九州大学学術情報リポジトリ
Kyushu University Institutional Repository

筆順フリーなオンライン文字認識のための画対応サーチアルゴリズム

迫江, 博昭
知能システム学専攻

愼, 重弼
知能システム学専攻 : 博士後期課程

https://doi.org/10.15017/1485126

出版情報：九州大学大学院システム情報科学紀要. 2 (1), pp.99-104, 1997-03-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

KYUSHU UNIVERSITY

九州大学大学院
システム情報科学研究科報告
第2巻 第1号 平成9年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.1, March 1997

筆順フリーなオンライン文字認識のための画対応サーチアルゴリズム

迫江博昭* ・ 愼 重 弼**

A Stroke Order Search Algorithm for Online Character Recognition

Hiroaki SAKOE and Jung-Pil SHIN

(Received December 24, 1996)

Abstract: An efficient stroke-to-stroke correspondence search algorithm is proposed with application to stroke-order-free online character recognition. An exponential order algorithm complexity is achieved by using Dynamic Programming controlled by a cubic form automaton whose nodes correspond to matching status of reference strokes to input strokes. By preliminary experiments, it is demonstrated that the proposed algorithm works fast enough for practical application. A generalization to stroke–number free recognition is also investigated.

Keywords: Online character recognition, Stroke order, Stroke-to-stroke correspondence, Dynamic programming, Beam search

1. ま え が き

オンライン文字認識における筆順に関しては2つの見方がある．その1は，せっかく筆順情報が使えるのだから，これを積極的に利用すべきとするものであり，他の1つは，筆順の変動は避けられないので，ぜひとも筆順フリー性を実現すべきとするものである．著者らの立場は基本的に後者にあり，認識の枠組みは筆順フリーとすべきで，その上で，利用できる筆順は局所的，例外的に利用すればよいと考えている．特に，多画の漢字を用いる日本語入力においては，筆順フリー性は絶対的に必要と言える．

漢字を対象として筆順問題に取り組んだ研究としては小高，若原，増田[1)]，若原，小高，梅田[2)]，若原[3)]があげられる．入力の各画と標準の各画の間の距離(画間距離)を要素とする画間距離行列を用意し，この上で入力の画と標準の画の対応を定めるという，正攻法的なアプローチである．問題は，画対応の決定法である．画対応には，基本的に1対1対応であることが要求され，未対応画や，重複対応画の発生は好ましくないと考えられる．文献1)では，標準の画に最も近い(画間距離が小さい)入力画をローカルに選択して対応付けており，1対1対応の保証がなく，また，対応決定の不安定性が予測される．文献2),3)では，画間距離の総和を最小にするという大局的な評価基準を示しているが，その計算はローカルな対応付け処理によってなされている．もし厳密解を総当たり法で求めるとすると，計算量は，画数をNとして，$O(N!)$になる．

本研究は，これら先駆的検討がなされて後の，現在までの計算機の進化を考慮して，文献2),3)で用いられた，「画間距離の総和を最小にする1対1対応」を求める現実的なアルゴリズムを検討したものである．基本的な原理は，一方のパターンの筆順にしたがって，画を進めることとし，他方の画の対応済みの画の組合せパターンを状態とする有限状態オートマトン(FSA)を用いて，多重対応や対応もれを排除するというものである．このFSAはキューブ状のグラフとなり，対応付け状態が頂点(ノード)，各画の対応が，辺(エッジ)に対応する．画の最適対応付け問題は，画間距離をコストとして，このキューブ状グラフ上での最適経路問題として定式化される．DPで計算され，N画文字の場合の計算量は$O(N \cdot 2^{N-1})$，所要記憶量は$O(2^N)$である．20画程度の漢字への適用を考慮して，ビームサーチの導入による高速化を検討して効果を確認した．

なお，本論文では楷書という前提をおいて議論を進める．すなわち，画と画の間には，必ずペンアップがあるとする．これによって，入力の画数と同じ画数の標準パターンとのみマッチングすればよいとした．また，個々の画は正しい方向で筆記されているものとする．

続け書きによって画の接続が発生した場合に対処するための画数フリーアルゴリズムに関しては，上記のグラフの拡張として6.2節で議論する．

2. 問題の設定と準備

オンライン入力文字を次のように筆記順の画の系列として表現する．

平成8年12月24日受付

* 知能システム学専攻
** 知能システム学専攻博士後期課程

$$A = A_1 A_2 \cdots A_k \cdots A_N \tag{1}$$

ここにA_kは第 k画で，文字の局所的特徴(例えば筆線方向，座標等：a_{ik})の時系列表現である．

$$A_k = a_{1k} a_{2k} \cdots a_{ik} \cdots a_{Ik}, \quad I = I(k)$$

同様に標準パターンを次のように表現する．

$$B = B_1 B_2 \cdots B_l \cdots B_N \tag{2}$$
$$B_l = b_{1l} b_{2l} \cdots b_{jl} \cdots b_{Jl}, \quad J = J(l)$$

画A_kとB_lの間の相違度の評価値を

$$\delta(k,l) = D(A_k, B_l) \tag{3}$$

と示し，画間距離と呼ぶ．すべての(k,l)の組に関して画間距離を求めることとし，これをテーブルに記憶する．

$$\delta(k,l), \quad k = 1,2,\cdots,N; \quad l = 1,2,\cdots,N \tag{4}$$

ここでは，画間距離の評価にDPマッチングを用いることとする．その漸化式は

$$g(i,j) = d(i,j) + \min \begin{bmatrix} g(i-1,j) \\ g(i-1,j-1) \\ g(i-1,j-2) \end{bmatrix} \tag{5}$$

のタイプとする．ここに，$d(i,j)$はa_{ik}とb_{jl}との距離とする．

いま，入力の第 k画に対応付けられる標準パターンの画を$l(k)$とする．この時，画対応サーチの問題は次のように定式化される．

「画の対応$l(1)l(2)\cdots l(N)$を最適に定めて，$\delta(k,l(k))$の総和を最小とする．」すなわち，次の最小化問題として定式化される．

$$D(A,B) = \min_{\{l(k)\}} \left[\sum_{k=1}^{N} \delta(k,l(k)) \right] \tag{6}$$

この問題を解くことは，構造解析を行なうことであり，結果として，パターン間距離$D(A,B)$と画対応$l(k)$が得られる．上記のように画間距離上で(6)式を計算するのは文献 2),3)と同様であり，これが本研究の出発点となる．なお，本論文では「正確に書かれた楷書」を対象としており，上記の対応に「１対１」の性質を要求する．画数フリーを実現するためにはこの枠を超える必要がある．これに関しては 6.2節で簡単に可能性を示す．

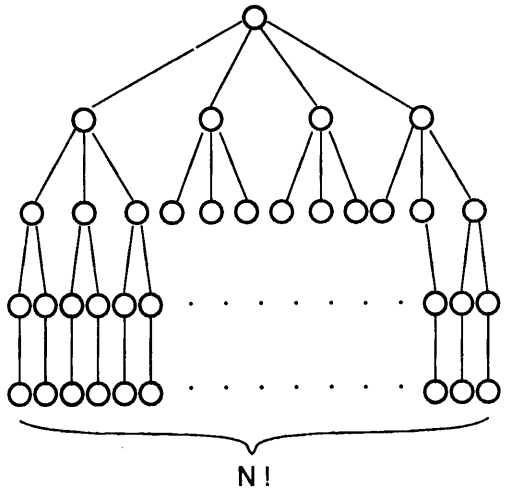

Fig.1 Stroke order search tree (stroke number N= 4)

3. 画対応サーチの基本原理

上記 (6)式をシステマティックに解こうとする場合，まず考えられるのは，**Fig. 1**のような枝分かれ型のグラフにおけるサーチ問題としての定式化である．深さが入力の画番号 kに対応する．その段の各エッジはその入力画に対応付けられる標準画 lの選択を示し，画間距離$\delta(k,l)$が付随する．ノードは書き始めからの画対応の情報を担う．この問題を直接計算すると，計算量，ワークエリア記憶量とも

$$\sum_{k=1}^{N} k! = O(N!) \tag{7}$$

となり，多画文字の場合実用性に欠けると言える．

以下では，このグラフの冗長分を除去し，且つＤＰによる解法を示し，計算量を$O(N \cdot 2^{N-1})$に，ワークエリア記憶量を$O(2^N)$ に，それぞれ低減する．ついで，ビームサーチを導入して実用的なアルゴリズムを提案する．

3.1 キューブ状画対応サーチグラフ

基本的にあらゆる画順が可能であるとしてサーチする場合，制約は，「画対応において，同一の画は１度しか現われない．また，すべての画が現れなくてはならない」という簡単な事実である．然るに，**Fig. 1**のグラフの各ノードは，そこまでの画対応の順序の情報までも担っており，冗長と言える．すなわち，１対１画対応の過程で，次の対応を決定するのに必要なのは，過去にどれどれの画が対応付けられているかという情報で充分なのである．

上記の冗長分を除去した画対応サーチグラフとして**Fig. 2**を提案する．このグラフを画順生成のモデル(FSA)として用い，入力パターンの画順を解析する．横

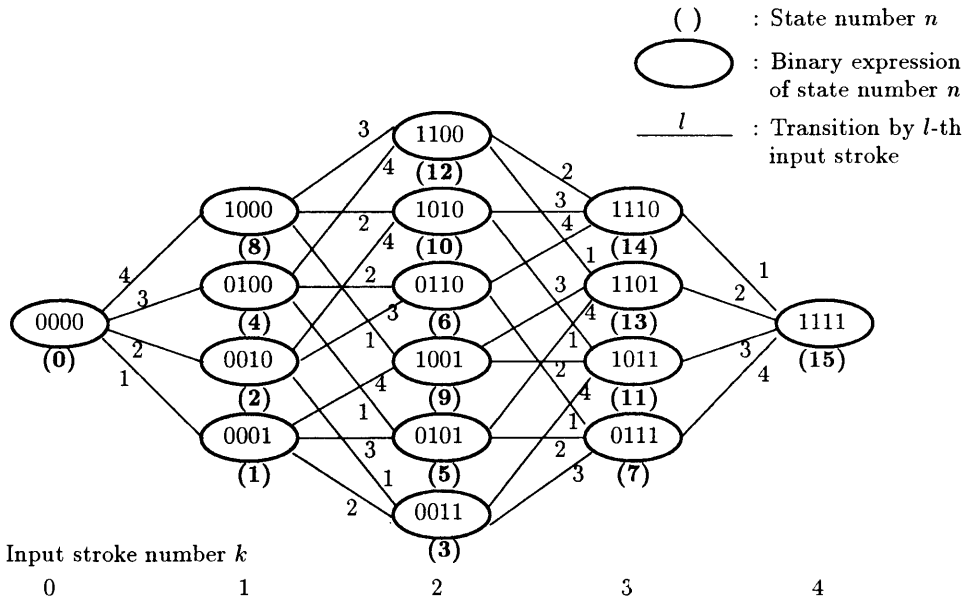

Fig.2　Cube Search graph(stroke number N= 4)

方向には入力の画番号 kが対応する．これが画対応サーチＤＰの段(stage)となる．各段 kは，そこに至るまでに，入力の$1,2,\cdots,k$画に対して，標準パターンのいずれかのk画が重複・脱落なく，なんらかの順序で，対応付けられたということを意味する．

各段 kのノードには状態 nが対応する（以下ではグラフと FSA，エッジと遷移，及びノードと状態を同義語として使用する）．各状態 nは Nビットよりなる．各ビットは既対応を示すフラグの働きをする．すなわち，lsb から$1,2,\cdots,N$と番号付けられるビット位置は標準パターンの$1,2,\cdots,N$画に対応する．第 k段では，その内kビットのみが１となる．この１となっている k個のビットは標準パターンの画の内，これらに対応する画が，なんらかの順序で既にマッチング済みであることを示す．状態遷移は第 k段から第 $k+1$段への推移にともなって，いずれかの１ビットが，０から１に変化するという形になる．第 lビットが反転するエッジにはコストとして画間距離 $\delta(k,l)$が付随する．グラフ全体としての初期状態は $0=(0,0,\cdots,0)$，最終状態は $2^N-1=(1,1,\cdots,1)$である．

このように，画対応サーチの問題は，$\delta(k,l)$をエッジコストとする N次元キューブ状グラフにおける，ノード$(0,0,\cdots,0)$から，ノード$(1,1,\cdots,1)$への最適経路問題となる．このグラフをFSA として構造解析を行なうことによって，各遷移が，0→1の反転を１ビットだけ行なうことで１対１対応が保証され，最終状態を$(1,1,\cdots,1)$とすることで未対応画の発生が排除される．

なお，ここで定義した状態 nはビット系列としては上記意味のフラグの組であり，数値としてはワークエリアの番地指定に用いられることを注意しておく．

3.2　ＤＰによる画対応サーチ

上記最適経路問題は，ＤＰによって計算することができる．

1. 画順サーチ処理の初期化
   $G(n)=\infty,\quad n=1,2,\cdots,2^N-1$
   $G(0)=0$
2. 画順サーチ処理$(k=1,2,\cdots,N)$
   (2-1)　第$k-1$段の各状態 nを生成しながら(2-2) ～ (2-4)を実行．
   (2-2)　nのビットの内'0'であるものを1個を'1'に変え → nw.
   そのビットの位置を → l.
   (2-3)　$\min[G(n)+\delta(k,l),G(nw)]\rightarrow G(nw)$
   (2-4)　'1'を立てるビットの位置を変えながら(2-2)以下を繰り返す．
3. 終結
   入力 Aと標準 Bのマッチング結果としてパターン間距離
   $D(A,B)=G(2^N-1)$

を得る．

以上のアルゴリズムでは，ワークエリア記憶量（ノード数）は2^N個であり，$N!\rightarrow 2^N$の低減が達成された．計算量は遷移（エッジ）数で評価されるが，

$$\sum_{k=1}^{N}(N-k)\cdot {}_NC_k=N\cdot 2^{N-1} \tag{8}$$

となる．よって，$N!\rightarrow N\cdot 2^{N-1}$の低減が達成された．アルゴリズム論の見地からは，依然として指数オーダーの複雑さであり，NP完全問題に属する．しかし，オンライン文字認識においては，$N=20$程度でよく，かつ底が2という実効的な最小整数であるという好条件もあって，最近の計算機では，忌避すべきオーダーとは言えない．しかし，実時間動作の要求と，経済性の要求を考える場合には，当然，より高効率なアルゴリズムが必要となる．

4. ビームサーチDP

前節に述べた原理を実行するには次の問題点がある．

(1) 計算量が大である．計算量((8)式)は依然として大きい．画数$N=20$では，$N\cdot 2^{N-1}=5.2\times 10^6$となる．

(2) (2-1)における状態nをいかに効率よく生成するか．第k段の状態nはNビットの内kビットのみが1になっているビットパターンであるが，それらを効率よく生成するアルゴリズムが見あたらない．このようなビットパターンは${}_NC_k$種あり，テーブル参照方式で生成するのは得策でない．$({}_{20}C_{10}=1.8\times 10^5)$

まず(1)に対しては，ビームサーチを導入する．すなわち，各段で$G(n)$の値をチェックして最適解としての可能性の低い状態を以後の探索から排除するという枝刈を行なう．(2)の問題はビームサーチに関連して解決する．すなわち，$(k-1)$段の枝刈で残ったアクティブな状態nを基にして，第k段で必要な状態nを生成するという方式をとることとする．

Fig. 3に，ビームサーチDPの詳細を示す．枝刈りの閾値θは文献4)を参考として第k段での$G(n)$の最小値に余裕常数λを加えた値とした．ワークエリア$mbeam$と$nbeam$によってビーム内に残った状態nを集中管理し，アクセスの効率を高めている．⑦によって第$(k-1)$段のノードnに後続し得る第k段のノードnwを生成することにより，前記(2)の問題を解決している．

画対応$l(k)$が必要な場合には$G(n)$あるいはバックポインタテーブル上でのバックトラックによって求めることができる．

5. 実　　験

主としてアルゴリズムの速度を知る目的で，一部の漢字を対象として実験を行なった．文字データは，32×32の平面上に表現される．筆点上の各点の特徴情報a_{ik}やb_{jl}としては，その点のxy平面座標と8方向コードとの組を用いた．画間距離は，xy平面座標のDP距離と8方向コードのDP距離の加重和を用いた．なお，実験にはhyperSPARC CPU(78.0 SPECint92, 102.0 SPECfp92)のワークステーションを使用した．

[ワークエリア]

$\delta(k,l)$　　**：画間距離**

$G(n)$　　**：DP ワークエリア**

$mbeam(k)$ **：第k段でアクティブであった状態の個数を記憶する．**

$nbeam(k,m)$**：第k段でアクティブな状態 nを記憶する．**

$m=1,2,\cdots mbeam(k)$

[ビームサーチアルゴリズム]

① **初期化**

$G(n)=\infty,\quad n=1,2,\cdots 2^N-1$

$G(0)=0$

$mbeam(0)=1$

$nbeam(0,1)=0$

$G_{\min}=0$

$\delta(k,l)$**を生成**

② $k=1,2,\cdots N$ **について**③～⑬**を実行**

③ $\theta=G_{\min}+\lambda$　**(閾値の生成．λは余裕定数)**

$mcount=0$

$G_{\min}=\infty$

④ $m=1,2,\cdots mbeam(k-1)$ **について**⑤～⑫**を実行**

⑤ $n=nbeam(k-1,m)$

$G_w=G(n)$

if $G_w>\theta$　**go to** ⑫

⑥ $l=1,2,\cdots N$ **について**⑦～⑫**を実行**

⑦ n**の第 lビットが '1'ならば go to** ⑫

else　**そのビットを '1'に反転し→** nw

⑧ **if** $G(nw)\neq\infty$　**go to** ⑨**,**

else　$nbeam(k,mcount)=nw$,

$mcount=mcount+1$

⑨ $G_x=G_w+\delta(k,l)$

⑩ **if**　$G(nw)\leq G_x$　**go to** ⑫**,**

else　$G(nw)=Gx$

⑪ **if**　$G_x<G_{\min}$,　$G_{\min}=G_x$

⑫ **continue**

⑬ $mbeam(k)=mcount$

⑭ **終結：入力Aと標準Bのマッチング結果**

$D(A,B)=G(2^N-1)$

Fig.3　Beam Search DP algorithm

5.1 実験（1）

Table 1に示す4,8,16,20画の文字，各10字を用いて，余裕定数λをパラメタとしてビームサーチ効果を調べた．上記10字に標準パターン各1個を用意し，各文字20個ずつを入力として，入力と同じ標準パターンとのマッチン

Table 1 Test kanji characters(Experiment (1))

4 画	反, 友, 木, 六, 少, 介, 五, 区, 中, 云
8 画	知, 和, 昔, 青, 幸, 供, 征, 東, 枝, 定
16 画	錬, 錠, 親, 頭, 謡, 賢, 薬, 瞠, 錯, 謀
20 画	競, 護, 議, 懸, 譲, 醸, 響, 籍, 露, 鐘

Table 2 Result of experiment(1) (4-20 stroke number's character matching)

N	B-S margin λ	1	2	3	4	5
4	Success.(%)	100	100	100	100	100
	B-size (%)	24.5	24.5	24.6	24.7	24.8
8	Success.(%)	100	100	100	100	100
	B-size (%)	2.30	2.35	2.41	2.46	2.51
16	Success.(%)	99.50	99.50	99.50	100	100
	B-size (%)	0.016	0.018	0.021	0.023	0.027
20	Success.(%)	97.57	98.12	99.05	100	100
	B-size (%)	0.001	0.001	0.002	0.002	0.002

Table 3 Result of experiment(2) (10 stroke number's character recognition)

Writer	Success. match. (%)	Recog. rate (%)	Search time (*sec*)
A	98.85	98.47	0.25
B	100.0	100.0	0.20
C	100.0	99.64	0.22
D	100.0	100.0	0.18
Average	99.71	99.52	0.21

グが成功する率(Success.(%))と，10個の標準パターンとのマッチングにおいてビーム内に残り，実際にＤＰが計算された遷移（エッジ）の個数(B-size(%); 100%は $N \cdot 2^{N-1}$)を測定した．結果を **Table 2**に示す．

5.2 実験（2）

10画のJIS第１水準漢字 290種を用いて，入力と同一の標準パターンとのマッチングが成功する率(Success. match)，認識率(Recog. rate) 及び画対応サーチ時間(Search time)を測定した．結果を **Table 3**に示す．

6. 考　　察

6.1 実験結果の検討

以上の実験により，提案したアルゴリズムの有用性が確認された．実験(1)では，ビームサーチの効果が多画文字の場合ほど大であることが観測された．小画数の場合のビームサーチの効果が低い点は，もともと計算量が少ないので問題にならない．

実験(2)では 10画のJIS第 1水準全漢字を用いて画対応サーチ時間を測定し，実時間性を確認した．20画の漢字

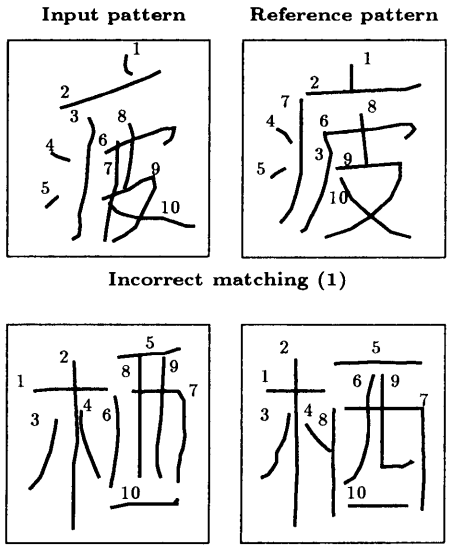

Fig.4　Incorrect matching examples

は種類が少ないこと，実験(1)で，20画の場合，16画の場合に比して，ビームサーチの効率が１桁高いこと等を考えると，20画でも充分な認識速度が達成されると考えられる．

Fig. 4に誤った画対応の例を示す．一部対応を間違えていて，それでも認識は正しく行なわれている場合も少なくない．文献 1)の場合と同様に，漢字パターンの冗長性に救われたものと考えられる．

6.2 画間続け書きへの対応

現実のオンライン文字認識では，続け書きによる画間接続の発生を考慮する必要がある．以下では，連続 3画以上の続け書きは生じないとした場合を例にとって，画数フリーな画対応サーチアルゴリズムの考え方を述べる．画間距離テーブルとして

$$\delta_1(k,l), \quad k=1,2,\cdots,M,\ l=1,2,\cdots,N$$
$$\delta_2(k,l,m), \quad k=1,2,\cdots,M,\ l=1,2,\cdots,N,\ m=1,2,\cdots,N,\ l\neq m$$

を用意する．ここに，M は入力文字の画数であり，$N-C \leq M \leq N$とする．すなわちC 回までの画間接続を許容するものとする．δ_1 は(4)式の δと同じものである．δ_2は，入力文字の第k画が標準の第 l画と第 m画の接続した，あるいは第 m 画と第 l画が接続した，連続画に対応付られた場合のＤＰ距離である．

$$\delta_2(k,l,m) = min[\,\delta(k,l,m),\ \delta(k,m,l)]$$

ここに，$\delta(k,l,m)$は l→mの接続, $\delta(k,m,l)$は m→lの接続に対応する．これらは標準の第 l画の終端から第 m画

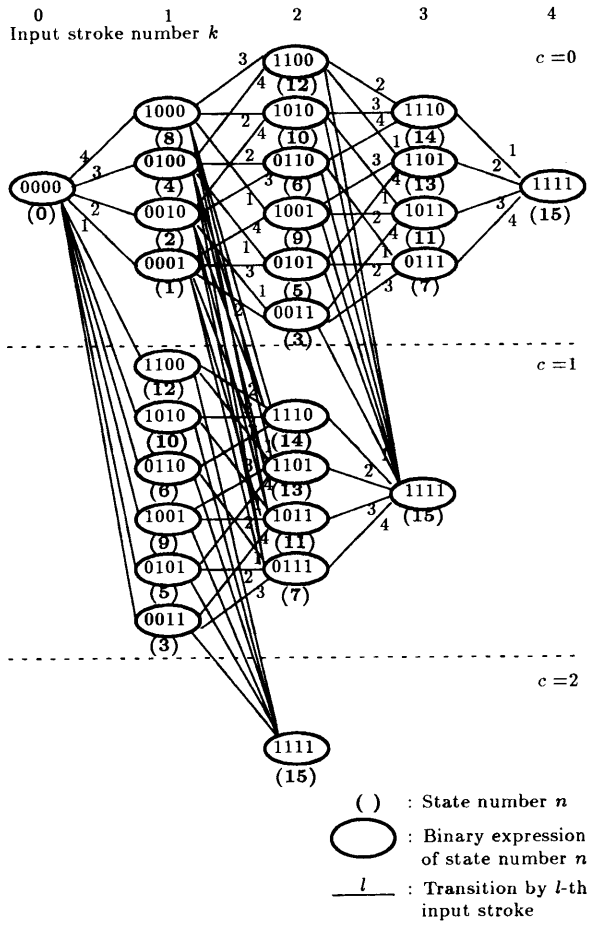

Fig.5 Multi-Layer Cube Search graph($N = 4$)

への仮想の線分を補って，第 m画の終端までのＤＰマッチングを実行すること等により得られる．

画順と画接続の同時サーチには，**Fig. 2**のキューブグラフ状FSA の上に，その部分構造を積層した FSA を用いる（**Fig. 5**）．層を番号 $c = 1, 2, \cdots, C$で識別する．cは画間接続が生じた回数を意味する．すなわち，画間接続が無い場合は同一層で遷移し，δ_1をコストとするのは，３～４節と同様である．画間接続が生じた場合は，$c+1$の層に遷移し，δ_2をコストとする．すなわち，その遷移で状態 nの第 lビットと第 mビットが0→1 の反転を行なう場合 $\delta_2(k,l,m)$をコストとする．

画接続の導入によって，計算量，ワークエリア記憶量は増加するが，FSA の構成からみて，爆発的な増加はないと言える．前節までの画間接続が無い場合の C 倍よりは，相当小さい．また，常識的に生起し得ない画接続の遷移や状態を排除するという高速化が有効と考えられる．

なお，3画以上の連続画接続，画切断を考慮した FSA も構成することができるが，その実用性に関しては，複雑さ，認識性能の両面からの検討が必要であろう．

7. あ と が き

画数の階乗オーダーの複雑さとされていた，オンライン文字認識の画順変動問題に対して，キューブ状のFSA とＤＰを用いる画対応サーチ手法を提案し，指数オーダーの複雑さに簡単化した．さらに，ビームサーチを組み込んだアルゴリズムを示し，実験により実時間認識の実現性を確認した．さらなる拡張としては画数フリー化，画相互間の位置情報(例えばペンアップ移動の方向や距離の評価)の利用等が考えられるが，これらに関しては稿を改めて報告したい．

参 考 文 献

1) 小高; 若原; 増田: “筆順に依存しない手書き文字 認識アルゴリズム”, 信学論文誌 **J65–D**, 2, pp.679–686 (1982–02).
2) 若原; 小高; 梅田: “選択的ストローク結合による画数・筆順に依存しないオンライン文字認識”, 信学論文誌 **J66–D**, 5, pp.593–600 (1983–05).
3) 若原: “発見的組合せ探査による点対応の決定”, 昭和 63 信学講論 **D–450** (1988–03).
4) 迫江博昭; 藤井浩美; 吉田和永; 亘理誠夫: “フレーム同期化, ビームサーチ, ベクトル量子化の統合によるDP マッチングの高速化”, 信学論, **J71-D**, No. 9, pp. 1650–1659(1988–9).