

輪郭特徴を基にした手書き漢字認識システムの
SIMD計算機による実現

佐々木 一陽 高橋 義造

徳島大学工学部知能情報工学科

E-mail: {sat,taka}@n30.is.toushima-u.ac.jp

並列手書き漢字認識に用いる文字特徴として画素間の相対位置を示した画素特徴と、それらを一次的に結合して文字の輪郭を表す輪郭特徴について提案する。また、これらの特徴と並列化したDPマッチングをSIMD型小形並列プロセッサLISCAR上で実現する手法について述べる。

Realization of Hand-printed Kanji Recognition System using
Character Outline Features on a SIMD Machine

Kazuaki Sasaki Yoshizo Takahashi

Department of Information Science and Intelligent System,

Faculty of Engineering, Tokushima University

2-1 Minami-josanjima-cho, Tokushima 770, Japan

E-mail: {sat,taka}@n30.is.toushima-u.ac.jp

We propose a massively parallel hand-printed Kanji recognition system using the character outline features and present the implementation on a SIMD machine LISCAR utilizing SIMD DP matching algorithm. The result of simulation on character samples are presented.

1. はじめに

我々の研究室では、以前より並列文字認識の必要性を考え、実際に並列計算機を用いて実験を行ってきた。この研究の過程で、これまでの文字認識に用いられている手法をそのまま並列化するのは、適当でないことが明らかになった[1][2]。従来より文字認識に用いられてきた線や形に関する特徴は、画素より得られる局所的な方向成分や形状の集積や、画素密度など直接的な情報が多く、かつ処理が逐次的で並列処理に向かない。そこで今回、並列抽出を目的とし、文字全体の情報を反映しつつ、局所的に処理が行えるものとして、大域的な性質を持ち、文字の正規化に使用される二次モーメント量[3]、すなわち画素間の距離と角度を特徴として利用することについて考えた。この特徴を画素特徴とし、更にこれを角度を中心に一元化したものを、文字の輪郭や形状を表した輪郭特徴として並列に抽出する。認識については画素の対応関係を調べる方法で並列化を図った。更にSIMDマシンによる並列DPマッチングについての研究も行った。

2. 従来の逐次的な特徴抽出法

従来からの画素より端点、支点、分岐点で分解された線分として定義されるセグメントと、文字を書く際、一面に相当するストローク[4]などを抽出する代表的な方法について、並列処理に対するの難点や問題点を述べてみる。

まず、比較的単純な方法として、ある一定の距離、通常は4または8連結した画素群を各方向別にクラスタリングすることによるラベル付けによる方法[5]が挙げられる。この方法は、縦、横方向のセグメントやストロークの抽出は容易であるが、斜め方向や丸みを帯びたストロークの抽出は困難である。そして、並列化は各ストロークの方向分程度に留まり、更に抽出された特徴を様々な方法で組み合わせて認識する必要性があり、認識部の並列化が困難である。

次に、まず入力パターンに対して画像処理を施す、細線化による方法であるが、線分の大きさなど冗長な情報を除去でき、セグメントやストロークの方向性を決定付ける方向成分の抽出が簡単であるため、その有効性ゆえ様々な手法が研究されている[6]が、決定的なものも確立されていない。しかし、細線化は比較的局所的に行えるので並列処理には向いているであろうと思われる。

そして、断片的な線分の集合から、より有効な線特徴

を抽出できる、Hough変換による方法[3]がある。これも直線に対しては非常に有効であるが、直線以外だとパラメータ空間が三次元以上となり、高次元のクラスタ化問題の解法が困難であるといわれている。なお、並列化に伴う領域の割当てやプロセッサ間の通信なども問題となる。

更に、弛緩法による認識法も手書き漢字認識に対して用いられてきたが、マッチング部分の並列化は線分単位で行わせることができ、各処理の計算量も多く超並列処理にも向いていると思われる。しかし、用いる特徴、すなわち多角形近似による線分の抽出に問題がある。

以上の手法に対して、我々は並列化の容易な手法をとることにした。並列に各プロセッサが、独立して文字を構成する画素間の相対位置を調べ、それを結合したものが結果的に輪郭の特徴となる方法である。この方法は手書き文字の特徴としては、有効であると思われる。この方法については次章に述べる。

3. 並列処理向き輪郭特徴抽出法

ここで用いる輪郭特徴は、単に文字の輪郭を表した特徴を直接示すのではなく、各画素特徴がその画素からみた文字全体の輪郭および骨格を表していることを示すのである。以下では画素特徴と輪郭特徴について述べる。

(1) 画素特徴

画素の文字における相対位置を全画素間の距離と角度で表したものを、画素特徴とする。この特徴は、文字全体の情報を反映しつつ、局所的に処理が行えることが利点となっている。その反面、全画素に対して同様の処理を行うため、画像データサイズの大きさはもちろん、画素の量に大きく影響され、処理量も膨大なものとなり、従来の汎用計算機単体では実時間で処理は不可能となる。そこで、各画素における処理の完全独立化に対して、超並列計算機を用いることで解決を図る。

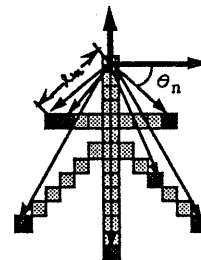


図1. 画素特徴

画素特徴は以下のように定義する。

画素座標 d

$$d_1 = (x_1, y_1)$$

$$d_p = (x_p, y_p)$$

画素特徴 dot

全画素特徴 $D = \{dot(d_n) \mid 1 \leq n \leq p\}$

$$dot(d_n) = \{l_m(d_n), \theta_m(d_n) \mid 1 \leq m \leq p, m \neq n\}$$

画素間距離 $l_m(d_n) = |x_n - x_m| + |y_n - y_m|$

画素間角度

$$\theta_m(d_n) = \begin{cases} \text{atan}((y_n - y_m) / (x_n - x_m)) & x_n \neq x_m \\ \text{atan}(y_n - y_m) & x_n = x_m \end{cases}$$

(2) 輪郭特徴について

各画素で求められた画素特徴を用いてDPマッチングを行うためには、二次元である画素特徴を何等かの方法により一次元化する必要がある[7]。この一次元変換では、各画素の相対位置を失わずに画素間の結合を一意に決定付けなければならないので、理想としては、筆順による結合方法、線幅を考慮に入れた一筆書きで、輪郭などを基準に結合していく方法などが考えられるであろう。前者の筆順による方法は、まず認識対象画像のどの線分が各ストロークに対応しているかを決定付ける必要がある。これは手書き漢字におけるストローク抽出の困難さと同様のものである。後者は、局所的な画素間の連結性に大きく左右されてしまい、正しい結合を得るのは困難であろうと思われる。

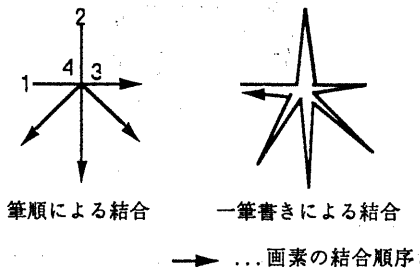


図2. 筆順、一筆書きによる画素結合方法

そこで、単純に画素特徴の角度を基準（一定方向に増加または減少）として結合していき、更に同一角度であれば距離を基準（同じく増加もしくは減少）として結合する方法を考えた。この方法は、基準となる画素より放

射状に走査線を出して、線幅をもった各線分を横切った断面を表すことになる。これに角度の連続性を加えると、間接的に線分の連結性も反映され、その延長として基準画素からみた文字の輪郭特徴も表すことになる。線幅の違いはDPマッチングにより吸収でき、また文字の回転に対してはマッチングにおける始点をずらすことで対処できる。更に、処理的に単純でかつ独立して行うことも利点となっている。

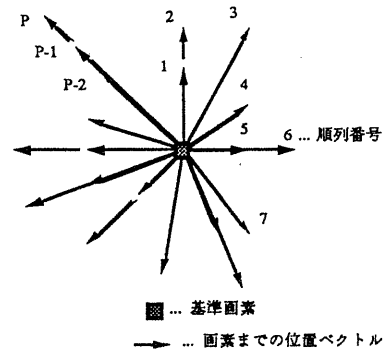


図3. 角度と距離を基準とした画素結合の一方

輪郭特徴は以下のように定義する。

$$dot(d_n) = \{l_m(d_n), \theta_m(d_n) \mid 1 \leq m \leq p, m \neq n\}$$

輪郭特徴 dot'

$1 \leq m \leq p$ において、以下の条件の下に並び替える。

- i) $\theta_m(d_n) \geq \theta_{m-1}(d_n)$
- ii) $\theta_m(d_n) = \theta_{m-1}(d_n)$ のとき $l_m(d_n) \geq l_{m-1}(d_n)$

よって、

$$dot'(d_n) = \{l_{m'}(d_n), \theta_{m'}(d_n) \mid 1 \leq m' \leq p, m' \neq n\}$$

欠点としては、文字を構成しない画素、つまりノイズに弱くなることが挙げられる。これは、基準パターンに画素に入力パターンに画素を対応付けることにより解決できるのではないかと考えている。つまり、基準パターンは通常、形状が整っており、かつノイズのないものとしているのが、入力パターンはその逆で形状も比較的崩れており、かつノイズも存在する可能性もある。そこで逆の対応付け、入力パターンに画素に基準パターンを対応付けると、次章で述べる最小距離優先による投票制のためノイズに対しても対応付けが行われる場合が起こるが、先の方法ではノイズよりも正しい画素間の対応付けが優先される可能性の方が大きいであろうと思われる。

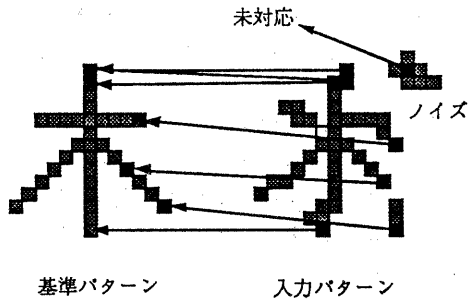


図4. 対応付けによる入力パターン上のノイズ対策

3. 並列認識システムの構成

(1) DPマッチングの並列化

DPマッチングは、音声認識の分野ではかなり研究開発が行われ、基本的なマッチング手法として確立しているように、連続性をもつ次元パターンに対して逐次的に操作することを前提としてマッチングを行うと、非常に有効であることが分かっている。そのため、DPマッチングを二次元データに対して行うと、空間的な連続性を表現できず、対応付けに失敗してしまう。これに関しては処理の並列化だけでは対処不可能であると思われるが、この二次元DPマッチングの手法が確立されれば、視覚パターン認識分野の研究に飛躍的な向上をもたらすことは間違いないであろう。

そこで、現段階ではDPマッチングを一次元データに対して用いることにし、各画素毎に並列的に行うことに留める。

(2) 認識の並列化

認識システムを構成するにあたって、理論上プロセッサは無尽蔵にあるという事を前提としているので、必要とあらばプロセッサはふんだんに使用してよいことにする。更に、以下ではある一文字のみを担当としたシステムについて述べる。

まず、特徴抽出部は入力パターンの画素特徴を抽出するために、画素毎にプロセッサを割り当てる。また、同時に担当した文字で、予め登録している基準パターンの画素特徴を抽出させてもよい。これは、小規模の並列計算機などで複数の文字を担当しなければならない場合に、辞書の肥大化を防ぐ手段の一つである。続いて画素特徴の一次変換を行い、基準パターンのそれとのDPマッ

ングにより、両パターン間の画素同士の対応付けを行う。この一連の処理は、入力パターンの画素を担当したプロセッサ毎に並列して行う。結果はこれらを統括する画素統合プロセッサに対してへの投票制とし、基本として対応付けた画素のマッチング距離の小さいものを優先とする（これを最小距離優先による投票制とする）。対応付けを破棄されたプロセッサは、再度別の画素への対応付けを行う事を、画素統合プロセッサより報告される。

最後に、全ての画素の対応付けが終了したとき、対応付けの距離値の総和を全ての文字を統括する文字統合プロセッサに渡す。文字統合プロセッサは全文字から送られてくる距離値をもとに、最小距離値から順に候補として挙げる。

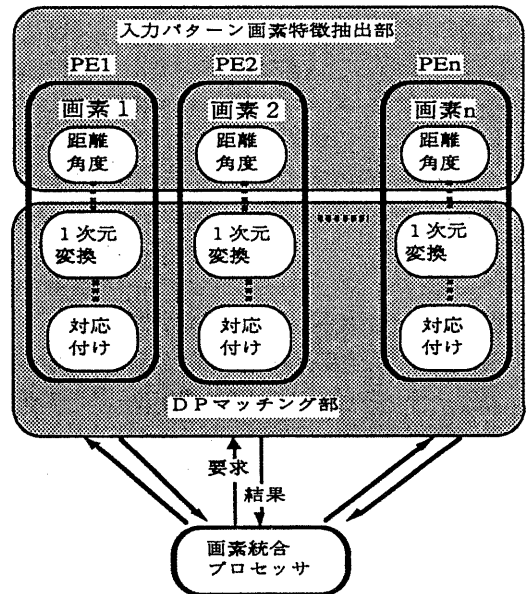


図5. 認識システムの構成図（一文字分）

4. SIMDマシンによる実現

今回、使用したSIMDマシン、LISCAR (Line-Scannable Cellular ARray processor) は、NTTのヒューマンインターフェース研究所で開発された小型並列プロセッサで、そのハードウェアと基本性能について簡単に述べる。

(1) LISCARについて[8]

・ハードウェア

LISCARは、プロセッサアレイをAAP2-L S

I (Adaptive Array Processor LSI) 4個で構成したアレイサイズが 16×16 のビットシリアル型2次元SIMDプロセッサである。PE (プロセッサエレメント) 間の接続は、段違い型のトラス接続構造となっており、通常の2次元データ処理の他に、ビットライン単位の並列処理 (線順次処理) を行うことが可能となっている。主記憶としては、各PEごとに1対1で接続された256ビット幅のアレイデータメモリ (ADM) を持つ。データメモリ (DM) はホスト装置とのバッファメモリを兼ねた16ビットワード演算用に、ADMはラインと呼ぶ256ビットを1ワードとする並列演算用に用いる。マイクロプログラムメモリ (PM) から出すインストラクションコードは、逐次演算および並列演算と各メモリの参照アドレスを制御する。

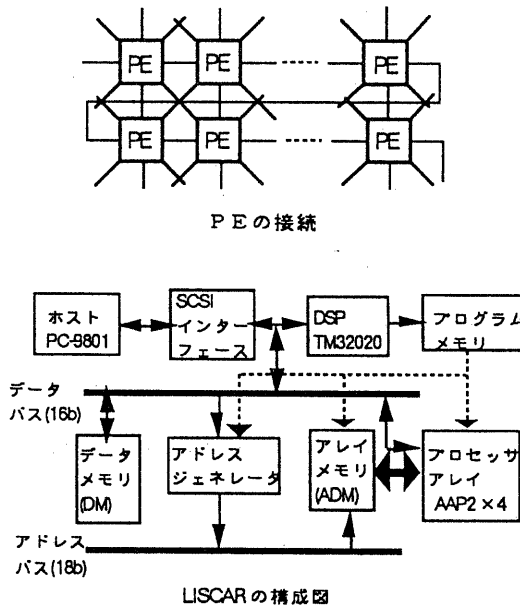


図6. PEの接続図とLISCARの構成図

・基本性能

マシンサイクルは143ns、処理速度としては500MOPSとなる。特に2値画像の処理は、 512×512 画素に対して、回転や拡大などで数~10数m秒という性能を誇る。

・開発環境

PC-9801をホスト装置として、SCSIを介して実行プログラムの転送や、LISCAR内の各メモリ

の内容を読み取りを行う。プログラムの開発は、パソコン上で専用のアセンブラを用いて行い、必要ならアセンブラソースレベルのデバッガも用意されていて、ステップ実行やレジスタ、変数の内容なども随時参照可能である。ただ、現段階ではLISCARプログラムの実行中に、ホストとのデータのやり取りができないので、処理の寸断化を招いている。

(2) 特徴抽出法

画素特徴の抽出は画像を直接扱ってもよいが、座標値より求める方が処理し易いため、線順次処理での方法について述べる。

入力される画像データは2値化されたものとし、256ドットづつ畳み込んでアレイメモリに格納しておく。まず、画素の座標値を得るため、画像データの大きさ分の順列Nを用意し、画素に対応した順列の値を以下のようにPに取り込む。このとき、プロセッサは横一列に並んでおり、縦方向にそれぞれ対応したアレイメモリをアクセスすることが可能となっている。

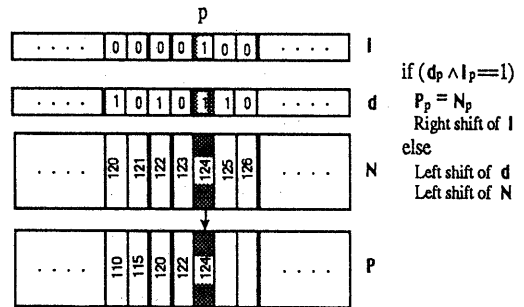


図7. 座標値の取得

続いて、X座標とY座標を求め、基準となる画素の座標をBYとして他の画素との距離と角度を求める。このとき、それぞれの角度と距離は同時期に求まる。なお、一次元変換はLと θ を合わせたラインを用いてソーティングを行えばよい。ソーティングは、各PEの担当する数値を他のPEの数値と比較して求めた勝ち数を基に、配置換えを行うことによる方法をとる。この方法は、ソーティングする要素数nがPEの個数を越えないときには、 $O(n)$ のオーダーで処理可能である。

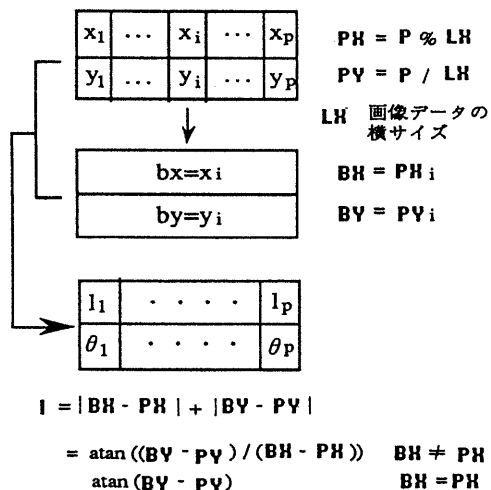


図 8. LISCARによる画像特徴抽出

(3) DPマッチング

DPマッチングは、基本的には逐次処理であるが、以下の図に示した順序で距離値を求めていくと、同数の箇所は並列化が可能となる。

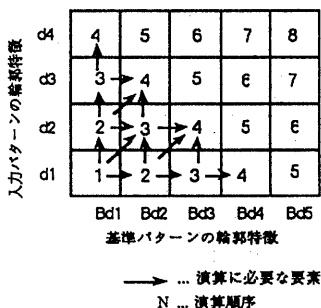


図 9. DPマッチングの並列化

そこで、横軸方向にプロセッサを割り当て、各箇所距離値を段違いにアレイメモリ上に格納することでDPマッチングの並列化を図った。通常で $O(n*m)$ (ここで、n...基準パターンの特徴数、m...入力パターンの特徴数とする) のオーダーかかるところが、この手法では、 $O(n+m)$ で済む。ただし、使用するメモリは、下図に示すとおり、 $(n+1)*(m+1)$ と通常よりも倍以上必要となる。また、縦軸方向の入力パターンの特徴数はメモリの許す限り構わないが、横軸方向の基準パターンの特徴数がプロセッサの個数を越えた場合は考慮に入れていない。以下に上

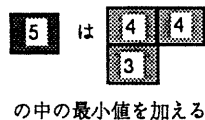
記の図10の計算順序に対応した距離値をアレイメモリ上に格納した図と処理アルゴリズムを示す。

255	0	0	0	0	8
255	0	0	0	7	7
255	0	0	ε	ε	ε
255	0	5	5	5	5
255	4	4	4	4	255
255	3	3	3	255	0
255	2	2	255	0	0
255	1	255	0	0	0
255	255	0	0	0	0

PE1 PE2 PE3 PE4 PE PE6

初期値は対応パターン間の距離値

図 10. 距離値の格納状態



```

while (t ≤ n+m) {
  RF1p = AMp(t);
  RF1p = Min(RF1p, RF2p);
  RF2p = AMp-1(t); ... depend on AM(t) Right Shift
  RF1p = Min(RF1p, RF2p);
  t = t + 1;
  AMp(t) += RF1p;
}

```

AMp(t) ... p番目のPEのアレイメモリの内容
 RF1p, RF2p ... p番目のPEのレジスタ
 Min(a,b) ... a, bの内での最小値
 n ... 基準パターンの特徴数
 m ... 入力パターンの特徴数

図 11. DPマッチングアルゴリズム

5. シミュレーションによる結果

現時点ではワークステーション上で抽出された輪郭特徴をDPマッチングにより、対応画素を求めるシミュレーションを行っている。その実験方法と結果を以下に示す。

・実験方法

実験は、Sun-SPARC STATION2上でを行い、X-Window

を使用して文字データの作成、輪郭特徴の抽出、DPマッチングのプログラムをそれぞれ別のウインドウ上で実行し、イベントドリブン方式で処理を行っている。本来なら手書きの文字データをイメージスキャナなどで取り込んだ画像データを二値化したものを、文字データとして使用するのだが、スキャナからのデータ転送やスキャナの性能などの問題点と、今回はシミュレーション実験であることなどから、マウスでウインドウ上に直接描くことにした。

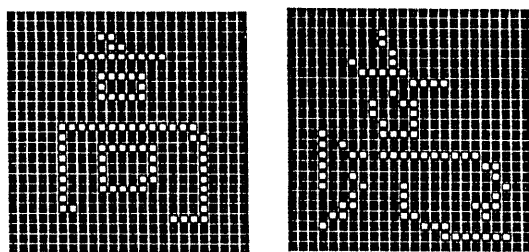
今回の実験の目的は、汎用計算機で処理した場合、どの程度の負荷がかかっているか、ということと、輪郭特徴の有効性の検証とする。しかし、このシミュレーションでは、次のような取り決めと制約を設けているので、認識システム全体をシミュレートしているわけではない。

- ・画素特徴の要素である画素間の角度は、スクリーン上での絶対角度を用いた。
- ・また画素間の距離は、同一角度の画素内で最長距離のものを1とし、他のものはこれに対する割合で表す。
- ・マッチング距離は画素間の距離と角度のみを用いる。輪郭特徴 dot '1(dn) と dot '2(dn) 間の距離は以下の式で求める。

$$d = | |l_1 - l_2| \times \|r_1 - r_2\|$$

- ・マッチングにおける文字の回転に対する特徴ずらしは行っていない。
- ・基準パターンへの入力パターンの対応付けは、基準画素と全画素との距離値を求めた後、8近傍画素の距離値の和をとり、最小となる画素とする。
- ・最終結果としては、基準パターンと入力パターンの距離値が求まる。

以上の条件の下で数種の文字間で実験を行った結果を示す。用いた基本パターンと入力パターン（比較用）の例を示す。



基準パターン

入力パターン

図12. 使用パターン例

以下に、求められた文字間の距離値の表を示す。

<画像データサイズ16×16>

基準パターン、比較パターン		距離値
・上0	上1	155.0
・川0	川1	217.0
・山0	山1	317.0
・山0	上1	305.0
・上0	川1	437.0
・川0	山1	55.0 (誤認識)

<画像データサイズ24×24>

・高0	高1	10.0
・亮0	亮1	33.0
・高0	亮1	38.0
・亮0	高1	38.0

<画像データサイズ32×32>

・佐0	佐1	14.0
・休0	休0	16.0
・佐0	休1	17.0
・休0	佐1	24.0

結果を見ると、扱う画像データサイズは大きいほど距離値に変動がなく全体的に良くなっている。これは、マッチングの距離値を求めている式による所が大きいのと、この輪郭特徴が、画像データサイズが大きいほど画素をより連続的に表現でき、威力を発揮できるためであろう。これは、認識のための特徴の有効性を示しており、輪郭特徴の手書き漢字認識に対する有効性と、プロセッサを大量に使用するほど処理効率のあがることが判った。今後、実際にLISCARでの特徴抽出とDPマッチングを行ってみたいの処理速度の評価を行う予定である。

6. まとめ

本研究では、従来から用いられている特徴の見直しを図り、処理が多少重くてもプロセッサ数の増加でカバー出来る、超並列処理向けの特徴として、画素特徴を結合した輪郭特徴を提案した。また、マッチングには線分の幅や歪みを吸収することを目的として、DPマッチングを用いた。更に、SIMDマシンによる並列DPマッチングの方式について考えてみた。

今後は、これらの方法をLISCARへインプリメントしてみたいの評価を行い、問題点などを検討する予定である。

・謝辞

LISCARの使用について御配慮いただき、使用方法について御指導頂いているNTTヒューマンインターフェース研究所の近藤利夫主任研究員に厚く感謝申し上げます。

<参考文献>

- [1] 須原 康次他：2進木計算機による並列パターン認識システムについて,信学技報,CPSY89-23,Vol.89, No.166,1989
- [2] 佐々木一陽他：超並列手書き漢字認識の研究,第43回情報処理学会全国大会,講演論文集(分冊2),2D-6,1991
- [3] 舟久保登著：視覚パターンの処理と認識,啓学出版,1990
- [4] 市原雅明他：知識ベースに基づく文字構造解析と認識について,信学技法,PRU-89-118,1989
- [5] Koji Suhara:Parallel Hand-Printed Character Recognition by Independent Feature Extraction,徳島大学工学研究科修士論文,1990
- [6] 山本淳晴他：並列型細線化アルゴリズムの一検討,電子情報通信学会春期全国大会論文集(分冊7)(情報・システム),1991
- [7] 塩野充他：DPを用いた伸縮整合法による常用手書き文字の認識の実験,信学技法,PRL80-22,1980
- [8] 多田俊吉他8：小型高並列プロセッサとその文字認識への応用,電子情報通信学会論文誌,D Vol. J71-D, No.8, pp.1546-1552,1988