

# 音声認識におけるDPマッチングのための ハードウェア・アルゴリズムに関する検討

A VLSI-ALGORITHM FOR DYNAMIC TIME WARPING  
IN CONNECTED SPOKEN WORD RECOGNITION

前場 隆史

辰巳 昭治

Takashi MAEBA

Shoji TATSUMI

豊橋技術科学大学

情報工学系

Dept. of Information & Computer Sciences, Toyohashi University of Technology

あらまし パターンマッチングによる音声認識において有用な手法であるDPマッチングのためのハードウェア・アルゴリズムを提案する。

標準パターンおよび入力パターンの最大長をそれぞれ  $I$ ,  $J$ , 単語数を  $N$  とするとき,  $p \times q$  ( $p < I$ ,  $q < J$ ) の処理要素 (PE) で構成された六角状配列を用いて,  $I+J+N+\lfloor \frac{N-1}{p} \rfloor(I-p) + \lfloor \frac{J-1}{q} \rfloor(\frac{IN}{p}-q)-2$  ステップの計算時間で正しくDPマッチング処理を行えることを示す。さらに, 本アルゴリズムは音声認識の実時間性を考慮し, 入力パターン長に依存しないため, 離散単語だけでなく連続単語音声認識へも適用可能である。

**Abstract** In this paper, we propose a VLSI-algorithm for dynamic time warping, which is a very useful technique for spoken word recognition based on a pattern matching method. We show the proposed VLSI-algorithm computes accumulated distances between an unknown input pattern and reference patterns correctly in time  $I+J+N+\lfloor \frac{N-1}{p} \rfloor(I-p) + \lfloor \frac{J-1}{q} \rfloor(\frac{IN}{p}-q)-2$  steps with  $p \times q$  PE's-hexagonal array, where  $I$  and  $J$  are the maximum lengths of the reference pattern and the unknown pattern respectively, and  $N$  is the number of words (reference patterns). This algorithm is also independent of the length of the unknown pattern, so that it can be applied to connected word recognition without difficulty.

## 1. まえがき

音声タイプライタ, 機械翻訳等音声入力による人間と機械との対話を推し進める研究が, 現在盛んに行なわれている。音声認識の一つの有力な手法として, 標準パターンと入力パターン間のマッチング処理を動的計画法に基づいて行なう手法(DPマッチング)がある[7]。しかし, DPマッチング処理は膨大な計算量が必要なため, 高速に実行することが望まれる。

一方, VLSI技術の発展の影響で従来, 計算機の上でソフトウェアとして実行させてきたアルゴリズムを直接, ハードウェアとしてVLSIチップ上に実現することを前提としたハードウェア・アルゴリズムの研究が行われ, VLSIアーキテクチャを効率的に動作させる手法の開発, 既存のアルゴリズムの高速化手法が検討されている[8]。

これまでに, いくつかの動的計画法を利用したハードウェア・アルゴリズムが提案されている。菊野等[4]は, 2本の系列から成る集合の最大部分系列を求めるハードウェア・アルゴリズムを提案し, 程等[3]は配分問題を動的計画法に基づいて解くハードウェア・アルゴリズムを報告している。また, Burr等[1]は音声認識のためのDPマッチングを実行するVLSI設計を示している。さらに, スtringマッチングおよびDPマッチングを行なうVLSIアーキテクチャがCheng等[2]によって議論されている。しかし, [1]の手法はDPマッチング処理特有のパターン境界部分の処理方法が明確に示されておらず, [2]のアルゴリズムでは, 入力パターン長が既知でなければ処理を開始できないので, 本質的に実時間処理は行なえない。ここでの実時間処理とは, 音声入力と同時に処理が開始され, 入力に同期して処理が進むこ



六角状アレイを構成している。各PEは計算を行うのに必要な数だけの計算用レジスタ、六つの隣接PE間通信用入出力ポートおよび入出力データ保持用レジスタを持つ。そして、算術論理ユニットでデータの加工を行うものとする。ただし、算術論理ユニットでの計算は一定時間で終了する。また、アルゴリズムに従って制御を行う制御装置を持つ。PEの同期は隣接PE間と通信を行うときのみ、一斉に全PEで同期をとるものとする。

### 3.2 PEの構造と動作

PEの内部構造を図3に示す。各PEは自身を表わす固有の識別番号( $i, j$ )を持ち、レジスタIDに格納している。また、 $H_{in}, H_{out}, V_{in}, V_{out}, S_{in}, S_{out}$ と名前付けられた6つのポートを持つ。ただし、 $H_{out}, V_{out}$ は入力パターン $a$ 、標準パターン $b$ をラッチする必要があるため、特徴ベクトルの次元数に応じたサイズ( $l+1$ )のFIFOバッファで構成される。

各PEは基本的に以下の動作を行う。

#### アルゴリズムPEA:

1°: ポート $V_{in}, H_{in}$ から $a_{j,r}, b_{i,r}$  ( $1 \leq r \leq l$ )を $R_a, R_b$ でそれぞれ受け、(3)式のスレーム間距離を計算し、その結果を $R_d$ に格納する。また、ポート $V_{out}, H_{out}$ へ $a_{j,r}, b_{i,r}$  ( $1 \leq r \leq l$ )を出力する。  
 2°: ポート $V_{in}, H_{in}, S_{in}$ から、 $D_{i-1,j}, D_{i,j-1}$

Algorithm PEA:

```

begin
  Rd:=0;
  for r:=1 to l do begin
    receive (Ra, Rb) from (Vin, Hin); Rd:=Rd+abs(Ra-Rb);
    send (Ra, Rb) to (Vout, Hout);
  end;
  case ID of
    (1, 1):begin
      receive (dummy, dummy, dummy) from (Vin, Hin, Sin);
      RDv:=∞; RDh:=∞; RDs:=0;
    end;
    (1<i≤I, 1):begin
      receive (RDv, dummy, dummy) from (Vin, Hin, Sin);
      RDh:=∞; RDs:=∞;
    end;
    (1, 1<j≤J):begin
      receive (dummy, RDh, dummy) from (Vin, Hin, Sin);
      RDv:=∞; RDs:=∞;
    end;
    (1<i≤I, 1<j≤J):
      receive (RDv, RDh, RDs) from (Vin, Hin, Sin);
  end;
  RD:=min(RDv+Rd, RDh+Rd, RDs+2*Rd);
  send (RD, RD, Dreg) to (Vout, Hout, Sout);
  Dreg:=RD;
end.
  
```

図4. アルゴリズムPEAの詳細

$D_{i-1,j-1}$ を入力し、レジスタ $RDv, RDh, RDs$ へ格納する。累積距離を(2)式に従って計算する( $RD = \min(RDv+Rd, RDh+Rd, RDs+2*Rd)$ )。そして、 $V_{out}, H_{out}$ へ $RD$ の内容( $D_{i,j}$ )を出力し、 $S_{out}$ へ $Dreg$ の内容(1単位時間前に計算した $D_{i,j}$ )を出力する。  
 3°:  $Dreg$ に $RD$ の内容を代入する(対角線方向のデータ出力に関する遅延操作)。

ただし、境界部分を担当するPEでは、2°の動作が境界状態の差異により若干異なる。PEが実行するアルゴリズムPEAの詳細は3.3で示す。

1°から3°の動作をそれぞれ、サイクル $i$  ( $1 \leq i \leq 3$ )の動作と呼び、サイクル1°からサイクル3°までを合わせてステップ $i$ と呼ぶ。すなわち、1ステップは3サイクルから構成され、1ステップを時刻の単位として採用する。

### 3.3 アルゴリズムPEA

アルゴリズムの詳細を図4に示す。アルゴリズムは、Pascal風言語に基づいて記述する。ただし、PE間通信を記述するための命令を以下に定義する。

[定義1] 入力命令 receive  $v$  from  $p$

入力ポート $p$ からのデータをレジスタ $v$ へ格納する。この命令の実行は、すべてのPEで同時に行うため、最も遅いPEに合わせて通信のための同期がとられる。■

[定義2] 出力命令 send  $v$  to  $p$

レジスタ $v$ に格納されているデータを出力ポート $p$ へ出力する。ただし、この命令が実行されても、直ちに隣接PEへデータが送られるわけではなく、出力ポートのバッファへデータがラッチされるのみである。実際に、データが通信路を通して転送されるのは、隣接PEが上で定義した入力命令を実行したときである。■

以上の命令において、ポート $p$ およびレジスタ $v$ の欄にはそれぞれの並びを書くことが許される。この場合、ポート並びとレジスタ並びは記述順に対応付けられる。

$I = 3, J = 3$ のときの動作例を図5に示す。

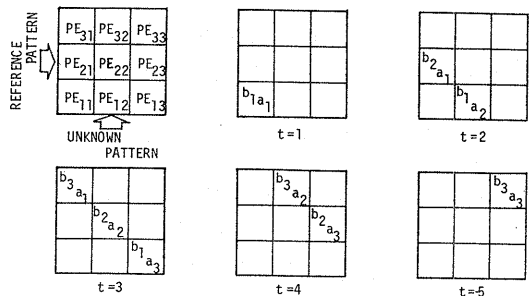


図5. HADPの動作例

### 3.4 アルゴリズムHADPの正当性

HADPの正当性を示すための準備として、いくつかの定義を与える。

【定義3】  $ID=(i,j)$ であるPEをPE(i,j)で表わし、PE(i,j)の任意のレジスタ $r$ を $r(i,j)$ で表わす。■

【定義4】 入力パターン $a_{1,r}$ 、標準パターン $b_{1,r}$  ( $1 \leq r \leq l$ )が共にPE(1,1)に入力されるステップを時刻の始点とし、 $t=1$ と定義する。また、便宜上 $a_{j,r}$ 、 $b_{i,r}$  ( $1 \leq r \leq l$ )をそれぞれ $a_j$ 、 $b_i$ と表記する。■

【入力仕様1】  $a_j$ は時刻(ステップ) $t=j$ においてPE(1,j)に入力される。同様に、 $b_i$ は時刻 $t=i$ にPE(i,1)に入力される。■

上述の定義および入力仕様から、次の補題が導かれる。

【補題1】  $a_j$ 、 $b_i$ は時刻 $t=k$ で $k=i+j-1$ を満たす全てのPE(i,j)に入力される。ただし、 $1 \leq i \leq I$ 、 $1 \leq j \leq J$ である。

(証明) ある時刻 $t$ における $a_j$ に関する動作に着目する。アルゴリズムPEAより、 $a_j$ はサイクル $1^\circ$ でポート $V_{in}$ から入力され、ポート $V_{out}$ へ出力される。入力仕様1より、 $a_j$ は時刻 $j$ にPE(1,j)に入力されるので、 $t=k \geq j$ のとき、 $k-j$ 個のPEを通してアレイ上を上方方向にシフトされる( $V_{in} \rightarrow V_{out}$ )。従って、 $t=k$ において $a_j$ はPE(1+k-j,j)に入力される。一方、 $t=k < j$ では、 $a_j$ はどのPEへも入力されず、PEアレイの外部(外界)に存在することになる。また、 $b_i$ についても同様に、 $t=k$ でPE(i,k-i+1)に入力される。よって、 $k=i+j-1$ より補題が成り立つ。■

補題1およびアルゴリズムPEAから次の補題を得る。

【補題2】 時刻 $t=k$ において、 $k=i+j-1$ を満たす全てのPE(i,j)で

$Rd(i,j) = d_{i,j}$   
が成り立つ。

(証明) 補題1より、 $t=k$ で、 $k=i+j-1$ を満たすPE(i,j)に $a_j$ 、 $b_i$ が入力され、PEAのサイクル $1^\circ$ において、 $Rd := \sum_{r=1}^l \text{abs}(a_{j,r} - b_{i,r})$ が実行される。これは、ADPの(3)式を計算していることに他ならない。よって、 $Rd(i,j) = d_{i,j}$ が成り立つ。■

【補題3】 時刻 $t=k$ において、 $k=i+j-1$ を満たす全てのPE(i,j)で

$RD(i,j) = D_{i,j}$   
が成り立つ。

(証明)  $t$ に関する数学的帰納法により証明する。 $t=1$ のとき、補題の条件を満たすのはPE(1,1)のみで、アルゴリズムPEAより、サイクル $2^\circ$ において $RDv := \infty$ 、 $RDh := \infty$ 、 $RDS := 0$ となる。また、補題2より $Rd = d_{1,1}$ が成り立つので、 $RD := \min(RDv + Rd,$

$RDh + Rd, RDS + 2 * Rd) = 2d_{1,1}$ となる。結局、これはアルゴリズムADPにおいて $i=1, j=1$ のときの(2)式を正しく計算していることに他ならない。よって、 $t=1$ のとき $RD(1,1) = D_{1,1}$ が成り立つ。

次に、 $t=k+1$ のときにも成り立つことを示す。境界条件の違いから、次の三つの場合が存在する。

1.  $1 < i \leq I, 1 < j \leq J$ のとき(ただし、 $i+j-1=k$ )。

補題の条件を満たす $i, j$ の対を $(u, v) \in \{(i, j) \mid i+j-1=k+1, 1 < i \leq I, 1 < j \leq J\}$ とする。PEAより、 $RDv$ 、 $RDh$ にはそれぞれ、時刻 $t=k$ のときのPE(u-1,v)、PE(u,v-1)のRDの内容が入力される。帰納法の仮定より、 $RDv(u,v) := RD(u-1,v) = D_{u-1,v}$ 、 $RDh(u,v) := RD(u,v-1) = D_{u,v-1}$ が成り立つ。また、PEAのサイクル $3^\circ$ の動作およびサイクル $2^\circ$ のポート $S_{out}$ に対する出力操作から、 $RDS$ には時刻 $t-1$ のときのPE(u-1,v-1)のRDの内容が入力される。帰納法の仮定により

$$RDS(u,v) = D_{reg}(u-1,v-1) \\ = RD(u-1,v-1) = D_{u-1,v-1}$$

である。また、補題2より、 $Rd(u,v) = d_{u,v}$ が成り立つので、アルゴリズムPEAより

$$RD(u,v) := \min(RDv(u,v) + Rd(u,v), RDh(u,v) \\ + Rd(u,v), RDS(u,v) + 2 * Rd(u,v)) \\ = \min(D_{u-1,v} + d_{u,v}, D_{u,v-1} + d_{u,v}, \\ D_{u-1,v-1} + 2d_{u,v})$$

である。これは、ADPにおける $i=u, j=v$ のときの(2)式を正しく計算している。従って $RD(u,v) = D_{u,v}$ が成り立つ。

2.  $i=1, j=k+1$ のとき

1. と同様の議論から、 $RD(1,k+1) = D_{1,k+1}$ 。

3.  $j=1, i=k+1$ のとき

同様に、 $RD(k+1,1) = D_{k+1,1}$ 。

以上のことから $t=k+1$ のときも成り立つ。■

次に、いくつかの標準パターンに対して、パイプライン的に連続してDPマッチングを行えることを示す。入力仕様を次のように定義する。

【入力仕様2】  $a_j$ は時刻 $t=j+n-1$ において、PE(1,j)に入力される。一方、 $b_i$ は時刻 $t=i+n-1$ にPE(i,1)に入力される。■

上述の入力仕様と補題3から、次の補題が成り立つ。

【補題4】 時刻 $t=k$ のとき、 $k=i+j+n-2$ を満たす全てのPE(i,j)で

$RD(i,j) = D_{i,j}^n$   
が成り立つ。

(証明) アルゴリズムPEAより明らかに、 $RD(i,j)$ の内容は現時刻にポート $V_{in}$ 、 $H_{in}$ 、 $S_{in}$ から入

力したデータのみ依存する。また、入力仕様2より、標準パターン  $n$  の特徴ベクトル  $b_i^n$  は、標準パターン1の特徴ベクトル  $b_i^1$  を入力した後、 $(n-1)$ ステップ経過してからPE  $(i,1)$ へ入力される。今、時刻の始点を、 $b_i^1, a_1$ がPE  $(1,1)$ へ入力されたときに変更すると、 $t' = k - (n-1)$ で、単語  $n$ の標準パターンに対しても補題1から補題3が成立する。よって  $t = k$ で、 $k = i + j + n - 2$ を満たす全てのPE  $(i,j)$ で  $RD(i,j) = D_{i,j}^n$ が成り立つ。■

補題4から次の定理が導かれる。

【定理1】 ハードウェア・アルゴリズムHADPは、 $t = I + J + n - 2$ の計算時間で、累積照合距離  $D(A, B_n)$ を正しく計算する。

(証明)  $D(A, B_n) = D_{I,J}^n / L$ であるから、 $D_{I,J}^n$ が正しく計算されることを示せば良い。補題4よりPE  $(I,J)$ において、 $RD(I,J) = D_{I,J}^n$ が成り立つ。また、このとき、 $t = I + J + n - 2$ は明らかである。■

#### 4. 連続単語音声認識向きハードウェア・アルゴリズム

##### 4.1 ハードウェア・アルゴリズムHADPSの概要

任意長のパターンを処理するアルゴリズムを実現するハードウェア・アルゴリズムを図6に示す。PEで構成されるアレイのサイズは  $p \times q$ で、3.で述べたHADPとは異なり一定である。固定サイズのPEアレイで任意の大きさのパターンマトリクス空間を標準パターン方向に走査しながら処理する[5]。このため、累積距離の中間結果をPEアレイ外部に設けたシフトレジスタへ格納し、適当な時刻に再びPEアレイへ入力する。標準パターン長を  $I$ 、登録単語数(標準パターン数)を  $N$ とすると各シフトレジスタのサイズは  $I \times (p - q)$ で与えられる。ここで注意しなければならないのはシフトレジスタのサイズは入力パターン長  $J$ に依存しないことであり、後述する連続単語音声認識における重要な要因となる。図6は、 $p = q = 3, I = J = N = 6$ の場合である。

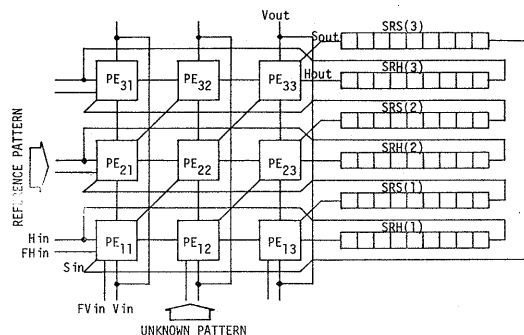


図6. ハードウェア・アルゴリズムHADPSの概要

##### 4.2 PEの構造と動作

PEの構造は基本的に3.2で述べた構造と同一である。ただし、PE  $(k, 1)$  ( $1 \leq k \leq p$ )は入力ポート  $FH_{in}$ とレジスタEF、PE  $(1, n)$  ( $1 \leq n \leq q$ )には入力ポート  $FV_{in}$ とレジスタDFがそれぞれ増設されている。これらのポートあるいはレジスタは、PEが処理しているパターンマトリクス空間の位置を表す制御情報を外界から入力するために用いられるので、境界条件を満たすように処理できる。

各PEの動作は3.2で述べた動作とサイクル2°を除いて同一である。サイクル2°の動作は次のようになる。

##### アルゴリズムPEAS:

2°: PE  $(1,1)$ :  $FV_{in}, FH_{in}$ から制御情報を入力し、EF、DFへ格納する。DF=true、EF=trueならパターンマトリクス空間の  $(1,1)$ -格子点の処理をする。DF=trueでEF=falseなら  $(1,j)$ -格子点 ( $q < j \leq J$ )の処理をする。DF=falseでEF=trueなら  $(i,1)$ -格子点 ( $p < i \leq I$ )の処理を担当する。DF=falseでEF=falseなら  $(i,j)$ -格子点 ( $p < i \leq I, q < j \leq J$ )の処理をする。

PE  $(k,1)$ :  $FH_{in}$ から制御情報を入力し、EFへ格納する。EF=trueなら、 $(i,1)$ -格子点 ( $1 < i \leq I$ )の処

Algorithm PEAS:

```

begin
  Rd:=0;
  for r:=1 to l do begin
    receive (Ra, Rb) from (Vin, Hin); Rd:=Rd+abs(Ra-Rb);
    send (Ra, Rb) to (Vout, Hout);
  end;
  case ID of
    (1, 1):begin
      receive (DF, EF) from (FVin, FHin);
      if DF and EF then begin
        receive (dummy, dummy, dummy) from (Vin, Hin, Sin);
        RDv:=∞; RDh:=∞; RDs:=0;
      end else
      if DF and not(EF) then begin
        receive (dummy, RDh, dummy) from (Vin, Hin, Sin);
        RDv:=∞; RDs:=∞;
      end else
      if not(DF) and EF then begin
        receive (RDv, dummy, dummy) from (Vin, Hin, Sin);
        RDh:=∞; RDs:=∞;
      end else receive (RDv, RDh, RDs) from (Vin, Hin, Sin);
    end;
    (1, j>1):begin
      receive DF from FVin;
      if DF then begin
        receive (dummy, RDh, dummy) from (Vin, Hin, Sin);
        RDv:=∞; RDs:=∞;
      end else receive (RDv, RDh, RDs) from (Vin, Hin, Sin);
    end;
    (i>1, 1):begin
      receive EF from FHin;
      if EF then begin
        receive (RDv, dummy, dummy) from (Vin, Hin, Sin);
        RDh:=∞; RDs:=∞;
      end else receive (RDv, RDh, RDs) from (Vin, Hin, Sin);
    end;
    (i>1, j>1): receive (RDv, RDh, RDs) from (Vin, Hin, Sin);
  end;
  RD:=min(RDv+Rd, RDh+Rd, RDs+2*Rd);
  send (RD, RD, Dreg) to (Vout, Hout, Sout);
  Dreg:=RD;
end.

```

図7. アルゴリズムPEASの詳細

理をする。EF=falseなら、(i,j)-格子点( $1 \leq i \leq I, q < j \leq J$ )の処理をする。

$PE(1, n)$ : FV<sub>in</sub>から制御情報を入力し、DFへ格納する。DF=trueなら、(1,j)-格子点( $1 < j \leq J$ )の処理をする。DF=falseなら、(i,j)-格子点( $p < i \leq I, 1 < j \leq J$ )を処理する。

$PE(\xi, n)$ : (i,j)-格子点( $1 < i \leq I, 1 < j \leq J$ )の処理。(サイクル1°, 3°はアルゴリズムPEAと同様)

PEが実行するアルゴリズムPEASの詳細を図7に示す。

#### 4.3 アルゴリズムHADPSの正当性

HADPSの正当性を示すため、以下の定義を与える。

[定義5]  $x, y \in N$  ( $N$ は自然数全体の集合)に対し、次の関数を定義する。

$$\text{Mod}(x, y) = (x-1) \bmod y + 1$$

ここで、modは剰余関数である。これより、 $\xi = \text{Mod}(i, p)$ ,  $n = \text{Mod}(j, q)$ ,  $\tilde{n} = \text{Mod}(n, p)$ ,  $\tilde{J} = \text{Mod}(J, q)$ とする。■

[入力仕様3] 時刻  $t = k$  で、 $k = n + \tilde{n} + p \lfloor (i-1)/p \rfloor + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (j-1)/q \rfloor - 1$  ( $i=1, p+1, \dots, \lfloor (I-1)/p \rfloor p + 1, 1 \leq j \leq J, 1 \leq n \leq N$ )を満たす  $j$  に対し、 $a_j$  は  $PE(1, n)$  に入力する。■

[入力仕様4] 時刻  $t = k$  で、 $k = i + \tilde{n} + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (j-1)/q \rfloor - 1$  ( $1 \leq i \leq I, j=1, q+1, \dots, \lfloor (J-1)/q \rfloor q + 1, 1 \leq n \leq N$ )を満たす  $i, n$  に対し、 $b_i$  は  $PE(\xi, 1)$  に入力する。■

[入力仕様5] 時刻  $t = k$  で、 $k = i + \tilde{n} + I \lfloor (n-1)/p \rfloor - 1$  ( $1 \leq i \leq I, 1 \leq n \leq N$ )を満たす  $i$  に対し、 $EF(\xi, 1) = \text{true}$ 。その他の  $i$  で、 $EF(\xi, 1) = \text{false}$ 。■

[入力仕様6] 時刻  $t = k$  で、 $k = n + \tilde{n} + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (j-1)/q \rfloor - 1$  ( $1 \leq j \leq J, 1 \leq n \leq N$ )を満たす  $j$  に対し、 $DF(1, n) = \text{true}$ 。その他の  $j$  に対し  $DF(1, n) = \text{false}$ 。■

以下、HADPSの正当性は入力パターンと  $N$  個の標準パターンで形成される  $I \times J \times N$  のパターンマトリクス空間を  $1 \leq i \leq I, 1 \leq j \leq q, n=1$  (補題6),  $1 \leq i \leq I, 1 \leq j \leq q, 1 \leq n \leq N$  (補題7),  $1 \leq i \leq I, q < j \leq 2q, n=1$  (補題8),  $1 \leq i \leq I, 1 \leq j \leq J, 1 \leq n \leq N$  (補題9) に場合分けして証明する。

今、 $n=1, 1 \leq j \leq q$  とすると入力仕様3, 4より、次の補題を得る。

[補題5] 時刻  $t = k$  で、 $k = i + j - 1$  ( $1 \leq i \leq I, 1 \leq j \leq q$ ) を満たす  $i, j$  に対し

$$Rd(\xi, n) = d_{i,j}^1$$

が成り立つ。

(証明)  $a_j$  は入力仕様3より、 $t = j + p \lfloor (i-1)/p \rfloor$  で  $PE(1, n)$  へ入力される。今、 $k_1 = k - (j + p \lfloor (i-1)/p \rfloor) < p$  とすると、補題1の証明と同様の議論により、 $t = k$  で  $a_j$  は  $PE(k_1+1, n)$  に取り込まれる。同様に、 $b_i^1$  は  $t = i$  で  $PE(\xi, 1)$  へ入力されるので、 $k_2 = k - i < q$  とすると、 $t = k$  で  $b_i^1$  は  $PE(\xi, k_2+1)$  へ入力される。ここで、 $k = i + j - 1$  とすると  $k_1+1 = i - p \lfloor (i-1)/p \rfloor = \text{Mod}(i, p)$ ,  $k_2+1 = j = \text{Mod}(j, q)$  が成り立つ。よって、 $k = i + j - 1$  を満たす  $i, j$  ( $1 \leq i \leq I, 1 \leq j \leq q$ ) に対し、 $t = k$  で  $a_j, b_i^1$  は  $PE(\xi, n)$  へ入力され、補題2と同様の議論により  $Rd(\xi, n) = d_{i,j}^1$  が成立する。■

【補題6】 時刻  $t = k$  で、 $k = i + j - 1$  ( $1 \leq i \leq I, 1 \leq j \leq q$ ) を満たす  $i, j$  に対し

$$RD(\xi, n) = D_{i,j}^1$$

が成り立つ。

(証明) HADPSがHADPと同一の動作を実現することを言えばよい。 $a_j, b_i^1$ 間の距離  $d_{i,j}^1$  は補題5により正しく計算されるので、累積距離  $D_{i,j}^1$  がHADPと同様に計算できることを  $t$  に関する数学的帰納法により示す。PEが処理を担当するパターンマトリクス空間の境界条件の差異から、 $\xi = 1$  と  $1 < \xi \leq p$  に分ける。

##### 1. $1 < \xi \leq p$

$t = k$  ( $k = p \lfloor (i-1)/p \rfloor + \xi + j - 1, i \in \{i \mid \text{Mod}(i, p) \neq 1\}$ ) のとき、補題が成り立つことを  $j=1$  と  $1 < j \leq q$  に場合分けして示す。

a.  $j = 1$  のとき

$t = k$  で、入力仕様5より  $EF(\xi, 1) := \text{true}$  であるので、PEASより  $RDh(\xi, 1) := \infty, RDs(\xi, 1) := \infty, RDv(\xi, 1) := RD(\xi-1, 1)$  となる。帰納法の仮定より、 $t = k-1$  で  $RD(\xi-1, 1) = D_{i-1,j}^1$  が成り立つ。よって、 $t = p \lfloor (i-1)/p \rfloor + \xi$  のとき  $PE(\xi, 1)$  はHADPの  $PE(i, 1)$  と同一の動作を実現する。

b.  $1 < j \leq q$  のとき

$t = k$  で、PEASより  $RDv(\xi, n) := RD(\xi-1, n), RDh(\xi, n) := RD(\xi, n-1), RDs(\xi, n) := RD(\xi-1, n-1)$  となり、帰納法の仮定により

$$RD(\xi-1, n) = D_{i-1,j}^1, RD(\xi, n-1) = D_{i,j-1}^1, RD(\xi-1, n-1) = D_{i-1,j-1}^1$$

が成り立つ。よって、 $t = p \lfloor (i-1)/p \rfloor + \xi + j - 1$  のとき、 $PE(\xi, n)$  はHADPの  $PE(i, j)$  と同一の動作を実現する。

##### 2. $\xi = 1$

1と同様の議論から、 $t = p \lfloor (i-1)/p \rfloor + j$  のとき成り立つ。

従って、1および2から  $t = k$  で、 $k = i + j - 1$  ( $1 \leq i \leq I, 1 \leq j \leq J$ ) を満たす  $i, j$  に対し、 $PE(\xi, n)$  は

ADPのPE(i, j)と同一の動作を実現するので、補題3よりRD( $\xi, n$ ) =  $D_{i,j}^1$ が成り立つ。■

【補題7】時刻  $t = k$  で、 $k = i + j + \tilde{n} + I \lfloor (n-1)/p \rfloor - 2$  ( $1 \leq i \leq I, 1 \leq j \leq q, 1 \leq n \leq N$ ) を満たす  $i, j, n$  に対し  $RD(\xi, n) = D_{i,j}^n$  が成り立つ。

(証明) まず、 $1 \leq n \leq p$  なる任意の  $n$  に対し補題4の議論(時刻の始点を変更)と同様に  $t' = k - (n-1)$  とすると、入力仕様3から6により、 $t'$  で補題6が成り立ち、 $RD(\xi, n) = D_{i,j}^n$ 。よって、 $t = k$  で補題が成り立つ。次に、 $p < n \leq N$  なる  $n$  を考える。今、 $1 \leq n_1 \leq p, p < n_2 \leq N$  で  $\text{Mod}(n_1, p) = \text{Mod}(n_2, p) = \xi$  を満足する  $n_1, n_2$  を仮定する。このとき、入力仕様4より  $b_{i,j}^{n_1}$  および  $b_{i,j}^{n_2}$  を  $PE(\xi, 1)$  へ入力する時刻の差  $\Delta k$  は、 $\Delta k = I \lfloor (n_2-1)/p \rfloor$  で与えられる。 $t' = k - \Delta k$  とすると、 $t'$  で  $p < n_2 \leq N$  なる任意の  $n_2$  に対して、 $1 \leq n \leq p$  の議論から補題が成り立つ。よって、 $t = k$  で補題が成り立つ。■

次に、パターンマトリクス空間の残りの部分である  $1 \leq i \leq I, q \leq j \leq J$  に対しても正しく計算することを示す。

【補題8】時刻  $t = k$  で、 $k = i + n + IN/p - 1$  ( $1 \leq i \leq I, q < j \leq 2q$ ) を満たす  $i, j$  に対し

$$RD(\xi, n) = D_{i,j}^1$$

が成り立つ。

(証明) 省略(補題6と同様の議論による)。■

【補題9】時刻  $t = k$  で、 $k = i + n + \tilde{n} + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (j-1)/q \rfloor - 2$  ( $1 \leq i \leq I, 1 \leq j \leq J, 1 \leq n \leq N$ ) を満たす  $i, j, n$  に対し

$$RD(\xi, n) = D_{i,j}^n$$

が成り立つ。

(証明) 補題8に対し、補題7と同様の議論を行うことにより、 $q < j \leq 2q$  に対して補題が成り立つ。次に、ある  $i, n$  に対して、入力仕様3より  $q < j_1 \leq 2q$  なる  $a_{j_1}$  を  $PE(1, n)$  へ入力する時刻と  $2q < j_2 \leq J$  なる  $a_{j_2}$  を  $PE(1, n)$  へ入力する時刻の差  $\Delta k$  は  $\Delta k = IN/p \cdot (\lfloor (j_2-1)/q \rfloor - 1)$  であるので、 $t' = k - \Delta k$  を仮定すると、 $t'$  で  $2q < j_2 \leq J$  なる任意の  $j_2$  に対し、上で示した  $q < j \leq 2q$  と同様の議論から補題が成り立つ。ただし、 $\text{Mod}(j_1, q) = \text{Mod}(j_2, q) = n$  を満たす。よって、 $t = k$  で補題が成り立つ。■

補題9より、次の定理が導かれる。

【定理2】ハードウェア・アルゴリズムHADPSは、時刻  $t = I + J + \tilde{n} + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (J-1)/q \rfloor - 2$  ( $1 \leq n \leq N$ )、累積照合距離  $D(A, B_n)$  を正しく計算する。

(証明)  $D(A, B_n) = D_{I,J}^n / L$  より、 $D_{I,J}^n$  が正しく計算されることを示せばよい。今、 $\text{Mod}(I, p) = p$ 、

$\text{Mod}(J, q) = q$  とすると、補題9より、任意の  $n$  に対して  $i = I, j = J$  で  $RD(p, q) = D_{I,J}^n$  が成り立つ。よって、HADPSは累積照合距離を正しく計算する。また、このとき  $t = I + J + \tilde{n} + I \lfloor (n-1)/p \rfloor + IN/p \cdot \lfloor (J-1)/q \rfloor - 2$  は明らかである。■

## 5. 連続単語音声認識への適用

連続単語音声認識は次のように定式化できる[6]。

X個の単語列に対応する標準パターンの結合を

$$R = R^n(1) R^n(2) \dots R^n(X) \\ = b_1^n(1) b_2^n(1) \dots b_{I^n(1)}^n(1) \dots b_1^n(X) b_2^n(X) \dots b_{I^n(X)}^n(X) \\ = b_{1I}^R b_{2I}^R \dots b_{IR}^R$$

とし、入力パターンを

$$U = a_1 a_2 \dots a_j$$

で表す。このとき、RとUの最小累積照合距離が最小となる単語列、 $n(1) n(2) \dots n(X)$  を求める。

UとRの対応関係を表す時間変換関数を  $u(j)$  とすると、累積照合距離は

$$D(U, R) = \min_{\{j\}} \sum_{j=1}^j d^R(u(j), j)$$

である。ただし、 $0 \leq u(j) - u(j-1) \leq 2$ ,  $u(1) = 1$ ,  $u(J) = I^R$  を満たす。そして、 $D(U, R)$  を最小化する  $\hat{R} = \text{argmin} D(U, R)$  が求める解である。

各標準パターンに対して、入力パターンの各フレームで終端する最適な照合位置と照合距離が求めれば word spottingが実現できる。今、入力パターンの  $m \sim j$  フレームと標準パターン  $n$  の  $1 \sim j$  フレーム間の最小累積距離の  $m$  についての最小値を  $D_{i,j}^n$  とすると

$$D_{i,j}^n = \min_{\{k\}} \sum_{k=1}^j d^n(k, v(k)) \quad (4)$$

で与えられる。ここで、 $0 \leq v(k) - v(k-1) \leq 2$ ,  $1 \leq v(1) \leq j$ ,  $v(i) = j$  である。

(4)式は、前述の離散単語DPマッチングアルゴリズムにより同様に計算できる。ただし、連続単語音声認識では  $D_{i,j}^n$  に対応する入力パターンの始点位置情報  $B_{i,j}^n$  (バックポインタ)を一掃に求めておく必要がある。word spottingアルゴリズムは、次のように与えられる。

i) 境界条件

$$\text{for } n=1, 2, \dots, N, i=1, 2, \dots, I^n$$

$$D_{i,-1}^n = D_{i,0}^n = \infty$$

ii) 漸化式計算

$$\text{for } j=1, 2, \dots, J, n=1, 2, \dots, N$$

$$D_{1,J}^n = d_{1,J}^n$$

$$B_{1,J}^n = j$$

$$\text{for } i=2, 3, \dots, I^n$$

$$\hat{j} = \text{argmin}_{j-2 \leq j' \leq j} D_{i-1,j'}^n$$

$$D_{i,j}^n = D_{i-1,\hat{j}}^n + d_{i,j}^n, B_{i,j}^n = B_{i-1,\hat{j}}^n$$

ただし、DPパスは図8(a)を仮定している。上述のアルゴリズムにおいて、 $B_{i,j}^n$ の計算が追加されている点を除けば、ii)の漸化式計算は、2.で述べたアルゴリズムADPのそれと同等であるので、ハードウェア・アルゴリズムHADPおよびHADPSに $B_{i,j}^n$ の計算を組み込むことにより、連続単語のDPマッチングハードウェア・アルゴリズムが実現される。特に、実時間処理を考えると、入力パターン長が未知でも処理を開始することができるHADPSの有効性が示される。

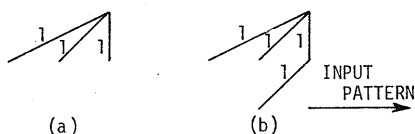


図8. 非対称DPパスと重みの例

### 6. アレイサイズ $p, q$ に関する議論

総計算ステップ数  $T$  は、定理2より次式で与えられる。

$$T = I + J + N - 2 + \lfloor \frac{N-1}{p} \rfloor (I - p) + \lfloor \frac{J-1}{q} \rfloor (\frac{IN}{p} - q)$$

上式より明らかに  $p = I, q = J$  のとき

$$T = I + J + N - 2$$

となり、定理1の結果と等しく、 $1 \leq p \leq I, 1 \leq q \leq J$  であることから、これがHADPSの最小計算時間である(逐次型計算機上では、アルゴリズムADPより、 $T = O(IJN)$ )。

$p, q$  のサイズと計算時間の関係を明確にするため  $N$  を定数として考えると、PEアレイの大きさ  $S$  が、 $S = p \times q$  のとき、計算時間  $T$  は

$$T = O(I + I/p + IN/pq)$$

$$= O(I + IN/pq).$$

これより、評価尺度として  $ST, ST^2$  を考えると

$$ST = O(pqI + IJ)$$

$$ST^2 = O(pq(I + IJ/pq)^2)$$

となる。ここで  $I \approx J$  とすると、 $pq = O(I)$  で  $ST, ST^2$  は最小となり

$$ST = O(I^2), ST^2 = O(I^3)$$

が最良のトレードオフである。次に、いくつかの単語を連続してパイプライン的に処理することを考える。このとき、結果が出力される平均周期  $P_{avo}$  は定理2より、 $P_{avo} = O(I/p)$  である。評価尺度として  $STP_{avo}$  を考えると

$$STP_{avo} = O(I^3/p)$$

を得る。 $p = O(I), q = O(1)$  のとき、 $STP_{avo} = O(I^2)$  で最良となる。実際、 $p = I, q = 1$  の線形アレイを考えるとパイプライン周期  $P$  は、 $P = 1$  である。

### 7. あとがき

連続単語音声認識向きのDPマッチングハードウェア・アルゴリズムを構成し、その正当性を示した。アルゴリズムHADPSは、 $p \times q$  のPEアレイを用いて、 $T = I + J + N - 2 + \lfloor \frac{N-1}{p} \rfloor (I - p) + \lfloor \frac{J-1}{q} \rfloor (\frac{IN}{p} - q)$  ステップで任意長の入力音声を正しく処理できる。さらに、入力パターン長には依存しないので、音声入力と同時に処理が開始でき、フレームに同期して処理が進むため、音声認識の実時間性は失われず、連続単語音声認識(word spottingに基づく)に適用できることを示した。これは、 $N$  個の標準パターンと入力パターンで形成されるパターンマトリクス空間を標準パターン方向にPEアレイを走査させることに起因する。

本稿では標準パターンの長さ  $I^n$  は、一定として議論を進めたけれども、一般には  $I^n$  は不均一で、例外処理等の制御が構造上の複雑さを増す一因となる。この点に関しては、今後更に検討することが必要である。

**謝辞** 本研究の一部は、文部省科学研究費補助金(一般研究(C) 課題番号62550262)によって行なわれた。ここに、謝意を表す。

### 参考文献

- [1] Burr, D.J., et al., "Array Configurations for Dynamic Time Warping", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-32, No.1, Feb. 1984, pp. 119-128.
- [2] Cheng, H.D., and Fu, K.S., "VLSI Architectures for String Matching and Pattern Matching", Pattern Recognition, Vol. 20, No.1, 1987, pp. 125-141.
- [3] 程, 他, "動的計画法に基づく配分問題のハードウェアアルゴリズム", 信学技法, CPSV86-27, 1986年9月.
- [4] 菊野, 他, "最大共通部分系列を求めるハードウェア・アルゴリズム", 信学論(D), J65-D, No.8, 1982年8月, pp. 997-1004.
- [5] 前場, 辰巳, "局所制御型プロセッサアレイの音声認識処理への適用", 信学技法, EC85-58, 1986年2月.
- [6] 中川聖一, "拡張連続DP法による連続音声認識アルゴリズム", 信学論(D), J67-D, No.10, 1984年10月, pp. 1242-1249.
- [7] Sakoe, H., and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-26, No.1, Feb. 1978, pp. 43-49.
- [8] 矢島, 富岡, 都倉編, 特集「VLSI向きハードウェアアルゴリズム」, 情報処理, Vol. 26, No.6, 1985年6月.