタン長距離** (D_L) **の寄与率**である。またC_LはD_Lを正規化する係数であり、

$$C_L = \frac{\sum_{\hat{P}_n \in \hat{P}} D_F(\hat{P}_n, \bar{P})}{\sum_{\hat{P}_n \in \hat{P}} D_L(\hat{P}_n, \bar{P})} \tag{5.4}$$

で定義する。ただし、$\hat{P}$は線型変換された全学習アクセントパタン集合であり、$\hat{P}_n$は個々のアクセントパタンである。また$\bar{P}$は線型変換された全学習アクセントパタンの平均パタン、つまり

$$\begin{aligned}\bar{P} &= E[\hat{P}] \\ &= (E[\hat{p}_{n1}], \cdots, E[\hat{p}_{nL}], E[L_n]) \end{aligned} \tag{5.5}$$

である。

量子化数 m でクラスタリング分類した結果、m 個の集合クラスタ $C_1, \cdots, C_m$ $(C_n \in P)$ を形成したならば、それぞれの代表ベクトル$\bar{C}_n$を計算する。このとき、$L+1$ 番目の要素 $\bar{c}_{n,L+1}$は各クラスタに属するアクセントパタンもとの長さの平均値であるから、最終的に長さ L の平均パタン $(\bar{c}_{n1}, \cdots, \bar{c}_{nL})$ を長さ $(\mathrm{int})\bar{c}_{n,L+1}$に線型変換して異なる m 種類の長さのテンプレートを作成する。

図 5.2はアクセントパタンの時間長と形状の比重を変えてクラスタリングした結果である。上段のものほど形状に依存し、最上段 $(\beta = 0.00)$ は形状のみによるクラスタリング結果である。また下段のほものど時間長に依存し、最下段 $(\beta = 1.00)$ は時間長のみによるクラスタリング結果である。形状のみでクラスタリングした場合はどのテンプレートの時間長も (実際は多少異なるが) ほぼ等しく、時間長のみでクラスタリングした場合はいずれも始端よりも終端のほうが下降する典型的な“へ” の字型のアクセントパタンになり、時間長と形状が互いに独立の要素であることが分かる。

5.3　複数時間長テンプレートによる句境界検出

5.3.1　まえがき

この節では第 4 章で提案した高さ可変および遷移確率付きテンプレートによる句境界検出を複数時間長テンプレートに拡張して実験を行なう。これまでに述べてきた句境界検出のアルゴリズムはいずれも **Step2**(d) のループが j=1〜J_k (J_kはテンプレート kの長さ) で記述されているので、全く同じアルゴリズムで複数時間長テンプレートによる句境界検出が可能である。

5.3.2　句境界検出実験

句境界実験条件の詳細は 4.2.4節と同様である。ただし、実験上の都合によりテンプレート数は最大で 16 までとした。また、テンプレートとして全パタン間距離に対するパタン長

距離の寄与率がβ=0.00, 0.25, 0.50, 0.75, 1.00 の 5 種類ものを準備した。句境界検出法は**高さ可変 (A)(B)**、**遷移確率付き (A)(B)** の 4 種類について行なった。それぞれの結果をパタン長距離の寄与率βについてまとめる。またそれぞれの句境界検出率および句境界挿入誤り率について合計 8 つのグラフと表を章末の図 5.5-5.12、表 5.1-5.8 に示す。図 5.3は**複数時間長テンプレート・遷移確率付き (A)** の例である。上段からそれぞれ、β=0.00 から 1.00 までの 5 種類を比較している。同様に、図 5.4は**複数時間長テンプレート・遷移確率付き (B)** の例である。(注：ポーズ後の確率遷移確率が 0 になっているのは表示ミスです。)

5.3.3　実験考察

図 5.3あるいは図 5.4のβによる句境界検出結果を比較した場合、もし、視察句境界と自動検出句境界とが 1 対 1 で対応している方が好ましいのであれば、β=0.00 のものが最良と言えるであろう。しかし、β=0.00 はテンプレートの長さがいずれもほぼ等しい場合であるので、4300ms から 4500ms の区間で発声されている/i/-/k/-/u/ という 200ms 程度の小さなアクセントパタンは検出不可能であることが分かる。そのようなアクセントパタンもβ=0.25,0.50 では検出可能である。

また、ポーズ後の最初のパタンの例では、微少な勾配の変化から 3 つのパタンの組み合わせとして認識されている。これは機械的に判断した場合には挿入誤りとしてカウントされるが、時には、文意から判断できないような発声上のまとまりとして正しいものであるかもしれない。この例では「リハビリテーション」という発声を 3 分したもので、あまり意味的な区切りにはならなかったが、文末近くの「充実して」という 1 つのアクセントパタンを「充実」「して」という 2 つのパタンとして抽出することができた。このことから、同様にアクセント結合によって 2 つのアクセントパタンが結合していると視察で判断されたものでも、分離抽出の可能性があることが言える。

テンプレート長が 1 種類のとき (ここではβ=0.00 が最も近い) には**高さ可変 (A)** と**高さ可変 (B)** の句境界検出による差があまり見られなかったが、複数時間長テンプレートの場合では、その違いがはっきり現れている。**高さ可変 (A)** では、短いテンプレートが準備されたことにより、特にポーズ区間の後の第 1 パタンを短めに検出してしまいがちである。それに対し、**高さ可変 (B)** では接続境界での値を連続にしなければならないので、いくら短いテンプレートがあろうと、境界でのピッチの値との差が広がらないように、テンプレートの終端の高さにほぼ等しくなるまでのピッチパタンをひとつのパタンとしてとらえる。**高さ可変 (B)** のβ=1.00 では、いずれのテンプレートも始端より終端の方が低くなるので、テンプレート系列のピッチの下降を抑制するために、可能な限り少ない数の最も長いテン

プレートで整合している。全般的に、**高さ可変 (A)** の方は挿入誤りを犠牲にして句境界の検出率を高めていると言える。

実際、**高さ可変 (A)** では“へ”の字型の短いパタンを与えれば、どの高さのどんなパタンとでも整合可能になるので、句境界検出率、句境界挿入誤り率ともに高くなる。そこで遷移確率による短いテンプレート同士の結合を抑制することが非常に重要になってくる。しかし、β=0.25 の R1 → R1 やβ=0.50 の R0 → R0 などの短いテンプレート間の遷移確率が意外と高いことや、前章でも述べたように、遷移確率重み係数の最適化が未だ不十分で低い確率の遷移が頻繁に起きていることもあって、テンプレートの結合の抑制が十分に作用していない。

この実験で最も高い検出を挙げたものは**遷移確率付き (A)**、β=0.50、テンプレート数 16 で検出率 89.602% である。特に図 5.9 では、量子化ビット数が 1 増加するにしたがって、検出率が 3%づつ増加している。ただし、ほぼ同様に句境界挿入誤りの方も 3%づつ増加していることが図 5.10のグラフより分かる。検出率のグラフと違う点を挙げれば、句境界検出率がβ=0.50 以上で大差ないことに対して、挿入誤りがほぼβに伴なって増加している点である。つまり、この実験を通して言えることであるが、テンプレートを作成するときのパタン形状距離とパタン長距離の比率 $(1-\beta)$: βは $1:1(\beta=0.50)$ の近傍が最適である。つまり、テンプレートとして、形状に関する情報も、時間長に関する情報も同様に重要であることが言える。

5.4　まとめ

この章では、これまでの単一の長さのテンプレートによる句境界検出に対して、テンプレートの固有の長さ (アクセントパタンの平均長) とアクセントパタンの形状を保つための DP 整合の傾斜制限の組み合わせにより検出の限界が生じる点について指摘した。具体的には長パタンに対する挿入誤りの必然性、短パタンに対する句境界検出の不可能についてである。そこで、これに対処する方法として、複数の長さのテンプレートによる句境界検出法を提案した。また、そのテンプレートの作成のためにアクセントパタンの時間長と形状の結合距離関数を定義し、これまで通り LBG 法のクラスタリングで容易に作成できることを示した。これにより、異なる長さのテンプレート間でも遷移確率を推定できるようになり、その結果、前章までの全てのパタン連続整合による句境界検出法について、複数テンプレートへの拡張を可能にした。

この手法により、約 40%の挿入誤りで、89.602%の句境界検出率を実現した。

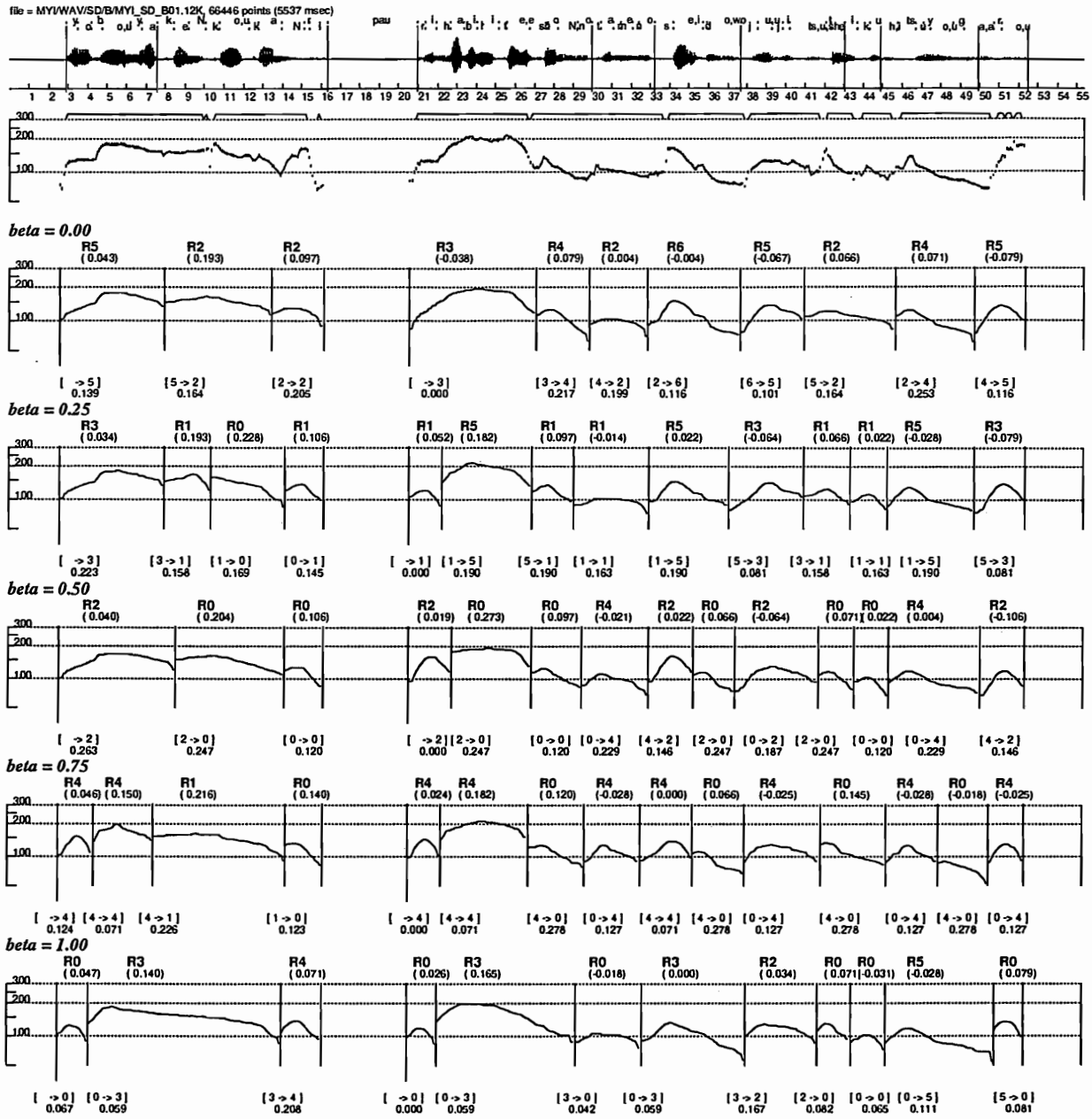


図 5.3: 複数時間長・遷移確率付きテンプレート (A) による句境界検出例

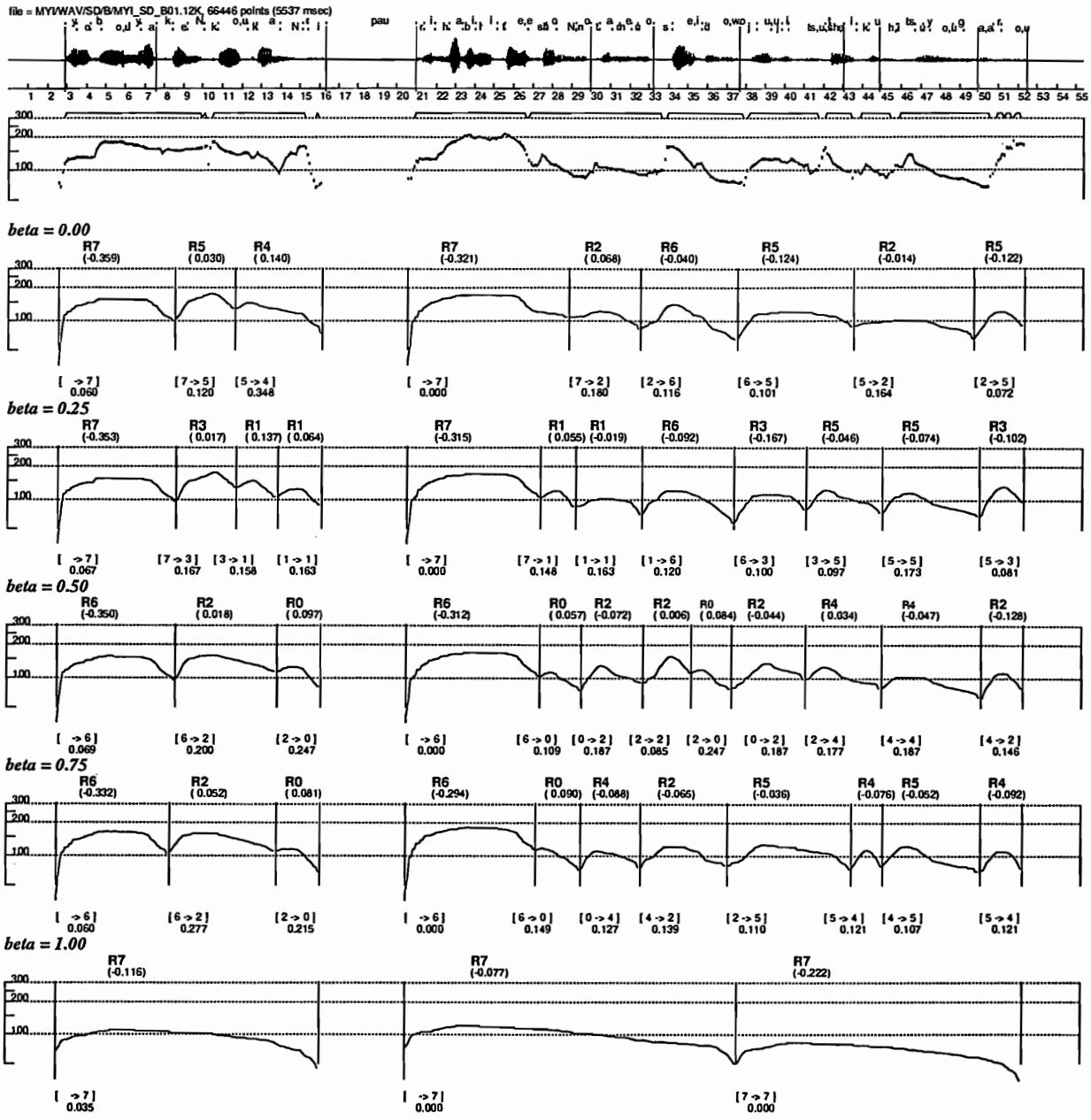


図 5.4: 複数時間長・遷移確率付きテンプレート (B) による句境界検出例

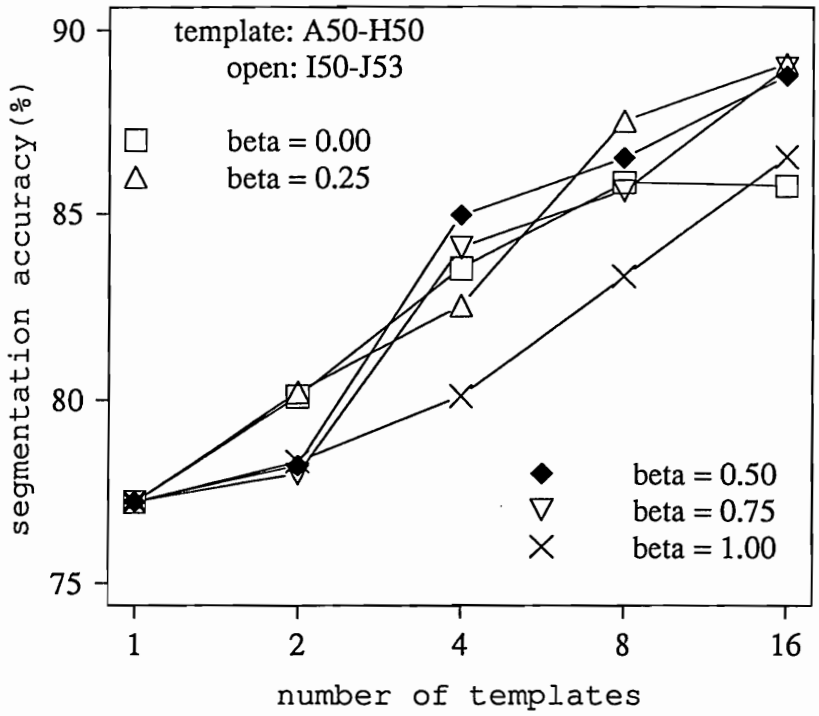


図 5.5: 複数時間長・高さ可変テンプレート (A) による句境界検出率

表 5.1: 複数時間長・高さ可変テンプレート (A) による句境界検出率

テンプレート数	1	2	4	8	16
高さ可変 (A)($\beta = 0.00$)	77.212	80.088	83.518	85.841	85.730
(0.25)	77.212	80.199	82.522	87.500	**89.049**
(0.50)	77.212	78.208	84.956	86.504	88.717
(0.75)	77.212	77.987	84.071	85.619	88.938
(1.00)	77.212	78.319	80.088	83.296	86.504

(単位：%)

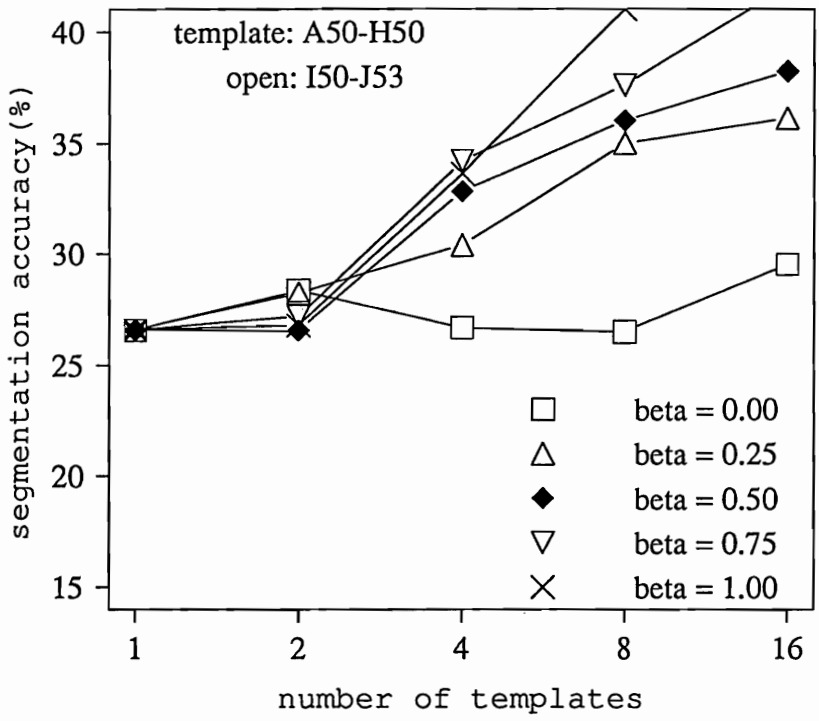


図 5.6: 複数時間長・高さ可変テンプレート (A) による句境界挿入誤り率

表 5.2: 複数時間長・高さ可変テンプレート (A) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
高さ可変 (A)($\beta = 0.00$)	26.556	28.347	26.660	26.488	29.496
(0.25)	26.556	28.213	30.392	35.004	36.095
(0.50)	26.556	26.528	32.847	36.031	38.186
(0.75)	26.556	27.174	34.208	37.605	42.027
(1.00)	26.556	26.768	33.653	40.990	47.091

(単位：%)

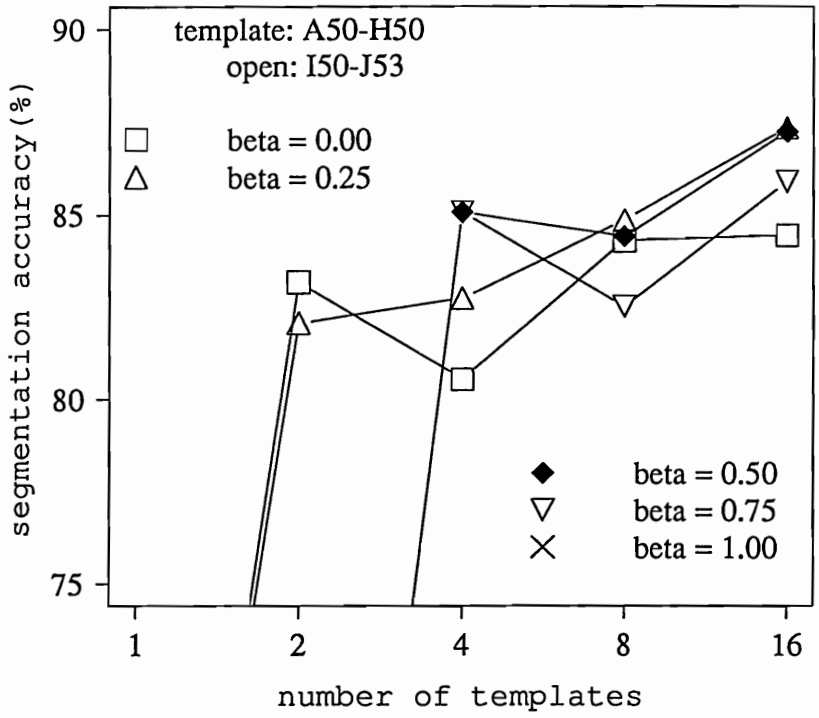


図 5.7: 複数時間長・高さ可変テンプレート (B) による句境界検出率

表 5.3: 複数時間長・高さ可変テンプレート (B) による句境界検出率

テンプレート数	1	2	4	8	16
高さ可変 (B)($\beta = 0.00$)	53.540	83.186	80.531	84.292	84.403
(0.25)	53.540	82.080	82.743	84.845	**87.279**
(0.50)	53.540	49.336	85.066	84.403	87.168
(0.75)	53.540	49.226	85.066	82.522	85.841
(1.00)	53.540	49.447	47.456	47.898	47.235

(単位：%)

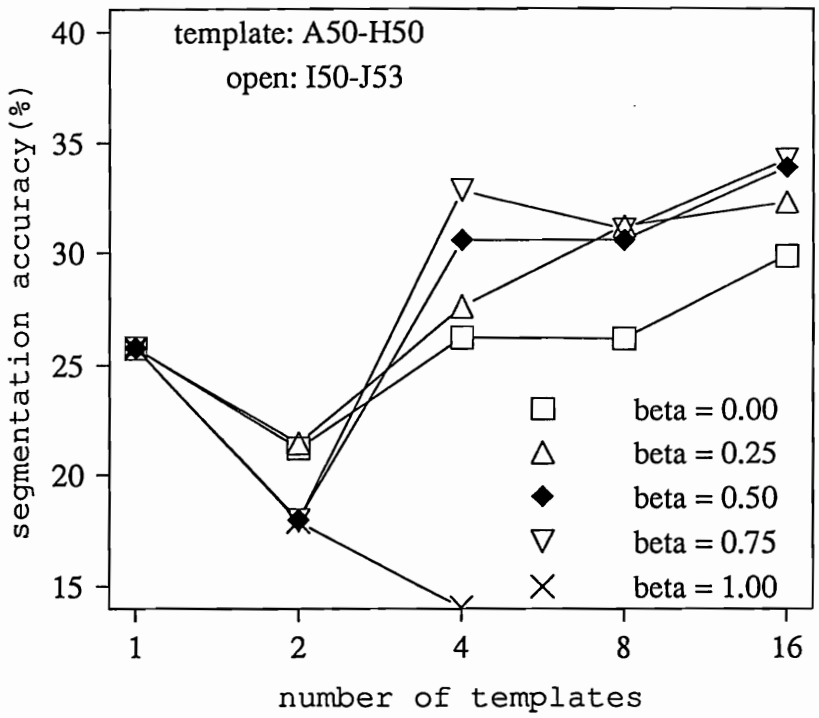


図 5.8: 複数時間長・高さ可変テンプレート (B) による句境界挿入誤り率

表 5.4: 複数時間長・高さ可変テンプレート (B) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
高さ可変 (B)($\beta = 0.00$)	25.755	21.192	26.216	26.171	29.789
(0.25)	25.755	21.429	27.604	31.180	32.203
(0.50)	25.755	17.939	30.561	30.550	33.798
(0.75)	25.755	17.939	32.818	31.014	34.157
(1.00)	25.755	17.871	14.050	9.111	9.471

(単位：%)

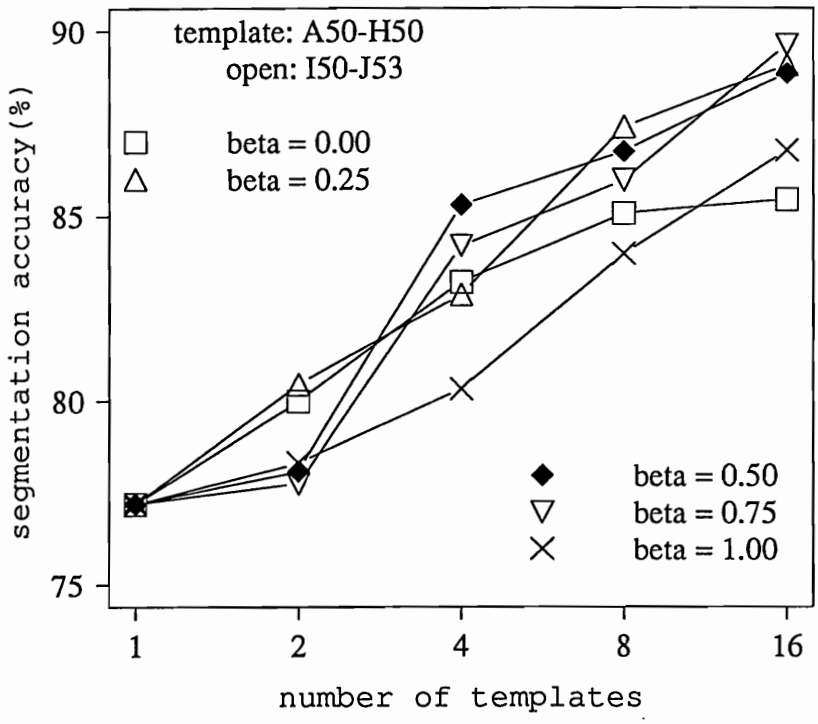


図 5.9: 複数時間長・遷移確率付きテンプレート (A) による句境界検出率

表 5.5: 複数時間長・遷移確率付きテンプレート (A) による句境界検出率

テンプレート数	1	2	4	8	16
遷移確率付き (A)($\beta = 0.00$)	77.212	79.978	83.186	85.066	85.398
(0.25)	77.212	80.420	82.854	87.389	89.049
(0.50)	77.212	78.097	85.288	86.726	88.827
(0.75)	77.212	77.765	84.181	85.951	**89.602**
(1.00)	77.212	78.319	80.310	83.960	86.726

(単位：%)

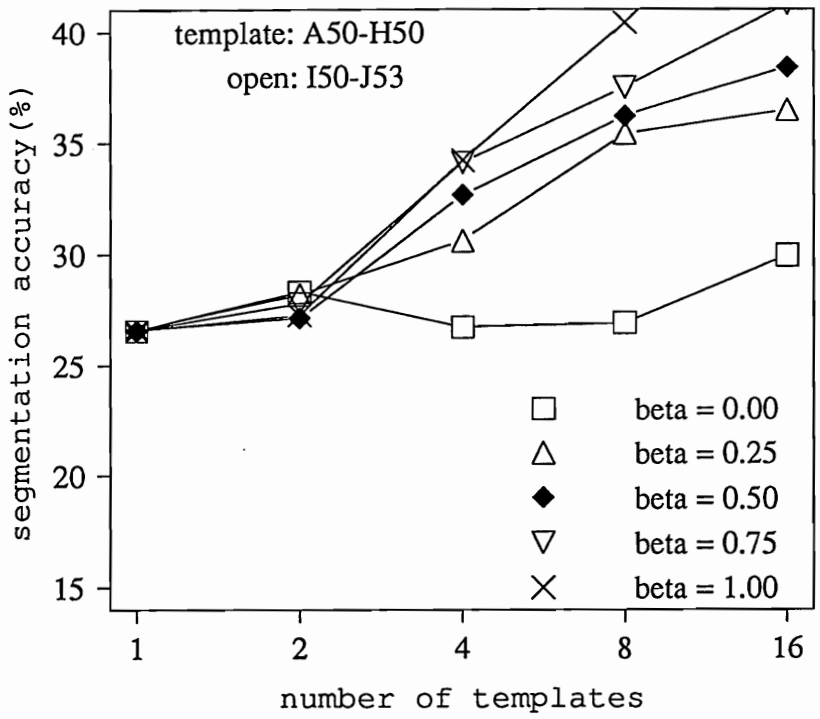


図 5.10: 複数時間長・遷移確率付きテンプレート (A) による句境界挿入誤り率

表 5.6: 複数時間長・遷移確率付きテンプレート (A) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
遷移確率付き (A)($\beta = 0.00$)	26.556	28.302	26.694	26.892	29.886
(0.25)	26.556	28.184	30.604	35.394	36.408
(0.50)	26.556	27.115	32.664	36.186	38.352
(0.75)	26.556	27.784	34.118	37.458	41.229
(1.00)	26.556	27.243	34.218	40.389	47.063

(単位：%)

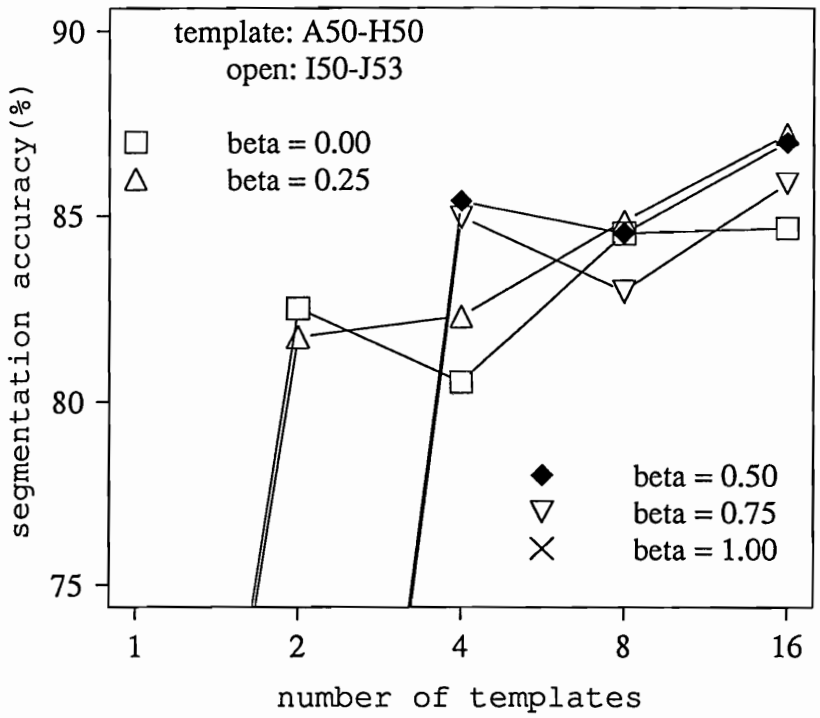


図 5.11: 複数時間長・遷移確率付きテンプレート (B) による句境界検出率

表 5.7: 複数時間長・遷移確率付きテンプレート (B) による句境界検出率

テンプレート数	1	2	4	8	16
遷移確率付き (B)($\beta = 0.00$)	53.540	82.522	80.531	84.513	84.624
(0.25)	53.540	81.748	82.301	84.845	**87.168**
(0.50)	53.540	50.553	85.398	84.513	86.947
(0.75)	53.540	50.000	84.956	82.965	85.841
(1.00)	53.540	49.558	47.788	47.788	47.898

(単位：%)

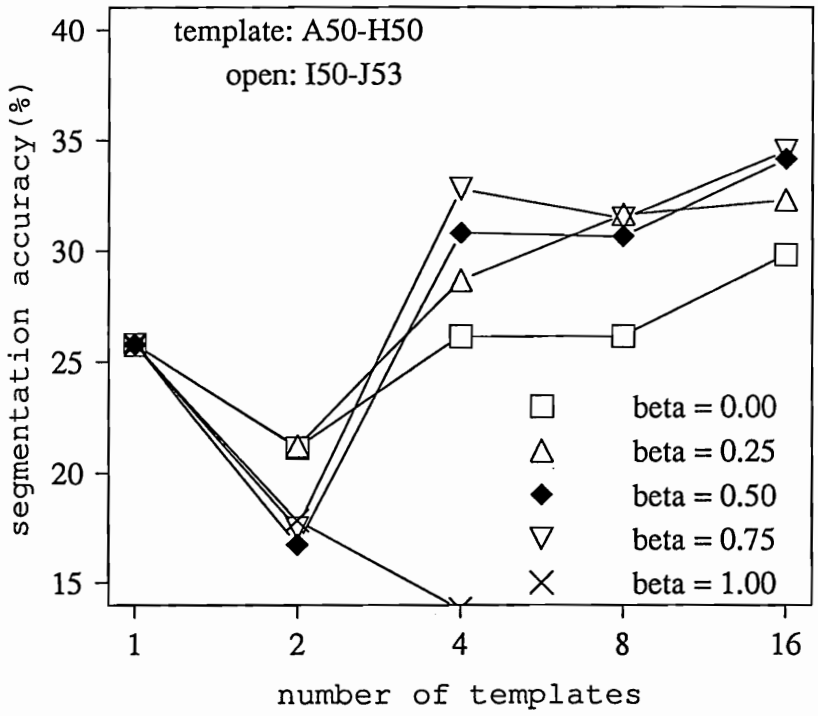


図 5.12: 複数時間長・遷移確率付きテンプレート (B) による句境界挿入誤り率

表 5.8: 複数時間長・遷移確率付きテンプレート (B) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
遷移確率付き (B)($\beta = 0.00$)	25.755	21.135	26.110	26.118	29.761
(0.25)	25.755	21.212	28.643	31.628	32.232
(0.50)	25.755	16.730	30.791	30.624	34.087
(0.75)	25.755	17.490	32.787	31.453	34.446
(1.00)	25.755	17.837	13.814	10.108	9.586

(単位：%)

第 6 章

結論

6.1 はじめに

音声認識の分野において、近年、連続音声を対象とした研究が盛んに行なわれ、特に文構造の推定のためにもアクセント句境界の自動検出は重要な課題として注目されている。本論文では、ピッチパタンの韻律構造に着目した句境界検出法であるピッチパタン連続整合法を基本とし、本論文を通して句境界検出の高精度化を図ってきた。この章は本論文の総括としてこれまでの研究成果についてまとめるものとする。

6.2 句境界検出法の改善とその成果

本論文が推進する「パタン連続整合法による句境界検出」は文献 [16] の One-Stage DP 法を基本アルゴリズムとして 1990 年に下平らによって提案された手法である [8]。まだ基本的なアルゴリズムおよびその概念が定着されて間も無いため、その高精度化に関してはまだまだ改善の余地がある。この論文では第 2 章で従来法の追実験を行ない、その結果考えられる問題点を 3 点に絞って句境界検出の高精度化を図った。

- **ピッチパタンの不連続部分が及ぼす影響 (第 3 章)**

  まず第 1 点目は音声の無声化によってピッチパタンの値が不連続になり、その個所がテンプレートパタンの整合境界として検出され易い点である。

  – これに対し、複数周波数帯域のピッチ候補から DP によってパタンの連続性を重視した候補の選択を行ない、発声全区間に対して連続なピッチパタンを作成した。このパタンは句境界構造を表現するものとして十分有効であり、句境界

検出率の向上にも結びついた。また、これによってピッチパタンによる性能の劣化が取り除かれ、これ以降の句境界検出の性能比較を可能にした。

- **パタンの類似性に関する問題 (第 4 章・前半)**

  絶対的な値によるアクセントパタンを用いてテンプレートを作成した場合、パタンの表現能力の限界があり、テンプレート数 4 から 8 で句境界検出率がそれ以上伸びなくなる。また、絶対的な値がピッチの高さに対する意味を持つため、テンプレートの高さが句境界検出結果に影響を及ぼすことも指摘される。

  – これに対し、相対的なアクセントパタンの学習により、よりアクセント成分に近い表現を可能にする参照テンプレートを作成した。このテンプレートは高さの値に意味を持たないので、句境界の自動検出の際には高さ方向に自由度を与え、パタンの変化に着目した整合を行うことで高精度化を図った。

- **検出数の増加に伴なう挿入誤りに関する問題 (第 4 章・後半)**

  テンプレートの表現能力が高まるにつれて、ピッチパタンの微少な変化までもがアクセントパタンとして検出されるようになった。その結果、句境界検出数が増加し、それに伴なって多くの挿入誤りが生じた。挿入によりアクセントパタンを更に細かなまとまりに分割できるという点で全ての場合が誤りとは言えないが、挿入が生じることによってその前後にある正解句境界の検出が不可能になるケースは避けなければならない。

  – そのため、テンプレートの遷移確率を導入して、整合するテンプレート系列が生成確率的に高くなるように結合を制御することで誤りの抑制を試みた。その結果、検出率は上がったが、数値的にはわずかな変化であった。その理由として確率重み係数が不適であることが挙げられ、今後、よりいっそうの最適化が必要である。

- **固定長テンプレートによる短パタン検出の限界 (第 5 章)**

  3 つ目の問題は単一の長さのテンプレートで検出可能なアクセントパタンの長さに限界がある点である。

  – そこで、複数の時間長を含むテンプレート集合のクラスタリング法を提案し、あらゆる長さのアクセントパタンに対応できるようにした。これによって理論上では、これまでの中で、唯一検出率 100%を達成することができる手法と言える。

以上の結果、従来法で最高 87.058%であった句境界検出率を 89.602%まで高めた。これは句境界の正解領域を前後 1 音素にした場合であり、文献 [8] に倣って±100ms にした場

合、従来 91.372%であったものを 92.478%まで高めたことになる。ただし、従来法の最大検出率がピッチとデルタピッチの混合によって得られたのに対し、本論文の手法は相対的なピッチパタンのみであったので、これにデルタピッチを組み合わせることで、更に検出率を上げることも可能であると考えられる。また、時間の都合上、従来法については複数時間長テンプレートまで拡張実験を行なわなかったので、厳密な性能比較はできない。

6.3　むすび ～連続音声認識への応用～

本論文では句境界検出率の数値のみによる評価を行なってきた。もちろん、連続音声の文構造などを知るには 100%の句境界検出率が必要であり、本研究は意義のあるものである。しかし、句境界の自動検出のプロセスは連続音声認識の中の一部分にすぎず、さまざまなシステムへの応用について、その評価を行なう必要がある。例えば、句境界位置を付加的な情報として連続音声中の音素、単語を単位とした認識も可能になるであろうし、また構文推定や言語構造を利用した文音声の理解への発展も今後の課題である。逆に、認識の結果をフィードバックすることにより、さらに正確な句境界位置を推定することも可能であり、今後は句境界検出と連続音声認識を結びつけた研究へと発展させて行かねばならない。

謝 辞

パタン認識の研究を行うにあたり、全般的な御指導とともにこの研究の機会を与えて下さった東北大学工学部 阿曽弘具 教授、北陸先端科学技術大学院大学 木村正行 教授に心から感謝致します。音声認識の専門分野においては北陸先端科学技術大学院大学 下平博 助教授に終始御指導、御鞭撻いただきましたことをここに深く感謝致します。また、本論文をまとめるに際し、貴重な御意見を戴いた東北大学工学部 丸岡章教授に深く感謝致します。

大学院の音声ゼミの場では東北大学電気通信研究所 曽根敏夫 教授、東北大学応用情報学研究センター 牧野正三 助教授、東北大学電気通信研究所 鈴木陽一 助教授、東北大学工学部 金井浩 助教授に貴重な御意見・御助言を戴き、心から感謝致します。また、同ゼミにおいて曽根研究室、応用情報学研究センターをはじめとする各研究室の皆様に活発な御討論をして戴き、誠にありがとうございました

日々の研究においては、計算機環境を整えていただいた東北大学情報処理教育センター 阿部 亨 博士、東北大学工学部丸岡研究室 (阿曽グループ) の大町真一郎氏、沼田一成氏、後藤英昭氏、福田大氏に感謝の意を表します。また、同研究室の成富敬氏、粟津辰功氏およびそのほかの皆様方にも数々の御討論・御意見を戴いたことに深くお礼申し上げます。

最後に本研究を行う上での貴重な音声資料を提供して下さった ATR 自動翻訳電話研究所に感謝致します。

参考文献

[1] 北原、武田、市川、東倉：“音声言語認知における韻律の役割”
信学論 J70-D No.11 pp.2095-2101 (1987-11)

[2] 北原、東倉：“音声の韻律情報と感情表現”
信学会技報 SP88-15 (1988)

[3] 小松、大平、市川：“韻律情報を利用した構文推定およびワードスポットによる会話音声理解方式”
信学論 J71-D No.7 pp.1218-1228 (1988-07)

[4] 中川、橋本：“HMM 法とベイズ確率を用いた連続音声のセグメンテーション”
信学論 J72-DII No.1 pp.1-10 (1989-01)

[5] 今井、古市：“連続音声の音素単位へのセグメンテーション”
信学論 J72-DII No.1 pp.11-21 (1989-01)

[6] 藤崎、須藤：“日本語単語アクセントの基本周波数パタンとその生成機構のモデル”
日本音響学会誌 Vol.27, pp.445-453 (1971)

[7] 鈴木、関口、重永：“日本語連続音声認識のための韻律情報を利用した句境界の抽出”
信学論 J72-DII No.10 pp.1609-1617 (1989-10)

[8] 下平、木村、嵯峨山：“ピッチパタン連続整合による連続音声のセグメンテーション”
信学会技報 SP90-72 (1990)

[9] H. Shimodaira, M. Kimura: “Accent Phrase Segmentation Using Pitch Pattern Clustering”
ICASSP-92 pp.I-217-220 (1992)

[10] 磯、渡辺、桑原：“音声データベース用文セットの設計”
昭 63 音響春季講演集 2-2-19 (1988-03)

[11] 阿部、匂坂、桑原：“言語・韻律情報を持つ連続音声の基本周波数データベース”
平元音響秋季講演集 2-3-22 (1989-10)

[12] 中津、好田：“会話音声の機械認識における音響処理”
信学論 J61-D No.4 pp.261-268 (1978-04)

[13] 服部：“「文節」とアクセント”
「言語学の方法」 pp.428-446 岩波書店 (1960)

[14] 藤崎、広瀬、高橋、横尾：“連続音声中のアクセント成分の実現”
信学会技報 SP84-36 (1984)

[15] Y. Linde, A. Buzo and R. M. Gray: “An Algorithm for Vector Quantizer Design”
IEEE Trans.Commu.,COM-28,1,pp.85-95 (1980-01)

[16] Hermann Ney: “The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition”
IEEE ASSP-32,2,pp.263-271 (1984-04)

[17] 斎藤、中田：「音声情報処理の基礎」“§7.3 ケプストラム分析”
オーム社 (1981-11)

[18] 嵯峨山、古井：“ラグ窓を用いたピッチ抽出の一方法”
昭 53 信学総全大 1235 (1978-03)

[19] 杉藤、東川、坂倉、高橋：“ささやき声におけるアクセントの知覚的、音響的、生理的特徴”
信学会技報 SP91-1 (1991-05)

[20] 萩原、米田：“時間的な連続性を考慮したピッチ候補の選択法”
信学論 J74-A,No.7,pp.948-956 (1991-07)

[21] 中井、下平：“複数周波数帯域のピッチ候補に基づく連続なピッチパタンの評価”
平 4 音響秋季講演集 1-5-19 (1992-10)

[22] H. Shimodaira, M. Nakai: “Robust Pitch Dection by Narrow Band Spectrum Analysis”
ICSLP-92 pp.1597-1600 (1992)

[23] 中井、下平：“複数周波数帯域のピッチ候補による連続なピッチパタンの作成”
平 4 東北連大 2B14 (1992-08)

[24] R. W. Hamming: "Numerical Method for Scientists and Engineers"
2nd ed., pp.349ff.

[25] W. S. Cleveland: "Robust Locally Weighted Regression and Smoothing Scatterplots"
JASA, Vol.74, No.368, pp.829-836 (1979-12)

研究業績一覧

1. “VQ 符号帳作成の高速化に関する研究”
   中井 満、 下平 博、木村 正行
   平成 3 年度 電気関係学会東北支部連合大会 2A3 pp.23 (1991-08)

2. “VQ 符号帳作成の高速化に関する研究”
   中井 満、 下平 博、木村 正行
   日本音響学会 平成 3 年度秋季研究発表会 1-6-23 pp.235-236 (1991-10)

3. “A Fast VQ Codebook Design Algorithm for A Large Number of Data”
   M. Nakai, H. Shimodaira and M. Kimura
   ICASSP-92 pp.I-109-112 (1992-03)

4. “複数周波数帯域のピッチ候補による連続なピッチパタンの作成”
   中井 満、 下平 博
   平成 4 年度 電気関係学会東北支部連合大会 2B14 pp.74 (1992-08)

5. “複数周波数帯域のピッチ候補に基づく連続なピッチパタンの評価”
   中井 満、 下平 博
   日本音響学会 平成 4 年度秋季研究発表会 1-5-19 pp.253-254 (1992-10)

6. “Robust Pitch Detection by Narrow Band Spectrum Analysis”
   H. Shimodaira and M. Nakai
   ICSLP-92 pp.1597-1600 (1992-10)

付録 A

正解句境界領域 ±100ms による実験結果

本論文中では句境界の正解領域を前後 1 音素とした。これは発声の速度等に対して動的に評価できるようにするためである。しかし、文献 [8] では正解領域として句境界の前後 ±100ms で固定しているので、比較のため同じ基準による実験結果を付録として収録しておく。

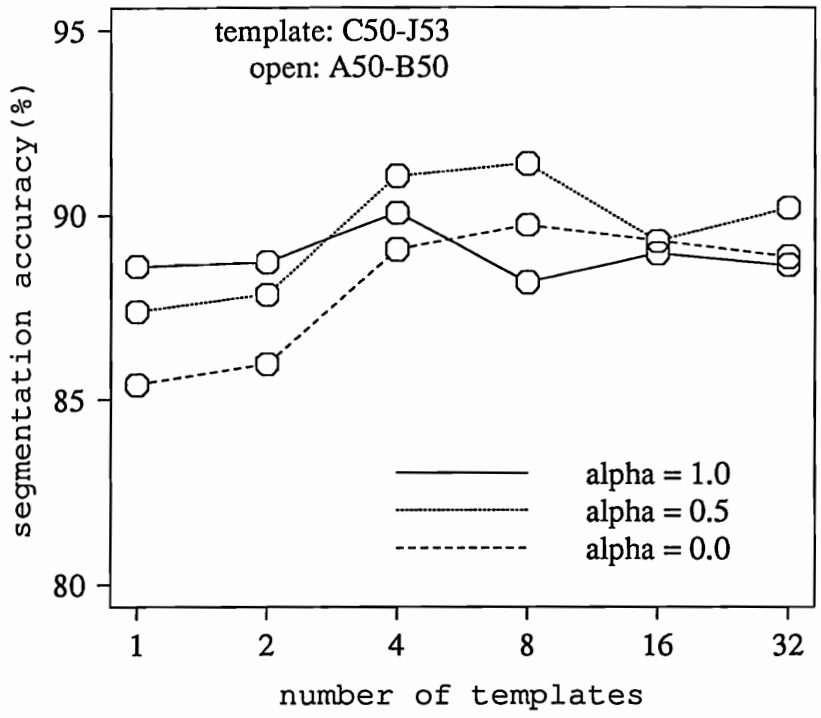


図 A.1: 連続整合セグメンテーションによる句境界検出率

表 A.1: 連続整合セグメンテーションによる句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	85.398	85.951	89.049	89.712	89.270	88.827
0.5	87.389	87.832	91.040	91.372	89.270	**90.155**
1.0	88.606	88.717	90.044	88.164	88.942	88.606

(単位：%)

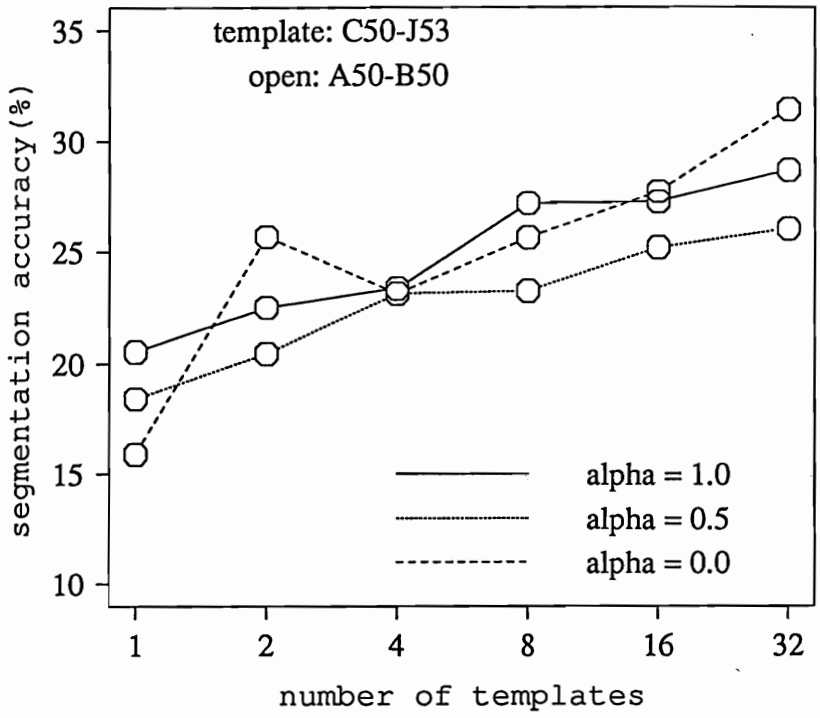


図 A.2: 連続整合セグメンテーションによる句境界挿入誤り率

表 A.2: 連続整合セグメンテーションによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	15.880	25.680	23.116	25.629	27.694	31.374
0.5	18.407	20.438	23.115	23.242	25.171	25.983
1.0	20.534	22.500	23.356	27.177	27.238	28.638

(単位：%)

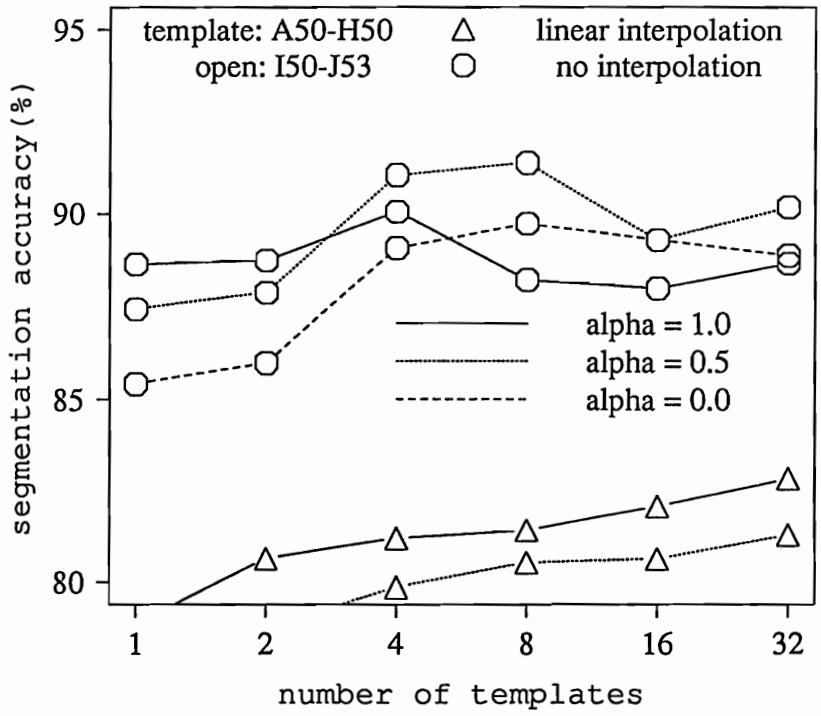


図 A.3: 線型補間パタンの句境界検出率

表 A.3: 線型補間パタンの句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	74.004	74.889	76.438	79.093	78.650	78.872
0.5	76.770	78.761	79.867	80.531	80.642	81.305
1.0	78.872	80.642	81.195	81.416	82.080	**82.854**

(単位：%)

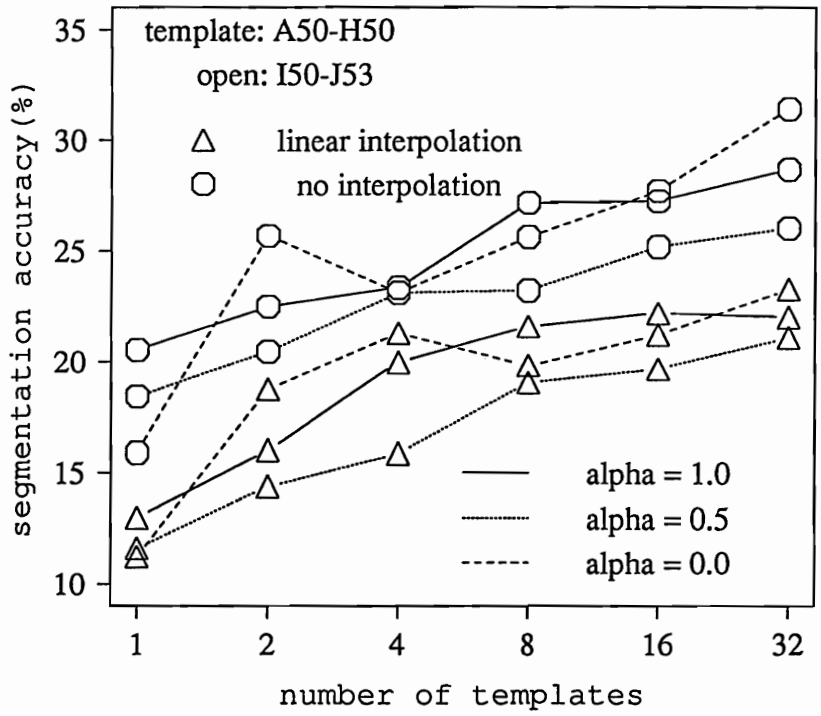


図 A.4: 線型補間パタンの句境界挿入誤り率

表 A.4: 線型補間パタンの句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	11.268	18.726	21.282	19.811	21.212	23.251
0.5	11.622	14.377	15.848	19.036	19.653	21.065
1.0	12.999	16.000	19.954	21.613	22.198	21.991

(単位：%)

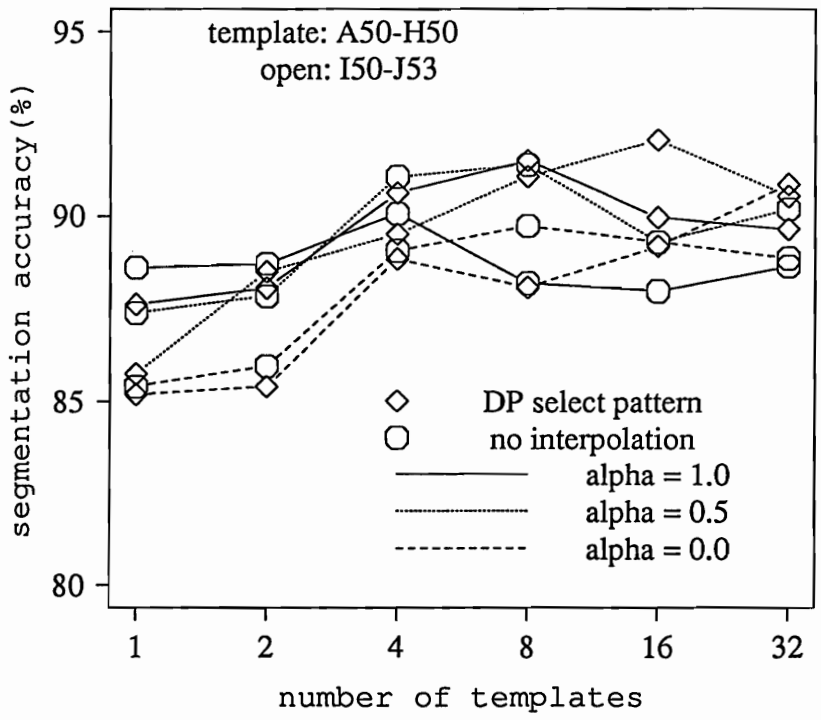


図 A.5: DP 連続パタンの句境界検出率

表 A.5: DP 連続パタンの句境界検出率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	85.177	85.398	88.827	88.053	89.159	90.819
0.5	85.730	88.496	89.491	91.040	**92.035**	90.487
1.0	87.611	88.053	90.597	91.482	89.934	89.602

(単位：%)

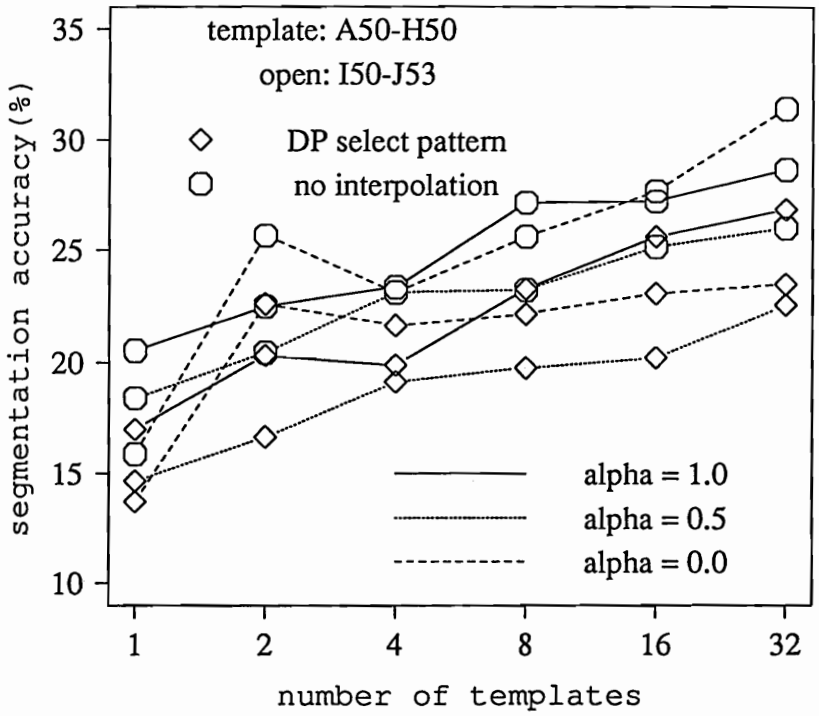


図 A.6: DP 連続パタンの句境界挿入誤り率

表 A.6: DP 連続パタンの句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
$\alpha = 0.0$	13.712	22.608	21.655	22.165	23.077	23.450
0.5	14.662	16.667	19.131	19.757	20.219	22.530
1.0	16.993	20.311	19.879	23.285	25.624	26.818

(単位：%)

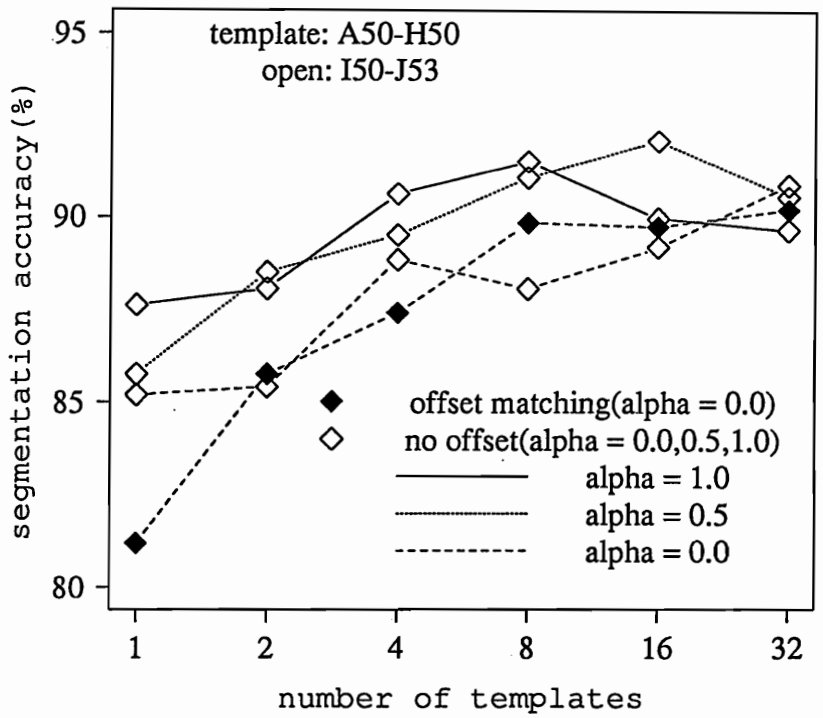


図 A.7: 高さ可変テンプレートによる句境界検出率

表 A.7: 高さ可変テンプレートによる句境界検出率

テンプレート数	1	2	4	8	16	32
高さ固定 ($\alpha = 0.0$)	85.177	85.398	88.827	88.053	89.159	90.819
(0.5)	85.730	88.496	89.491	91.040	92.035	90.487
(1.0)	87.611	88.053	90.597	91.482	89.934	89.602
高さ可変 (A)	81.195	85.730	87.389	89.823	89.712	**90.155**
(B)	54.646	86.283	85.841	88.938	89.049	87.832

(単位：%)

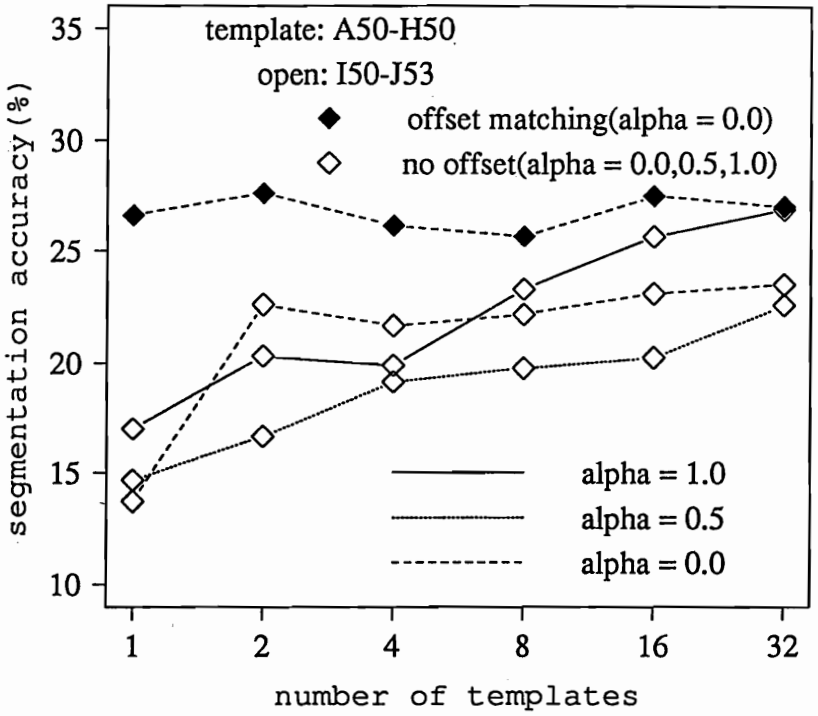


図 A.8: 高さ可変テンプレートによる句境界挿入誤り率

表 A.8: 高さ可変テンプレートによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
高さ固定 ($\alpha = 0.0$)	13.712	22.608	21.655	22.165	23.077	23.450
(0.5)	14.662	16.667	19.131	19.757	20.219	22.530
(1.0)	16.993	20.311	19.879	23.285	25.624	26.818
高さ可変 (A)	26.604	27.613	26.142	25.646	27.477	26.959
(B)	27.035	20.255	25.278	24.287	29.683	30.220

(単位：%)

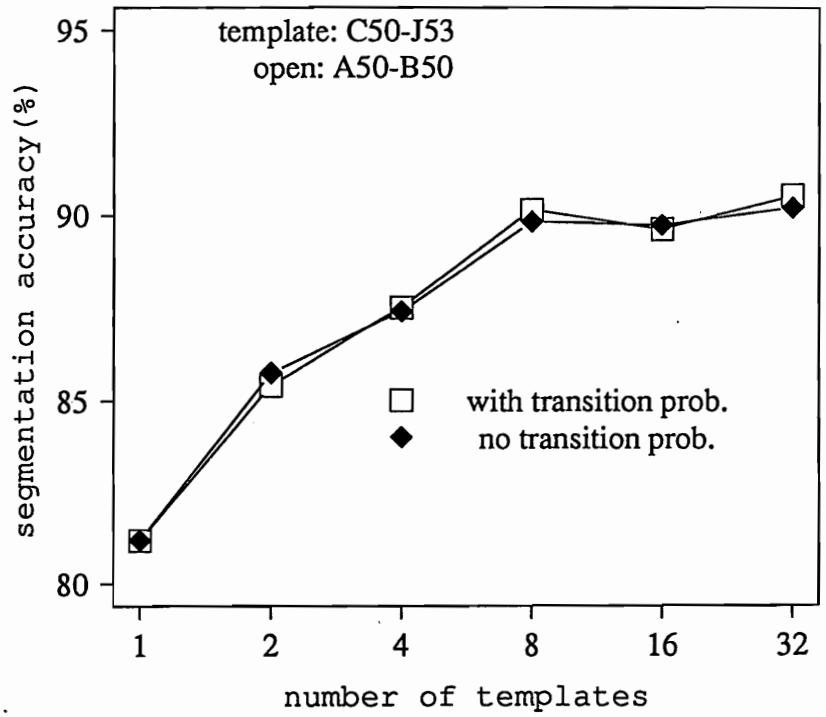


図 A.9: 遷移確率付きテンプレートによる句境界検出率

表 A.9: 遷移確率付きテンプレートによる句境界検出率

テンプレート数	1	2	4	8	16	32
高さ可変 (A)	81.195	85.730	87.389	89.823	89.712	90.155
遷移確率付き (A)	81.195	85.398	87.500	90.155	89.602	**90.487**
高さ可変 (B)	54.646	86.283	85.841	88.938	89.049	87.832
遷移確率付き (B)	54.646	86.062	85.841	88.938	88.606	87.500

(単位：%)

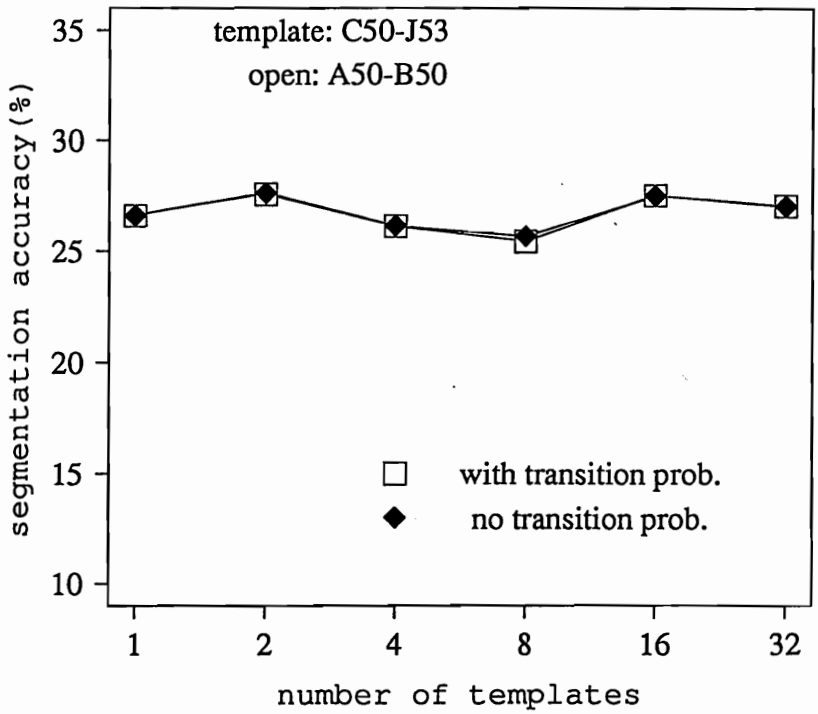


図 A.10: 遷移確率付きテンプレートによる句境界挿入誤り率

表 A.10: 遷移確率付きテンプレートによる句境界挿入誤り率

テンプレート数	1	2	4	8	16	32
高さ可変 (A)	26.604	27.613	26.142	25.646	27.477	26.959
遷移確率付き (A)	26.604	27.569	26.117	25.430	27.502	26.977
高さ可変 (B)	27.035	20.255	25.278	24.287	29.683	30.220
遷移確率付き (B)	27.035	20.106	25.025	24.385	29.764	30.495

(単位：%)

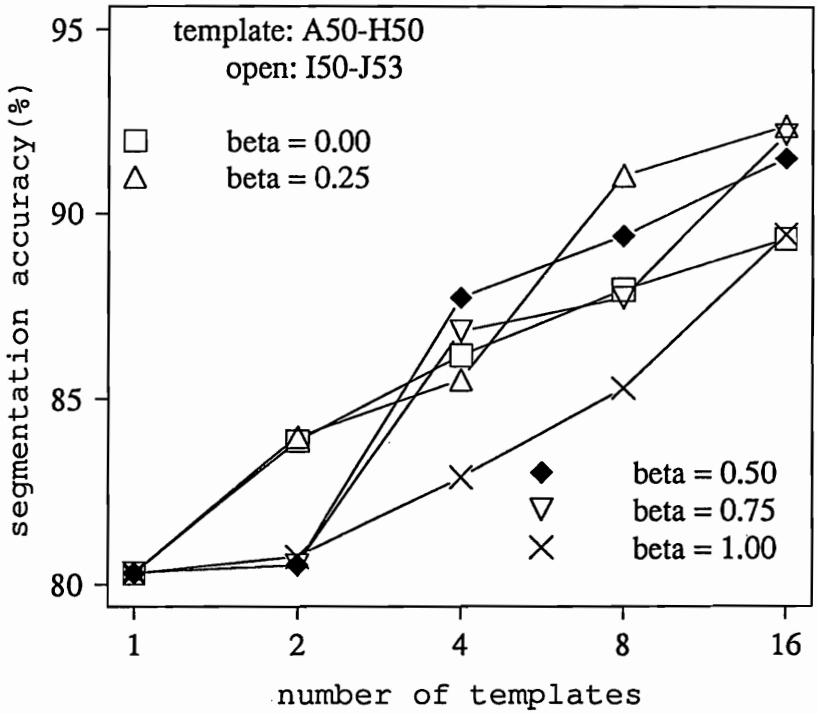


図 A.11: 複数時間長・高さ可変テンプレート (A) による句境界検出率

表 A.11: 複数時間長・高さ可変テンプレート (A) による句境界検出率

テンプレート数	1	2	4	8	16
高さ可変 (A)($\beta = 0.00$)	80.310	83.850	86.173	87.942	89.270
(0.25)	80.310	83.960	85.509	91.040	**92.367**
(0.50)	80.310	80.531	87.721	89.381	91.482
(0.75)	80.310	80.531	86.836	87.721	92.146
(1.00)	80.310	80.752	82.854	85.288	89.381

(単位：%)

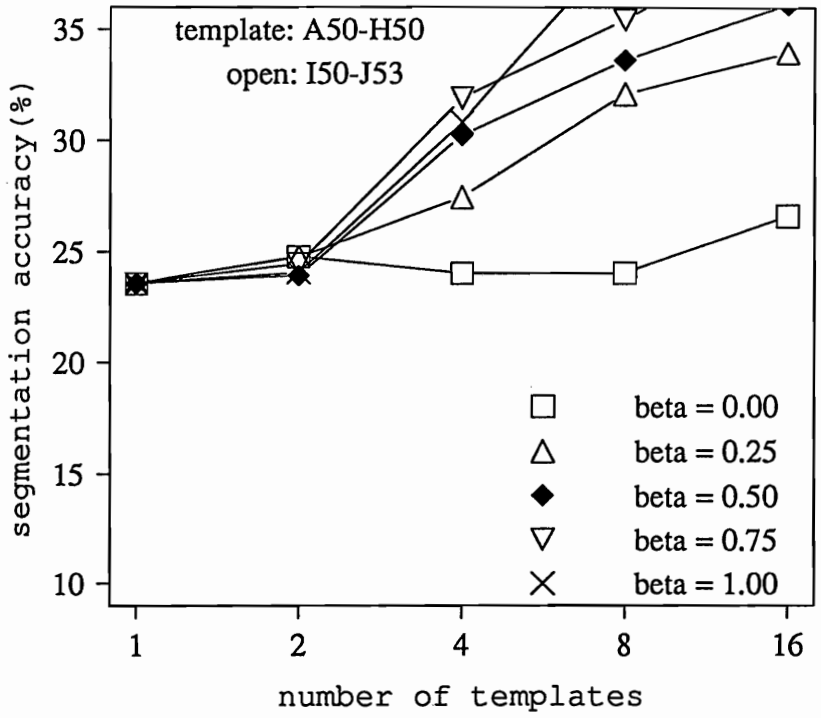


図 A.12: 複数時間長・高さ可変テンプレート (A) による句境界挿入誤り率

表 A.12: 複数時間長・高さ可変テンプレート (A) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
高さ可変 (A)($\beta = 0.00$)	23.556	24.791	24.004	24.008	26.546
(0.25)	23.556	24.765	27.451	32.031	33.880
(0.50)	23.556	23.908	30.201	33.560	36.118
(0.75)	23.556	24.457	31.855	35.427	39.717
(1.00)	23.556	24.048	30.777	39.123	45.429

(単位：%)

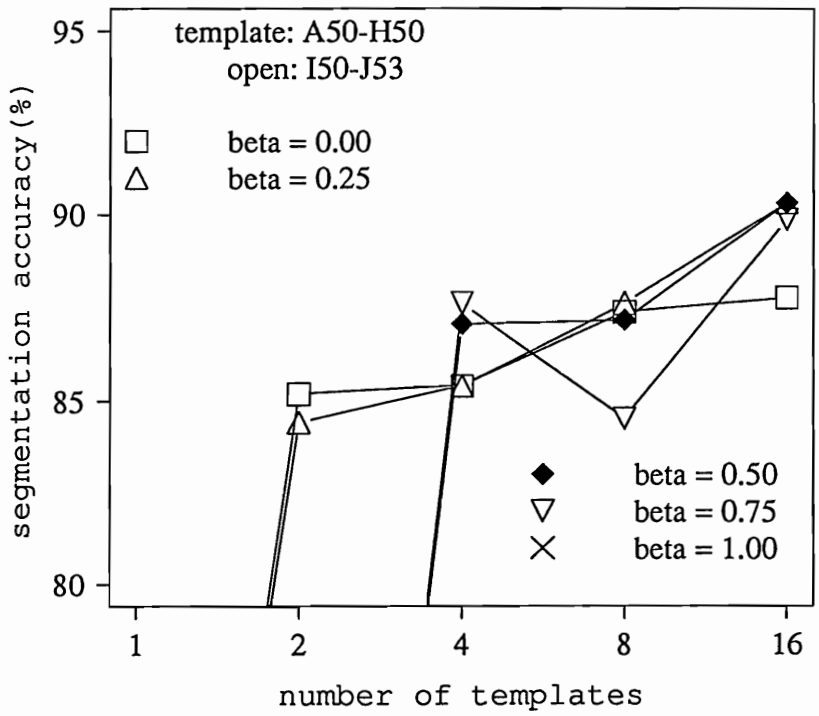


図 A.13: 複数時間長・高さ可変テンプレート (B) による句境界検出率

表 A.13: 複数時間長・高さ可変テンプレート (B) による句境界検出率

テンプレート数	1	2	4	8	16
高さ可変 (B)($\beta = 0.00$)	55.310	85.177	85.398	87.389	87.721
(0.25)	55.310	84.403	85.398	87.611	**90.265**
(0.50)	55.310	49.336	87.058	87.168	**90.265**
(0.75)	55.310	49.447	87.611	84.513	89.823
(1.00)	55.310	50.221	48.009	48.230	47.345

(単位：%)

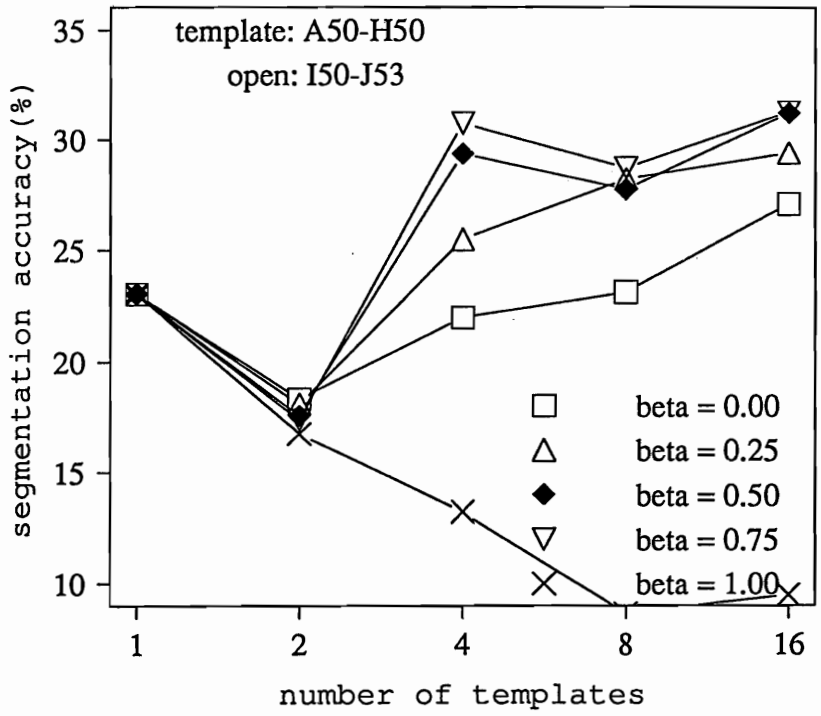


図 A.14: 複数時間長・高さ可変テンプレート (B) による句境界挿入誤り率

表 A.14: 複数時間長・高さ可変テンプレート (B) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
高さ可変 (B)($\beta = 0.00$)	23.052	18.322	21.987	23.116	27.011
(0.25)	23.052	18.080	25.480	28.184	29.349
(0.50)	23.052	17.557	29.346	27.704	31.185
(0.75)	23.052	17.366	30.727	28.696	31.244
(1.00)	23.052	16.730	13.223	8.677	9.471

(単位：%)

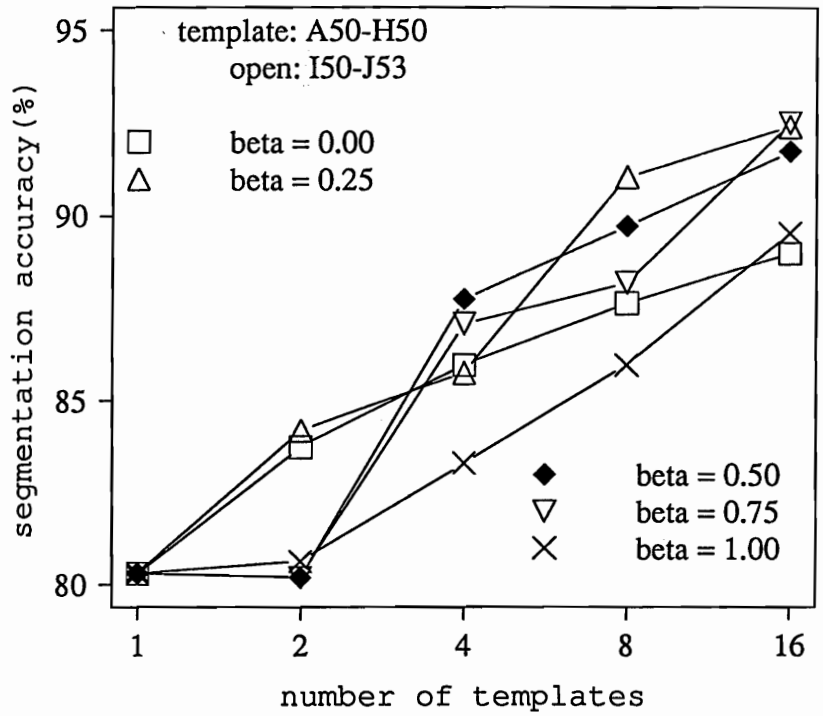


図 A.15: 複数時間長・遷移確率付きテンプレート (A) による句境界検出率

表 A.15: 複数時間長・遷移確率付きテンプレート (A) による句境界検出率

テンプレート数	1	2	4	8	16
遷移確率付き (A)($\beta = 0.00$)	80.310	83.739	85.951	87.611	88.938
(0.25)	80.310	84.181	85.730	91.040	92.367
(0.50)	80.310	80.199	87.721	89.712	91.704
(0.75)	80.310	80.199	87.058	88.164	**92.478**
(1.00)	80.310	80.642	83.296	85.951	89.491

(単位：%)

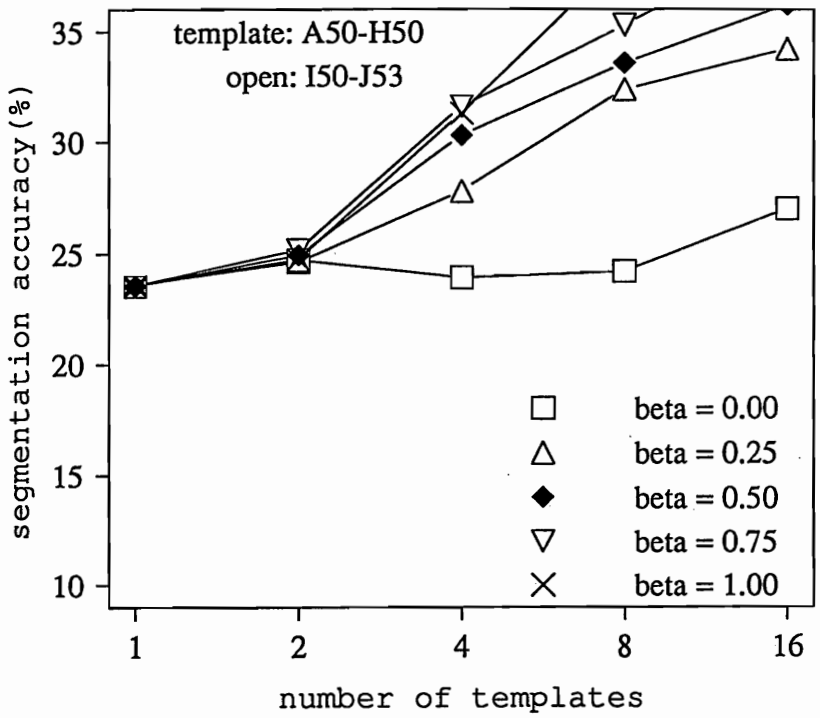


図 A.16: 複数時間長・遷移確率付きテンプレート (A) による句境界挿入誤り率

表 A.16: 複数時間長・遷移確率付きテンプレート (A) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
遷移確率付き (A)($\beta = 0.00$)	23.556	24.738	23.922	24.203	26.945
(0.25)	23.556	24.635	27.778	32.345	34.204
(0.50)	23.556	24.946	30.292	33.559	36.212
(0.75)	23.556	25.189	31.584	35.284	39.130
(1.00)	23.556	24.757	31.280	38.524	45.473

(単位：%)

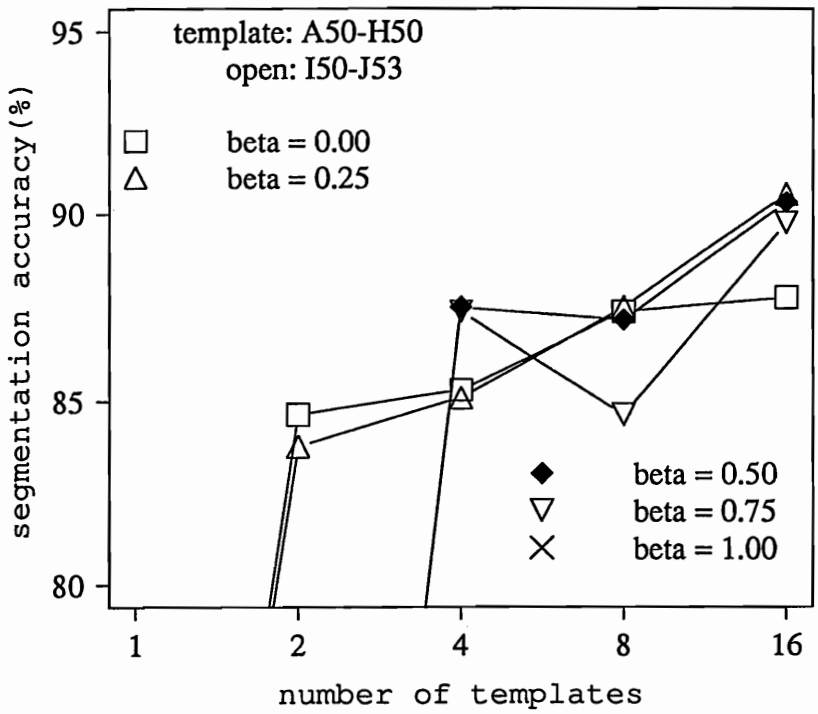


図 A.17: 複数時間長・遷移確率付きテンプレート (B) による句境界検出率

表 A.17: 複数時間長・遷移確率付きテンプレート (B) による句境界検出率

テンプレート数	1	2	4	8	16
遷移確率付き (B)($\beta = 0.00$)	55.310	84.624	85.288	87.389	87.721
(0.25)	55.310	83.739	85.066	87.500	**90.487**
(0.50)	55.310	50.000	87.500	87.168	90.265
(0.75)	55.310	50.000	87.389	84.624	89.712
(1.00)	55.310	50.111	48.451	47.898	48.119

(単位：%)

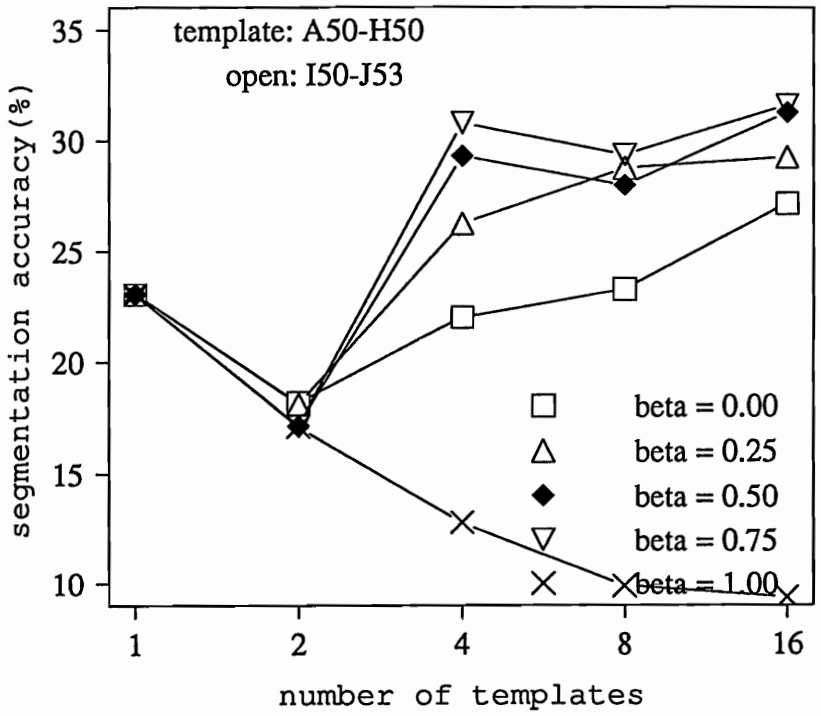


図 A.18: 複数時間長・遷移確率付きテンプレート (B) による句境界挿入誤り率

表 A.18: 複数時間長・遷移確率付きテンプレート (B) による句境界挿入誤り率

テンプレート数	1	2	4	8	16
遷移確率付き (B)($\beta = 0.00$)	23.052	18.131	21.987	23.272	27.081
(0.25)	23.052	18.070	26.231	28.744	29.196
(0.50)	23.052	17.110	29.302	27.977	31.217
(0.75)	23.052	17.110	30.783	29.350	31.547
(1.00)	23.052	17.078	12.784	9.892	9.368

(単位：%)

付録 B

音声資料 (A～B、100 文章)

A01 あらゆる 現実を すべて 自分のほうへ ねじ曲げたのだ。

A02 一週間ばかり ニューヨークを 取材した。

A03 テレビゲームや パソコンで ゲームを して 遊ぶ。

A04 物価の 変動を 考慮して 給付水準を 決める 必要が ある。

A05 救急車が 十分に 動けず 救助作業が 遅れている。

A06 言論の 自由は 一歩 譲れば 百歩も 千歩も 攻めこまれる。

A07 会場の 周辺には 原宿駅や 代々木駅も あるし ちょっと 歩けば 新宿御苑駅も ある。

A08 老人ホームの 場合は 健康器具や ひざ掛けだ。

A09 ちょっと 遅い 昼食を とるため ファミリーレストランに 入ったのです。

A10 嬉しいはずが ゆっくり 寝ても いられない。

A11 自然の 研究者は 自然を ねぜじ伏せようと しては いけない。

A12 おごりを 捨て 謙虚な 姿勢を 取り戻さねば 冬は 過ごせない。

A13 先だって ごく 短期間だが 久方ぶりに ヨーロッパへ 行(い)った。

A14 しかし この プロ野球ブームも 永遠に 続くとは 限らぬ。

A15 お客さんは それじゃ 練習さえ すれば だれに でも できるんじゃ ないか なって 考え 始め るよ。

A16 アフリカ人は 実に 巧みに ぴゅんと つばを 吐く。

A17 前者を 普遍文化と 呼び 後者を 個別文化と 呼ぶ ことに する。

A18 叔父さんは 岬の 一軒家 (いっけんや) に 独りぼっちで 住んでいた。

A19 立春が すぎても 厳しい 寒さの 日々が 続く。

A20 大昔の フィリピンには 豊かの 土地が あった。

A21 旅館や ホテルに 着くと 非常口を 尋ねる。

A22 やるべきことは やっており なんら 落ち度は ない。

A23 私は 上着を 脱ぎ 石組みの 上に 両手を ついて うつぶせに なった。

A24 外人サンは 完璧主義者で ある。

A25 人間とは 微妙で 複雑な 生き物で ある。

A26 ここ 一か月は ほとんど 不眠不休の 徹夜つづきで 目が 腫れ上がって いる。

A27 午前八時 健康な 捕虜は 作業場 (さぎょうじょう) へ トラックで 出発する。

A28 見上げる フジも いいが 露地植え また 鉢植えの 花も きれいです。

A29 母は 脳血栓の 後遺症で 老人痴呆症になり 一年前から 入院中です。

A30 パジャマと ティーシャツが めくれて 薄い 肋骨の 下に ぺちゃんこの 腹が 見えた。

A31 また 襟や 袖口 ポケット口などが 油汚れで 変色を おこすことも あります。

A32 着用中に ダウンや フェザーが 飛び出す 原因とも なります。

A33 インタビューは 午後 十時から 始まり 途中で 夕食を はさみ 延々 四時間に 及んだ。

A34 効果を 急ぐあまりの 過度の 練習は 避け ウオーミングアップも 念入りに やりましょう。

A35 弟子に 腕を 支えられながら 最後まで 引き続けた。

A36 気管支ぜんそくや 鼻炎も 広まっている。

A37 大ピラミッド 近くに 二つの 部屋が 埋まっていた のである。

A38 普通 中距離トラックの ドライバーは 中年の 人が 多い。

A39 自動車や 精密機械などで 技術系の 採用を 抑える ところが 目立ち 売り手市場の 技術系にも かげりが 見え始めた。

A40 ユーザーにも 責任が あるとの 理論は 暴論と 言わざるを えません。

A41 本書は 言葉の 政治人類学と いっても よい。

A42 筋目に 合わせて 本会場を 半分に ちょん切ると するか。

A43 首相 自ら 国民 一人一人 百ドル 舶来品を 買うように すすめた。

A44 十進法は 両手の 十本の 指を 数える ことから 起こった。

A45 ワインと 日本酒 (にほんしゅ) とを 問わず 原産地 成分表示を 急ぐべきでは ないか。

A46 年齢は まだ 十四だが 数えきれぬほど 日本の 舞台を 踏んだので 日本語は ぺらぺらだそうだ。

A47 日本の エスペラントとして やはり 標準語は 必要だ。

A48 翌年 父の 選挙を 手伝って 遊説行脚の マネージャーを 勤めた。

A49 何も かもが たちまち 腐り 指紋で よごれ ぐにゃぐにゃに なってしまう ようだ。

A50 逆境に 耐えた この プロデューサーの 作品には ヒューマニズムが 脈々と 息ずいている。

B01 予防や 健康管理 リハビリテーションの ための 制度を 充実して いく 必要が あろう。

B02 老若男女が 火を 囲んで 飲み 手を つないで 歌う。

B03 出口のない 飛行中の 航空機の 異変は 恐怖の 極限状況と いってよい。

B04 わずかな 収入を やりくりして 現金で サービスを 利用している。

B05 難しい 食事療法から 下痢の 世話まで 二十 四時間介護の 日々が 続いた。

B06 倒れて 道路を ふさぐ 恐れがある ブロック塀や 石塀を 点検し 改善しておく。

B07 ついで 財務省が 専門家を 集めて 具体案を 練った。

B08 こういう 絵が やすやすと 描(か)ける はずがない。

B09 背の 高さは 一七〇センチほどで 目が 大きく やや 太っている。

B10 なつかしい プロペラ機で ふわり ふわりと 地球を 一周した ところが すばらしい。

B11 よほど 具合が 悪くなければ 昼間に 横に なるなんて 夢の また 夢だ。

B12 強風と 冷えこみの 強い 神宮の グラウンドである。

B13 ぶらぶらと 球場まで 十分足らずの 道を 歩いて いく。

B14 大声を 出しすぎて かすれ声に なってしまう。

B15 足し算 引き算は できなくとも 絵は 描(か)ける。

B16 それを かばおうとして 右ひざと ふくらはぎを やられた。

B17 現地に 着いて いざというときに おぶってくれる 原住民を 頼んだが これが ピグミーで あった。

B18 ゆらゆら 電球が 揺れて 影が 草の 上を ちらちらした。

B19 ぼくは ほとんど 夢中で 駅前の 人ごみの 間を すりぬけた。

B20 この 喜びは むろん 家(いえ)へ 帰り着いても 消えずに つづいた。

B21 どの 部屋の 意匠にも 遊び心が あふれていて 楽しい。

B22 飛ぶ 自由を 得ることは 人類の 夢だった。

B23 育ちの よい 坊っちゃんの 良さと 逆境に 育った 人間の 強さ。

B24 初めて ルーブル美術館へ 入ったのは 十四年 前の ことだ。

B25 自分の 実力は 自分が 一番 よく 知っている はずだ。

B26 茶色の 眼は 柔和な かがやきを おびていた。

B27 今 流行の 単身赴任族の 淋しさを ちょぴり 味わわせてもらったのも 有意義な 体験だ。

B28 やはり 無表情のまま 何も ことばが ありません。

B29 生きた 潤滑油です。

B30 彼の 数学の 授業は 抜群に 面白く 試験前には 月給外補習授業を する程 熱心である。

B31 そうでなくとも 寿司屋の 職人は 減らず口を ききたがる 人間が 多い。

B32 リードが 大きければ 牽制球を 投げなければ ならない。

B33 アメリカが 風邪をひけば 日本も クシャミをすると いわれる程で 日本経済も 不況です。

B34 ラッパも 鳴らないし 笛も 鳴らないが ぞろぞろと 起き出し 洗面所へ ゆく。

B35 主人に 甘え 社会に 甘え 自分に 甘えて ぬるま湯に どっぷり 浸って いる。

B36 船は ひそかに 揺れつつ 錨を ひきずって 流れつづけた。

B37 これまで 少年野球 ママさんバレー など 地域スポーツを 支え 市民に 密着してきたのは 無数の ボランティア だった。

B38 もちろん 調査後は 元通り 密閉する。

B39 冷房では 冷え過ぎが 問題に なる。

B40 いずれは ハワイや カリフォルニアなど 日本人が 多く 住む 暑い 土地で 育ててみたい。

B41 最近の 不調を 理由に ソウル五輪候補選手から 外す ことを 発表した。

B42 事故の 直接原因となった 圧力隔壁の ずさんな 修理 その ずさんさを 見落とした チェックシステムなどが そうだ。

B43 女性とは 逆で 何とか 常識を 破って めだってやろうと 意気込む 人が ほとんどだ。

B44 もっと 広い 議場をという 声も あったが チャーチル首相が 抑えた。

B45 ギンザケの 卵を 輸入して ふ化させ 海中で 育てる 養殖も 始まっている。

B46 文書は 年々 増えていく。

B47 おしゃれとは 縁が なく ジーパンに ティーシャツ ジャンパーといった 格好で 駅まで 自転車を 走らせる。

B48 熱で うるんだ 青い 空に 積乱雲が ある。

B49 富者は 貧者と 同じ 栄養状態に 落ち込み 貧者は 餓死まで あと 一歩という 状態へ 落ち込んでいく。

B50 販売関係の 企業の 代表者は セミナー 終了後 会議室に 集れ。