ショートノート

文書画像からの個別文字切り出しと認識処理の並列化

金田 悠紀夫† 藤沢 邦昭†

日本語印刷文書画像からの高精度・高速な自動文字読取り技法の実現が要望されている．並列計算機システムを用いた並列計算を適用することにより読取り精度を高く保ちながら読取り時間を短縮する試みを行い良い結果を得た．処理手法の特徴は文書画像から文字を切り出すプロセス，切り出した文字の認識を行うプロセスをそれぞれ複数生成し，流れ作業的に文字切り出しと認識を行い，高い並列性を抽出している．主メモリ共有型の並列計算機システム（Symmetry S 81, 28 CPU (80386)）を用いて単一 CPU の場合に比して 25 CPU で約 20 倍の速度向上を実現した．

Parallelization of Character Extraction and Recognition from Japanese Printed Documents

Yukio Kaneda† and Kuniaki Fujisawa†

It is important to realize high speed and high precision Japanese character recognition from printed documents. By introducing parallel processing technique into extraction and recognition of characters from printed Japanese documents, we succeeded in speedup of overall processing. By using the shared memory type multiprocessor system (Symmetry S 81, 28 CPU (80386)), we realized about twenty times faster prosessing than by one CPU.

1. はじめに

既存の印刷文書画像から高精度・高速に文字読取りを行い電子ファイル化することが強く求められている．日本語文書においては字種が多いこと，複数字体や分離文字の存在，空白による区切りがないことなど文字の切り出しや認識処理が複雑となり処理時間を多く要することが欠点となっている．並列処理を導入することにより，高精度を保ちながら高速化する技法について研究を行い成果を得たので報告する．

2. 処理計算機システムの概要

すべての処理は主メモリ共有型の並列計算機 Symmetry S 81 (CPU : 80386, 28 CPU, 全体性能約 140 MIPS, 主メモリ 240 M バイト）で行った．プログラムはすべてC言語で記述されており，並列化はオペレーティングシステム（DYNIX）のもつパラレル・プログラミング・ライブラリの関数を用いて行っている．プロセス生成やプロセス間の同期・通信を行う関数の利用や共有メモリの使用が可能となっている．可動プロセス数が CPU 数より少ない場合には全プロセスが並列実行される．

† 神戸大学工学部
Faculty of Engineering, Kobe University

3. 処理の概要

文書画像は A 4 サイズ横書きで 300 dpi のイメージスキャナで入力され，2値化されている．図表の部分はないものとし，傾きもないとする．読取り対象文字は JIS 第一水準の日本語文字と英数字あわせて 3303 文字とした．文書画像に傾きがないとしているので文字列行の切り出しと文字高さの推定は簡単な処理で可能で単一 CPU で A 4 用紙 1 ページ 2 秒程度である．したがって文字列行からの個別文字切り出しと認識が主な処理となる．

3.1 個別文字の切り出し

日本語文書には分離文字の存在や英数字などの半角文字が含まれていること，接触文字も存在することなどにより文字切り出しを困難にしている．

本研究では非接触文字優先切り出し法を用いている[2]．これは文字列中で互いに接触していない切り出し可能な文字を優先して切り出し，未切り出し文字はすでに切り出されている文字の情報を用いて切り出していく手法である．まず，文字列を縦方向にスキャン

し，白画素のみで黒画素が存在せず区切れる部分を見つける．

文字列の高さから文字ピッチを推定し，全角の非接触・非分離文字，全角の句読点，全角の分離文字，全角の句読点，半角の文字を切り出していく（**図1**）．

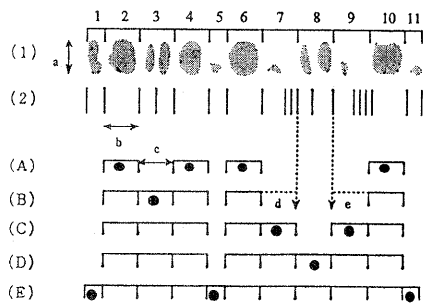
隣接文字と接触した文字は平均の文字ピッチと文字領域の縦方向の黒画素数のヒストグラムの谷から境を推定して切り出しを行う．文書中に含まれる英数字は半分程度のピッチのため連続した英数字列を位置だけで切り出すことはできない．接触文字も誤って切り出している可能性がある．そこで1文字切り出すごとに認識処理を行い認識結果の第一候補の類似度が一定値以下の場合は誤切り出しの可能性があるとして再切り出し処理を行っている．

3.2 個別文字認識処理

文字認識の手法として方向線素特徴量を用いた方式を採用した[1]．本手法では，

（1） 文字画像を 64×64 画素の大きさに正規化し Hilditch のアルゴリズムで細線化し，その後で太め処理をする（**図2**）．

（2） 文字画像を 8×8 の格子状に分割し，**図3**に示すように縦横2区間ずつ4区画を1小領域とし，各小領域を半分ずつ重なるようにする．細線化後太め処理された文字線画像の線上の各点での線方向（8方向）を求め，各小



（●は新たに切り出された文字を表す）

図1 基本切り出し処理
Fig. 1 Steps of character extraction from printed documents.

（a）正規化画像 （b）細線化画像 （c）太め処理をされた画像

図2 文字画像の細線化
Fig. 2 Thinning of binarized character images.

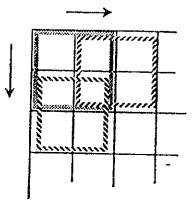


図3 7×7 の小領域
Fig. 3 7×7 size small region for calculating characteristic vector of a character image.

領域ごと，各方向ごと集計し，7×7×8＝392元の特徴ベクトルとしている．

（3） 特徴ベクトルを辞書に登録された各文字の特徴ベクトルと比較し類似度の高いものから候補文字としていく．分類は大分類と細分類の2段階で行われる．類似度の計算には次式を用いた（X，Yは特徴ベクトル）．

$$S(X,Y)=\frac{(X,Y)}{\|X\|\cdot\|Y\|}$$

ただし，$(X,Y)=X^{T}\cdot Y$，$\|X\|=(X,X)$

4. 並 列 化

処理の並列化に関しては，切り出し処理の並列化，認識処理の並列化，切り出し処理と認識処理の統合並列化などが考えられる．本研究では文字の切り出しを担当する切り出しプロセス群と認識を担当する認識プロセス群を設けて並列処理をした．認識プロセスは複数個で認識プロセスグループを作り，単一文字の認識を協力して実行し，認識処理の並列化を実現している．

4.1 文字切り出しの並列化

単一文字の切り出し処理は計算量が認識処理に比して1/10程度の約0.1秒と少ないので1文字切り出しは1プロセスで行うこととした．切り出しプロセスは文字列から文字を切り出し認識プロセスグループに文字画像を送る．文字が認識されると結果は切り出しプロセスに伝えられる．その結果を元にして文字列から次の文字が切り出され認識プロセスグループに送られる．

実際の処理は**図4**に示すように複数の切り出しプロセスと複数の認識プロセスグループを生成して実行している．未処理文字列は長いものから順に1本のリストにつないで管理しており，各切り出しプロセスは，排他制御によりこのリストにアクセスし，複数の文字列を取り出し，順番に各文字列から一文字ずつ切り出

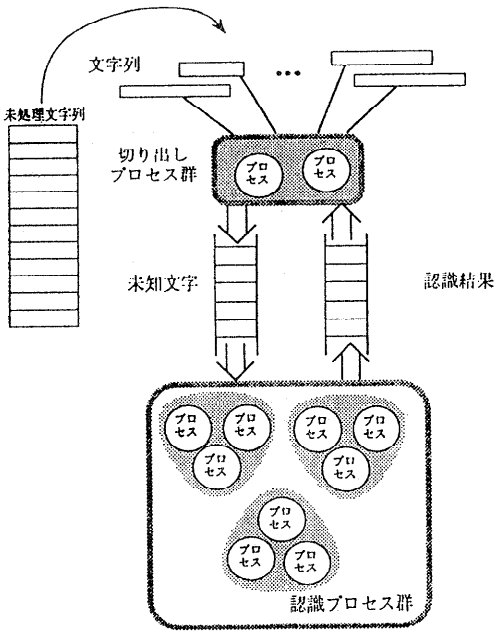


図 4 切り出しプロセス群と認識プロセス群
Fig. 4 Group of character extraction processes and recognition processes.

して共通の未知文字バッファに格納する．認識プロセスグループは次々と未知文字バッファから文字画像を取り出し認識を行い，切り出しプロセスに結果を伝える．結果を受けると切り出しプロセスはその文字が存在していた文字列から次の文字を切り出して未知文字バッファに格納することができるようになる．

4.2 文字認識処理の並列化

Hilditchの細線化アルゴリズムは2値画像を左上から右下へとラスタ（行）ごとに走査しながら特定の条件を満たす黒画素を消去（白画素化）する操作の繰返しである．この処理では注目黒画素が消去可能か否かは注目画素の8近傍画素の状態によってきまる．繰返し操作で消去できる黒画素がなくなったとき収束し処理は終了する．したがって複数のプロセスで左上から右下と処理を進めるという順に従いながら，上下の行が処理中でないという制約の基で（3行おきに）並列処理を進めることができる．

特徴ベクトルの計算は各小領域について並列にできるが計算量はあまり多くない．特徴ベクトルを用いた辞書の文字との類似度の計算はまず辞書中の文字を大分類用カテゴリに分割し，各カテゴリの特徴ベクトル（392元の特徴ベクトルを49元に集約したもの）と認識対象文字の大分類用特徴ベクトル（同様に49元に集約）との類似度計算を行うが計算はカテゴリごとに独立なので複数のプロセスで並列計算できる．類似度の高い順に複数のカテゴリを選び，そのカテゴリに属する文字との比較を行う細分類（392元の特徴ベクトルを用いる）もほぼ同様に並列計算できる．大分類，細分類とも結果を類似度順にソートする部分が直列処理となる．

以上述べた1文字に対する細線化，特徴ベクトル抽出，大分類，細分類の処理はこの順でシリアルに行われるが細線化と大分類，細分類の処理は事前に指定された数のプロセスが認識プロセスグループを構成し，排他制御を行いながら，それぞれ上記並列化手法に従って並列に行っている．これらのプロセスの生成は初期化時に行われ，その後の実行制御はDYNIXオペレーティングシステムによって行われており，管理用のマスタプロセスは存在しない．

5. 性能の評価

性能評価のため複数の印刷文書をイメージスキャナを用いて入力し，認識処理を行い性能評価を行った．

5.1 個別文字認識における並列処理

入力された文書画像から切り出した561文字を認識するのに要した処理時間を**表1**に示す．認識プロセスグループ内のプロセス数が5以上では並列効果が鈍ることが判明した．これは細線化処理の画像が64×64で小さいこと，収束が速いこと，細線化以外の部分でも並列処理の粒度が小さく頻繁にプロセス間の同期をとる必要があり並列効果が上がらなくなるためと考えられる．

5.2 切り出し処理と認識処理の統合並列化

切り出し処理と認識処理を組み合わせた場合の並列効果を測定した．処理した画像は1,698文字を含む43

表 1 個別文字認識処理のプロセス数と処理時間
Table 1 Speedup ratio of character recognition by increasing processes in recognition group.

プロセス数（個）	1	2	3	4	5	6	7
処理時間（msec）	577,120	299580	205660	161300	136710	122150	109460
速度比（倍）	1	1.93	2.81	3.58	4.22	4.72	5.27

表 2 1プロセスの認識プロセスグループを複数用いたときの並列効果
Table 2 Speedup of recognition by increasing number of recognition processes.

切り出しプロセス1個

総プロセス数（個）	2	3	4	5	6	7	8	9
処理時間（msec）	1841370	905150	604800	457880	365020	305580	263180	231080
速度比（倍）	1.10	2.23	3.34	4.41	5.54	6.61	7.68	8.75

総プロセス数（個）	10	11	12	切り出しプロセス2個	11	12	13	14
処理時間（msec）	221290	219960	220340		208920	184980	171470	159910
速度比（倍）	9.13	9.19	9.17		9.67	10.93	11.79	12.64

総プロセス数（個）	15	16	17	18	19	20
処理時間（msec）	146490	139790	131820	127710	127090	123230
速度比（倍）	13.80	14.46	15.33	15.83	15.90	16.40

行の文字列からなる．個別文字認識プロセスグループのプロセス数を1とした場合の並列効果を**表2**に示す．

切り出しプロセスを1つとして，認識プロセス（認識処理は単一プロセスで行う）を増やしていくと10程度までは良い並列効果が得られるがそれ以上増やしても効果が少ないことがわかる．これは文字の認識速度に文字切り出し速度が追いつかなくなったためである．切り出しプロセスを2個に増やすと再び並列効果が上昇するのがわかる．

次に，文字認識プロセスグループ内のプロセスの数と並列効果との関係を調べるために，文字認識プロセスグループ内のプロセス数を2，3，4と変化させたときの処理時間の変化を**図5**に示す．図5を見ると認識グループ内のプロセス数を増やしても総プロセス数が同一ならほぼ同一の並列効果を得ている．これはたとえば認識グループに各1個ずつのプロセスを割り当て，9グループ作ったときと，各3プロセスずつ3グループ作ったときと認識処理を行うプロセス数はどちらも9個で変わらないためほぼ同一の認識処理速度となる．

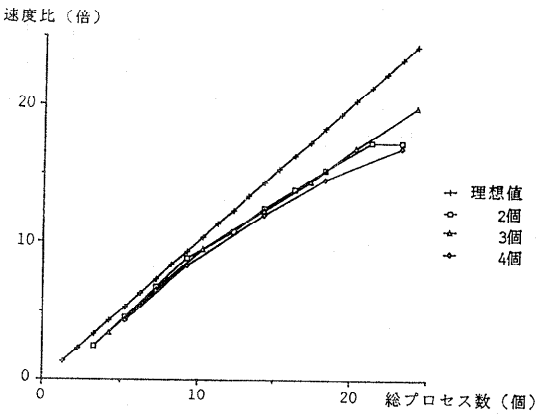


図 5 認識プロセスグループ内のプロセス数と並列効果
Fig. 5 Speedup by increasing processes.

しかし，認識グループ内のプロセス数に関係なくプロセス数を増やせば並列効果が上昇するわけではない．図5では認識グループ内のプロセスを4個にしたときには2個や3個のときに比べて若干効果が減少していることがわかる．プロセス数を5個以上にするとさらに並列効果が低下し，たとえば総プロセス数を17としたとき，認識プロセスグループのプロセス数を1としたときは速度比が約 15 倍であるのに対し認識グループ内プロセス数が5のときは 12 倍にとどまるという結果を得ている．これは表1に示すように個別文字認識処理の並列効果が理想値どおりには上昇しないからである．認識処理を行うプロセス数は変わらないが，各グループ内の並列効果が落ちているため全体の並列効果の上昇が鈍ってくるためである．

6. 結　　論

本方式では 25 CPU を用いて約 20 倍の高速化を実現し，A4 の文書画像を約 100 秒で読み取っている．最新の RISC 型マイクロプロセッサの性能は 80386 の 20 倍程度になっているので並列計算機の性能も比例して向上するとすると，5秒程度で読み取れることとなり，十分に実用性がでてくることが期待できる．今後の課題としては単語単位あるいは文単位での知識処理による切り出しミスや誤認識文字の検出と修正機能を付加することが考えられる．

謝辞 本研究の一部は文部省科学研究費補助金重点領域研究「超並列ハードウェア・アーキテクチャの研究」(課題番号 04235103) によっている.

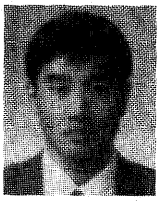
参 考 文 献

1) 孫寧ほか：方向線素特徴量を用いた高精度文字認識，信学論 (D-Ⅱ)，Vol. J 74, No. 3, pp. 330-339 (1991).
2) 秋山照雄ほか：非接触文字優先切り出しによる印刷物からの文字切り出し法，信学論(D)，Vol. J 67-D, No. 10, pp. 1194-1201 (1984).

(平成5年4月7日受付)
(平成5年9月8日採録)

金田悠紀夫 (正会員)

昭和 41 年神戸大学大学院工学研究科電気工学専攻修士課程修了．昭和 41 年電気試験所（現電子技術総合研究所）入所．電子計算機部において計算機システム研究に従事．昭和 51 年神戸大学工学部システム工学科（現情報知能工学科）教授．工学博士．並列処理，人工知能用言語，画像処理等の研究に従事．電子情報通信学会，IEEE，ソフトウェア科学会，人工知能学会各会員．



藤沢 邦昭

平成3年神戸大学工学部システム工学科卒業．平成5年同大学院修士課程修了．同年沖電気工業株式会社入社．