画素密度検出エージェントによる文字列の抽出と文字切り出し

岡村　健史郎* ・ ユジン ゴンザレス クルズ**・佐長　康久**・浜本　義彦**

Line and Character Segmentation by Pixel-Density Detect Agents

Kenshiro　OKAMURA* ，Eugene Gonzales CRUZ，Yasuhisa SACHO
and Yoshihiko Hamamoto **

Abstract

In handwritten document processing system, if the accuracy of character segmentation increases, the performance of the whole system will improve. A multi-agent based method using pexel-density for character segmentation has been presented. In this method, the center position of each character is estimated before character segmentation and the segmentation is carried out using these positions. In addition to this method, we introduce the line segmentation and improve the performance of character segmentation.

Key words: Document processing , character segmentation , Multi-Agent

1. はじめに

近年，文書に関する様々な情報処理機能を提供する文書理解システムは，大量の文章が一括して入力できることや，蓄積されてきた知識をデータベース化し再利用できることから，盛んに研究されてきた[1]，[2]．特に，一文字単位の手書き文字認識に対しては，高い認識率を達成する手法が存在し[3]，[4]，実用の段階に達成している．これらの手法を手書き文書理解システムに適応するためには，文書から文字を一文字ずつ分離する文字切り出し処理が必要になる．

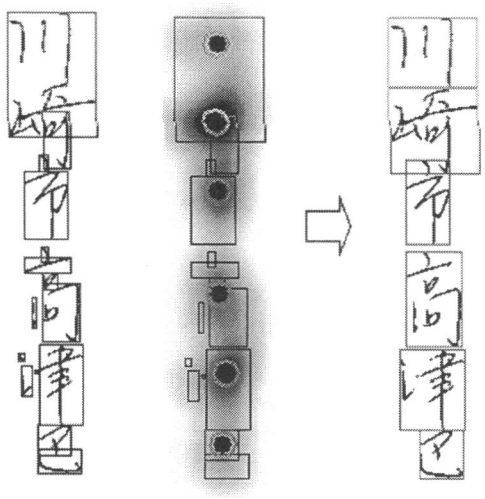
文書理解システムにおける文字切り出し処理に，郵便の宛名書き画像を対象とした手書き文書処理システムがある[5][6]．これらのシステムでは，まず画像から求めた黒画素の領域の角度や形状などに基づいて，文字を構成するストロークを切り出している．その後，これらの可能な全ての組み合わせを考え，一つ一つ文字認識し，住所が記録されている辞書との照合を行っている．この方法を用いると，組み合わせが莫大な数にのぼり計算コストが大きくなる，小単位の組み合わせを誤認識する，などの欠点があった．そのため，総合的手書き文書理解システムの実現のためには，正確な文字切り出し技術が必要となる．

そこで，筆者らは手書き文書からの文字切り出し方式として，文字切り出しに用いる図形的特徴に加えて画素密度を利用した文字切り出し手法を提案した[7]．この手法では，画素密度が局所的に高くなる部分を文字の存在する位置と仮定し，その文字位置情報に基づいて文字切り出しを行う．この手法の特徴として，文字の変形や隣接する文字どうしの接触などの変動に強いこと，文字列の方向に依存しないこと，などが挙げられる．しかし，画素密度を用いた文字切り出し手法は，文書画像中の文字サイズが大きく異なると，観測スケールを正しく決定することができず，文字の切り出しに失敗することがあった．

本論文ではこの問題を解決するために，画像中の黒画素の連結領域をもとに観測スケールの可能な範囲を求め，この値を用いて処理を行うことで切り出し率を高めた．更に，文字切り出しの結果をもとに文字列の抽出を行い，その結果から文字切り出しを修正するフィードバック処理を可能にした．その結果，手書き郵便宛名画像を対象とした実験において，縦書きの宛名データを対象にした時，文字列抽出率96.3%，文字切り出し率 89.4% ，横書きの宛名デー

＊情報工学科　＊＊山口大学工学部

2003 年 9 月 12 日受付



(a)連結成分　(b)画素密度と文字位置　(c)文字切り出し

図 1　画素密度による文字切り出し

タを対象にした時，文字列抽出率 96.0%，文字切り出し率 85.7% をそれぞれ得た．

本稿の構成は次の通りである．まず，2. において筆者らが文献[7]にて提案した文字切り出し手法の概要について述べる．3. において，新たに提案する文字列の抽出方法について述べる．4. において，計算機シミュレーションを行い，提案手法の有効性を検討する．5. において，まとめと今後の課題について述べる．

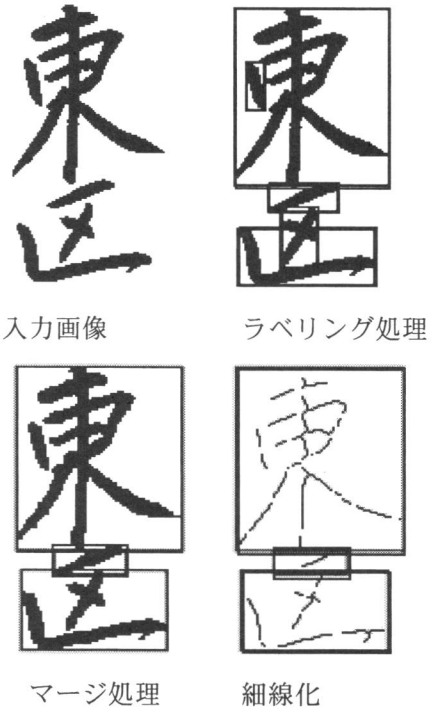
2.　画素密度検出による文字切り出し

ここでは筆者らが文献[7]において提案した画素密度検出エージェントによる文字切り出し手法について説明する．

2.1　文字切り出しの概要

画素密度検出による文字切り出しは，白黒 2 値画像において黒画素密度が局所的に高くなる位置に文字の中心が存在すると仮定する．この画素密度には，適当なスケールを持つガウスフィルタにより画像をぼかした値を用いる．このぼかした値を用いることにより，文字の変形，サイズなどの変動に対して処理結果がロバストになる．また，円形のフィルターであるため，文字列方向に依存しない処理が可能になる．

しかしながら事前にフィルターのスケールが決定できないため，従来からある逐次処理を用いて画素密度を用いると，フィルターのスケールを順に変えながら画像全面に畳み込みが必要になり，莫大な計算量になる．そこで，本手法では画素密度検出エージェントを用い，エージェントが文字の周辺から中心に向かって移動する軌跡部分にのみフィルター処理を行うようにすることで，大幅な時間短縮を可能にした．



入力画像　　ラベリング処理

マージ処理　　細線化

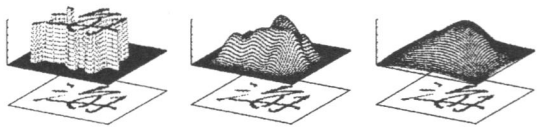
図 2　前処理例

文字切り出しにおける本手法の特徴は，文字の中心位置を推定した後で，個々の文字切り出しを行う点にある．これにより文字どうしが接触した場合でも，容易に文字を切断することが可能になった．

図 1 に画素密度を用いた切り出し例を示す．白黒 2 値画像に対し，8 連結成分を囲む矩形を求め，その後，画素密度から文字位置を推定する．この文字位置の情報から，連結成分をグループ化することにより，文字を切り出す．画素密度検出による文字切り出しは，前処理，文字位置検出，文字切り出しの三つの処理から構成される．以降では，文字切り出し方法の各部について述べる．

2.2　前処理

前処理として，雑音除去，ラベリング処理，マージ処理，細線化処理を行う．以降ではこれらの処理について述べる．図 2 に前処理例を示す．



(a)原画像　　(b) σ =4　　(c) σ = 8

図 3　観測スケールによる画素密度の違い

1.　雑音除去

入力 2 値画像に対し，3×3 の論理マスクにより孤立画素の除去を行う．

2. ラベリング処理

ラベリング処理により，黒画素 8 連結領域を求め，各領域を囲む外接矩形を求める．この矩形を基本単位とする．

3. マージ処理

ラベリング処理で求めた矩形に対し，明らかに同じ文字を構成すると考えられる矩形同士を統合するマージ処理[8]を行う．

4. 細線化処理

筆記用具による文字の太さの違が画素密度に影響を及ぼさないよう細線化処理を行う．

2.3　文字位置検出

文字位置検出では，細線化した画像の画素密度に基づいて文字の位置を検出する．画素密度を，対象画像とガウスフィルタの畳み込みにより求める．ここで，対象画像を

$$f(x,y) \quad (0 \le x \le X-1, \quad 0 \le y \le Y-1) \qquad (1)$$

とすると，画像上の座標 (x,y) における画素密度 $G(x,y,\alpha)$ を次式で定義する．

$$G(x,y,\sigma) = \sum_{s=-x}^{X-x-1} \sum_{t=-y}^{Y-y-1} f(x+s,y+t) \cdot g(s,t,\sigma)$$
$$g(s,t,\sigma) = \frac{1}{2\pi\ \sigma^2} \cdot \exp\left(\frac{-(s^2+t^2)}{2\sigma^2}\right) \qquad (2)$$

ここで，σ^2 はガウスフィルタ $g(s,t,\alpha)$ における分散を表し，σ を観測スケールと呼ぶ．画素密度は観測スケール σ の値により異なる．図 3 に観測スケール σ による画素密度の違いを表す．文字は一般に中心付近が複雑となり，画素が集中する構造を持つと考えられる.この例を図 3(c)に示す．本手法では，画素密度が局所的に高くなる位置を文字の存在する位置と仮定し，一つの文字から得られる画素密度を単峰状の図形としてとらえることで，その頂点を文字の中心位置として求める.そのためには，文字サイズに合わせて σ の値を適応的に決定する必要がある．

適応的処理を実現するための処理形態にエージェントシステムがある．エージェントシステムは，自律的に動作する複数のプロセスが協調動作を行うことにより，詳細で信頼性の高い解析を行うことができるという特徴を持つ．そのため，画像認識や音声認識などの分野で応用されている．筆者らは，このエージェントを導入することにより，画像に対して適応的かつ自動的に σ の値を決定する手法を提案した．以降では，文字位置検出を可能にする画素密度検出エージェントと管理エージェントについて述べる．

2.3.1　画素密度検出エージェントの配置

画素密度検出エージェントは，文字位置の検出を目的として，画素密度に基づいて自律的に動作する一つのプロセスである．以降，画素密度検出エージェントを単に検出エージェントと呼ぶ．この検出エージェントの文字位置検出の概要を図 4 に示す．

エージェントの配置

2. 2のラベリング処理で求めた矩形に対して，矩形の各辺と矩形の中央の計 5 箇所を基本に検出エージェントを配置する.ただし，形状に応じて配置数と配置位置を変化させる[7].

文字位置探索

画像中に複数配置した検出エージェントは，観測スケール σ を用いて自律的に文字位置の探索を行う．各検出エージェントは，画素密度が高くなる方向へ移動を繰り返し，密度が局所的に高くな

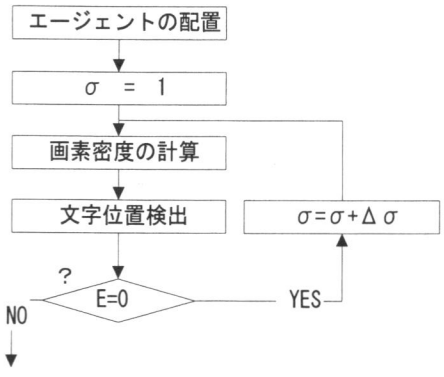
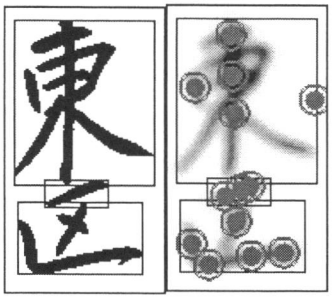
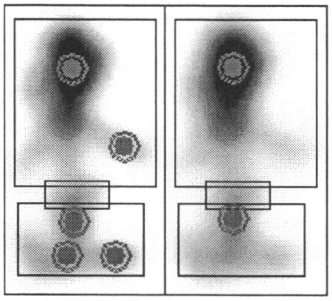


図 4 文字位置検出の流れ



(a)原画像　(b) σ =4, E=13



(c) σ =8, E=5　(d) σ =10, E=0

図5 文字位置検出過程

る画素を見つける．この位置を画素密度の頂点と呼ぶ．この時、検出エージェントは，移動する画素の周囲に対してのみ密度を計算する．そのため，画像全体に対して密度を求めるより，はるかに少ない計算量で文字位置を検出できる．

2.3.2 評価関数による観測スケールσの評価

画素密度の頂点を検出後，以下の二つの仮定を用いて観測スケールσの評価を行う．

仮定1

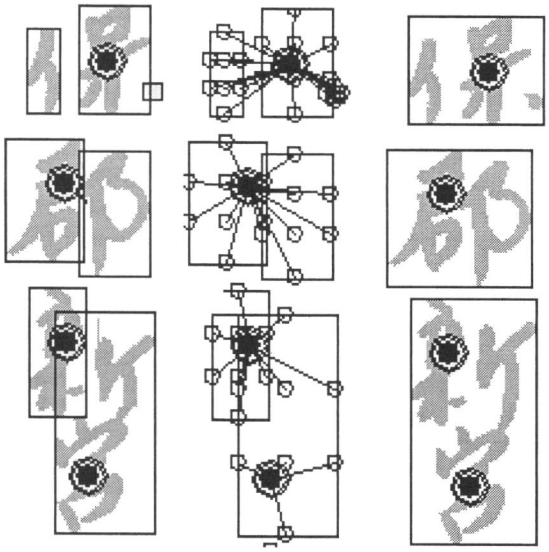
文字は一つ以上の8連結矩形から構成され，一つの矩形内に画素密度の頂点を表す検出点は複数存在しない．

仮定2

頂点付近には適当な大きさの文字領域が広がるため，検出点どうしは近くに存在しない．

以上の仮定を満足するか調べ，満足しなければ観測スケールを大きくしながら，次のようにエージェントが協調動作を行う．

1. 頂点を見つけると、画像中の検出エージェントが周囲の検出エージェントと通信しながら二つの仮定の評価を行い，評価結果を管理エージェントに報告する．
2. 管理エージェントは検出エージェントからの報告をもとに画像全体に対して二つの仮定が満足されているか評価関数Eの値を求める[7]．
3. E=0 の場合には，この時の検出エージェントの位置を文字中心位置とする．
4. E≠0 の場合には，管理エージェントは観測スケールをΔσだけ大きくするよう全ての検出エージェントに連絡する．



(a)検出点と矩形　(b)グループ化　(c)抽出文字

図6 文字位置を使った矩形のグループ化

図5に各σの値と，この時の評価値Eの値を示す．図5の各矩形は，2.2のラベリング処理で求めた矩形を示している．図5(b)～(d)に，各観測スケールσにおける画素密度と検出エージェントが求めた頂点を示している。原画像図5(a)に対して，検出エージェントは、σを1から始め徐々に大きくする．観測スケールσ=10の時，評価値Eが0になり，検出を終了する．図5(d)にある2点には全ての検出エージェントが集中している．この2点を最終的な文字の中心位置とする．

2.4 文字切り出し

文字切り出しでは，まず検出エージェントの多数決による矩形のグループ化を行い，文字を構成する矩形を決定する．次に，必要に応じて接触文字の切断処理を行う．以降では，各処理について述べる．

2.4.1 エージェントの多数決による矩形のグループ化

画像中に存在する矩形を文字位置検出部により求めた検出点に対応づけることより，矩形のグループ化を行う．矩形は，初期配置した検出エージェントが最も多く集まった検出点に所属すると決定する．次に，同一の検出点に所属する矩形同士を統合する．図 6 に画素密度を考慮した矩形のグループ化の例を示す．図 6(a)にエージェントが求めた文字中心位置を示す．図 6(b)に，初期配置したエージェントが文字中心位置まで移動した軌跡を示す．この軌跡から「保」を構成する二つの矩形は，一つの検出点に所属するため，一つの文字として統合された．一方，「新」と「宮」を構成する二つの矩形が一つの検出点に所属し，誤って一つのグループに統合されている．

2.4.2 接触文字の切断

図 6(c)「新宮」のように，グループ化処理で求めた矩形内に複数の検出点が存在する場合，複数の文字が存在すると考え、接触文字切断処理を行う．ここでは，検出点間の範囲で矩形の長辺方向への周辺分布を求め，この範囲で周辺分布が最小となる点を切断位置と決定する．ここで用いる周辺分布は黒画素を長辺方向の座標軸に単純投影したものである．図 7 にヒストグラムと切断結果を示す．この様に，文字同士が接触している場合でも，事前に文字数と文字中心位置が分かっているため容易に文字切断が出来るという特長がある．

3.　文字切り出し精度向上と文字列抽出

2．で述べた画素密度を用いた文字切り出し手法に加え，観測スケールの可能範囲を求めることで，切り出し率の向上を行う手法について述べる．更に，文字切り出し後，切り出しの結果から文字列を抽出し，この文字列を用いて文字切り出しの修正を行うフィードバック処理について述べる．

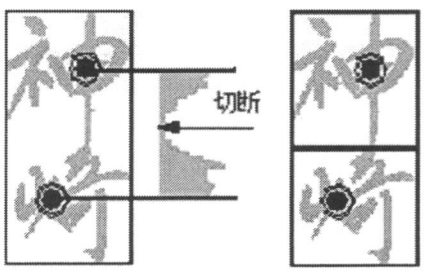
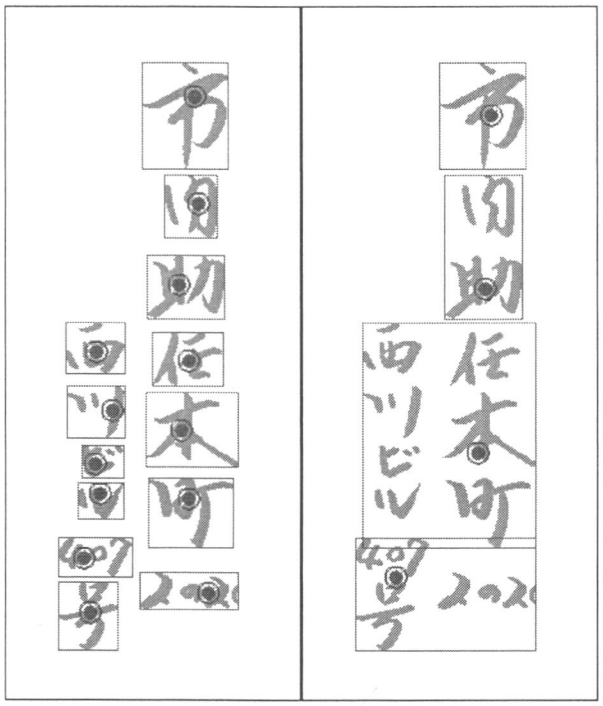


図 7　接触文字の切断例



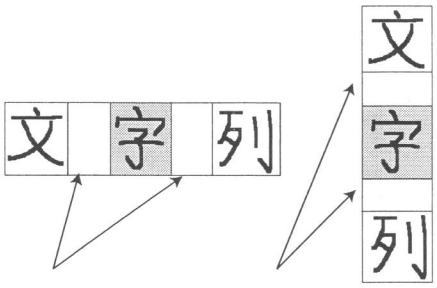
(a) σ =12　　(b) σ =30

図 8　文字位置検出失敗例

3.1　観測スケール可能範囲の設定

2．で述べた手法は，2．3．2における二つの仮定が満足されている場合には正しく文字位置を検出する．しかしながら文書画像中の文字サイズが大きく異なると，σ を変化させてもこの仮定が成り立たず，文字位置の検出に失敗し，文字切り出しに失敗する．この例を図 8 に示す．σ =12 において，正しい検出点が見つかっているにもかかわらず，小さい文字「ビ」と「ル」の検出点間距離が短いため仮定 2 を満足せず，$E = 0$ とはならず，エージェントは σ を大きくし続け，σ =30 になったところで評価関数 $E = 0$ となる．しかし，この観測スケールでは文字位置を正しく検出することができない．

そこで，観測スケールの値が適切な値になるように，あらかじめ観測スケールの可能範囲を決定する．前処理で求めた連結成分のサイズにより，文字のサイズを推定することができる．その値を利用し，観測スケールの最大値 σ_{th} を次式で定義する．



左右接続線対　　上下接続線対

図9　文字の接続線対

$$\sigma_{th} = \max_{i=1,\cdots,N} \{\frac{li}{4}\} \qquad (3)$$

ここで，Nは8連結矩形の数，liは矩形の長辺の長さを表す．観測スケールσを求める適応的な2．3．2の処理において，評価関数Eが0になるようにσを増やしながら画素密度を求め，文字位置検出を行う．このとき，$\sigma \geqq \sigma_{th}$であれば，$E=0$を満足していなくても，σ_{th}を観測スケールとし文字位置を検出する．

3.2　文字列抽出

文字列において，一つの文字に隣接可能な文字は高々二つである．また，文字は必ず同方向の文字列に所属すると考える．この二つの仮定を用いて文字列による文字のグループ化を行う．

文字列において隣接する文字を囲む矩形の角を接続する線を接続線対と呼ぶ．この，文字接続線対を図9に示す．まず，注目する文字iに対し，その長さが最小となるような上下左右の接続線対を求める．これらの接続線対の中で最も短いものを文字iの接続線対とし，$li1$で表す．また，接続線対の中で2番目に短いものは次式を満たした場合文字iの接続線対とし，$li2$で表す．

$$|li2| < \alpha \times (\sqrt{(xri - xli)^2} + \sqrt{(yri - yli)^2}) \qquad (4)$$

ここで，$(xli, yli), (xri, yri)$はそれぞれ文字iを囲む矩形の左上，右下座標を表す．また，σは文字サイズに対する接続線対の長さのパラメータであり，本研究ではσ=1.5を用いる．図10 (c)に「島」に対する$li1$，$li2$を表す．この処理を図10(d)のようにすべての文字に対し行う．

次に，誤った接続線対の削除を行う．図10 (d)の「助」，「町」などに示すように，縦方向と横方向の2種類の接続線対で接続されている文字の接続線対において，一組は誤っていると考えられる．そこで次のようにして，誤った接続線対を削除する．2種類の接続線対を持つ文字に対して，接続されている全ての文字の縦接続線対，横接続線対を数える．縦接続，横接続のうち多い方をその文字の方向と決定し，その方向ではない接続線対を削除する．誤った接続線対の削除処理の例を図10 (e)に示す．ここで，「町」という文字は横と縦に接続されている．この場合，「町」に接続されている文字「本」，「市」の状態を検討する．「本」は横接続線対を0組，縦接続線対を2組持つ．「市」は横接続線対を2組，縦接続線対を1組持つ．合計すると縦接続線対が3組，横接続線が2組となり，縦の方が多いため，「町」の横接続線対が誤りであると判断し削除する．

図10(f)に図10(d)の矛盾接続線対の削除処理を行った結果を示す．接続線対で接続されているすべての文字は一つの文字列を構成すると考える．図10(f)の場合，三つの文字列が生成される．

3.3　文字列の修正

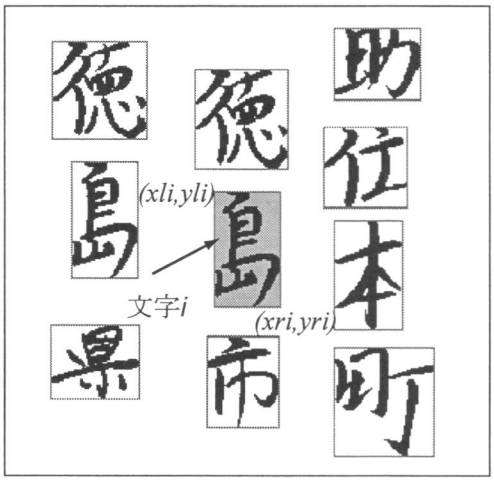
まず，文字列の方向を決定する．文字列の方向は，文字列内の文字の縦接続線数，横接続線数で多い方をその文字列の方向とする．次に，文字列の方向と異なる接続線対を持つ文字が存在する場合，文字列と同方向の接続線対に修正する．

更に，図11(a)のように文字列の先頭の文字が，隣の文字列の文字に接続される誤りが多くある．そのため，文字列内の各文字の接続線対の長さに基づいて接続線対の修正を行う．ここでは，各文字の間隔がほぼ一定であると仮定し，長さの平均に比べ極端に長い接続線対を削除する．以降では，それらの手順について述べる．

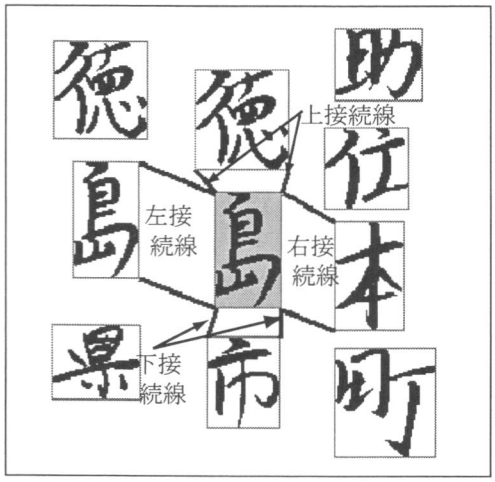
1. 文字列rの接続線対の平均長$|\hat{l}(r)|$を求める．
2. 文字列rにある文字iの接続線対の長さ$|li|$が次式を満足すれば，その接続線対を削除する．

$$|li| > \beta \times |\hat{l}(r)| \qquad (5)$$

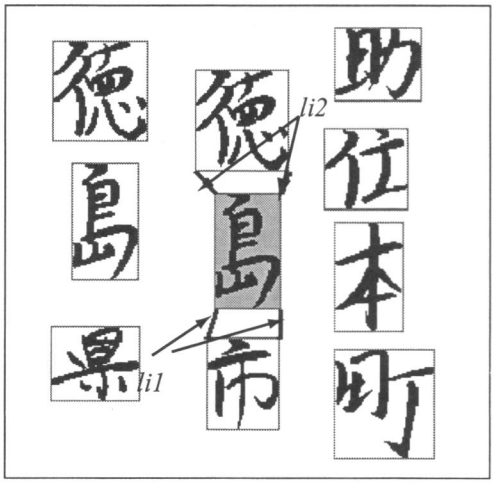
ここで，βは文字列内における接続線対の長さに対する変動パラメータであり，本研究ではβ=4とした．図11(a)に(5)式を満足する接続線対を示す．また，接続線対を削除すると，図11(b)にあるように文



(a)データ

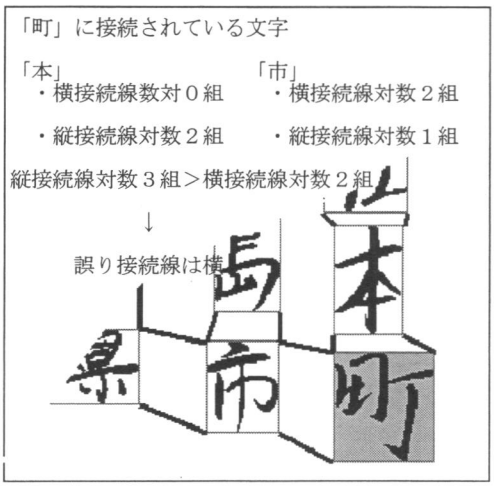


(b)「島」に対する４種類の接続線データ

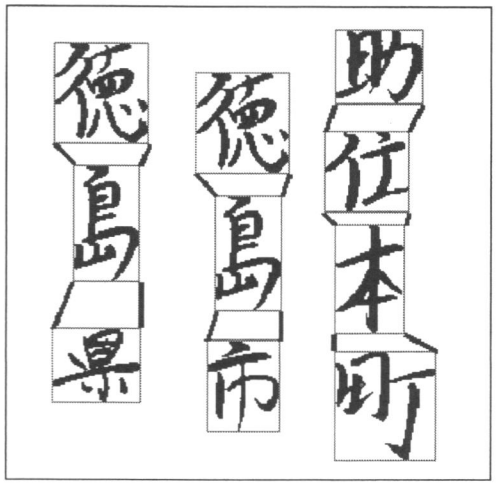


(c)「島」の接続線対

(d)全文字に対する接続線

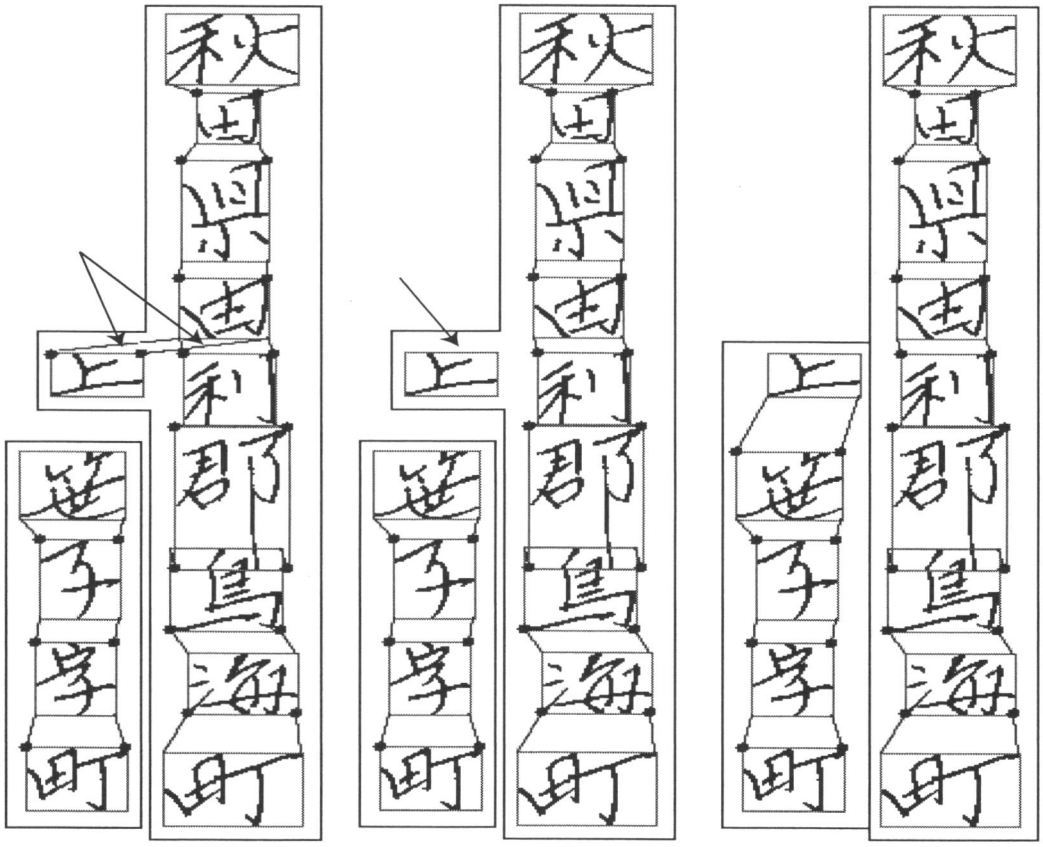


(e)矛盾する接続線対の削除



(f)接続線による文字列生成

図10　文字接続線対による文字列の作成方法



(a)接続線対による文字列生成　(b)長い接続線対の削除　(c)孤立文字の属する文字列修正

図 11　文字列修正

字列に含まれない文字が発生する．文字列に接続されていない文字は他の文字列に所属すると考える．その文字 i に対して，次式を満足するような文字列 r があれば，図 11(c)の様に文字をその文字列に追加する．

$$|d(i,r)| < \beta \times |\hat{l}(r)| \qquad (6)$$

ここで，$|d(i,r)|$ は文字 i と文字列 r の距離を表す．孤立した文字に対し，式(6)を満足するような文字列が存在しない場合には，一文字からなる文字列を生成する．

3.4　文字列を用いた文字切り出しの修正

文字列内の接続線対に基づいて文字切り出しを修正するフィードバック処理を行う．図 12 に示すように，同じ辺に接続されている接続線対を持つ二つの文字は一つの文字を構成すると考える．そのとき，二つの文字を統合する．統合後にマージ処理を行い，再びその文字の接続線対を求め，矛盾する接続線対を調べる．

次に，複数の文字を一つの文字に統合する誤りを修正する．文字の外接矩形の横縦比は 1.5 の時に複数の文字を統合したと仮定し，文字の切断を試みる．

4. 計算機シミュレーション

提案手法有効性について検討するため，手書き郵便宛名文書を対象とした実験を行った．まず，実験に用いたデータについて述べる．次に，文字切り出しと文字列抽出の性能について検討する．最後に，提案手法の有効性について検討する．

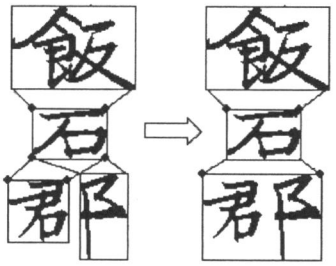


図 12　文字統合例

4.1 実験データ

計算機シミュレーションでは縦書き及び横書きの郵便宛名画像を対象に実験を行った．縦書きの画像は，旧郵政省郵政研究所の手書き漢字データベースIPTP CD-ROM2 に収録されている画像を用いた．IPTP CD-ROM2 は，郵便ハガキの宛名部分から収集した多重多様な記載品質，文字形状を有する手書き漢字画像データが 11128 枚収録されている．各画像は幅 45mm×高さ 105mm(720 ×1680 ピクセル)，256 階調の白黒濃淡画像である．実験では，サンプル番号 1-30，1001-1030，2001-2030，... ，9001-9030 の 300 枚の画像を使用した．これらの濃淡画像データを 2 値化し，縦横それぞれ 3 分の 1 サイズに縮小して実験を行った．

横書きのデータは，上白紙に多重多様な筆記具で書いた宛名住所を市販スキャナーで取り込み 2 値化した 300 枚の画像を用いた．宛名住所は全て縦書きと同じものを用い，画像が複数列から構成されるよう「大島商船高等専門学校教務係御中」という宛先を加えて，筆記者に書くよう指示した．この時，住所が何列になっても構わない旨を伝えた．筆記者は大島商船高等専門学校情報工学科平成 14 年度 4,5 年学生 80 名全員に依頼した．

各データは町域部と番地部（○丁目○番○号）からなる．従来，文献[7]，[8]などの認識実験では，町域部と番地部の文字サイズが大きく異なるため，町域部だけを対象に処理し，切り出し率を求めていた．実用化を考えた場合，この様な町域部と番地部を事前に区別することは不可能である．本研究では，何れの実験においてもこれらを区別せず一様に処理を行った．但し，文字切り出しにおいて，切り出し率は町域部に含まれる文字のみを対象に計算した．

4.2 実験概要

文字切り出し率を次式のように定義する．

$$\text{文字切り出し率}(\%) = \frac{\text{切り出し成功文字数}}{\text{全対象文字数}} \times 100$$

文字切り出しの実験では，

- 縦書きデータを対象にした実験
- 横書きデータを対象にした実験

の二つの実験を行った．この実験では，3．で提案した三つの処理

処理(A)　観測スケールの可能な範囲の設定
処理(B)　文字列による文字の統合
処理(C)　文字列による文字の切断

がそれぞれ切り出し率にどのように影響するかを調べるため，

- 文献[7]
- 処理(A)
- 処理(B)
- 処理(A)＋処理(B)
- 処理(A)＋処理(B)＋処理(C)

の 5 通りの実験を行った．三つの処理において，処理(B)と処理(C)が文字列を用いたフィードバック処理に相当する．

次に，文字切り出しの結果に基づいた文字列抽出手法の有効性について検討するため，実験を行った．文字列抽出率を次式のように定義した．

$$\text{文字列抽出率}(\%) = \frac{\text{文字列抽出成功数}}{\text{全対象文字列数}} \times 100$$

4.3 考察

文字切り出し率の観点から提案手法を従来手法と比較する．次に，本研究で新たに提案した文字列の抽出手法の有効性について検討する．

4.3.1 文字切り出し

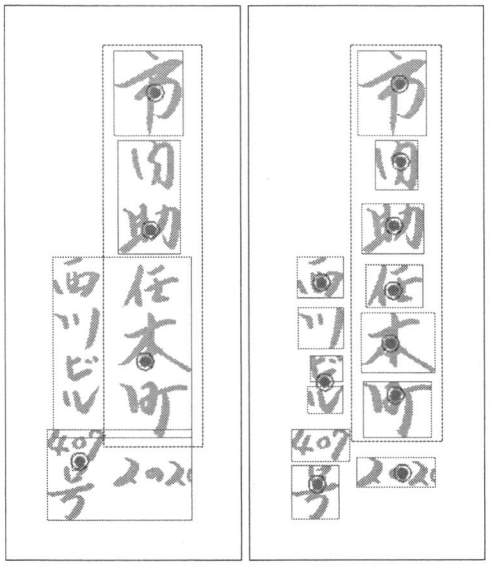
縦書きデータを対象に得られた結果を表 1 に示した．また，横書きのデータを対象に得られた結果を表 2 に示した．まず，従来手法に追加した処理(A)について検討する．処理(A)とは，文字位置検出に用いる観測スケールの可能な範囲を設定する処理である．文献[7]に処理(A)を追加すると，縦書きのデータに対し 4.1%，横書きのデータに対しては 50.1%も上昇している．処理(A)は，画像中にある文

表 1　縦書きデータを対象にした文字切り出し率(%)

文献[7]	(A)	(B)	(A)+(B)	(A)+(B)+(C)
83.2	87,3	84.2	88.4	89.4

表 2　横書きデータを対象にした文字切り出し率(%)

文献[7]	(A)	(B)	(A)+(B)	(A)+(B)+(C)
30.9	81.0	31.0	81.1	85.7



(a)従来手法　　(b)提案手法

図 13　縦書きに対する文字抽出例

字のサイズが大きく異なる場合でも，観測スケールが正しく決定できるように追加した処理である．従って，データの文字サイズが大きく異なる場合に効果がある．例えば，文献[7]の処理結果図 13(a)では，町域部と番地部の文字サイズが大きく違うため文字切り出しに誤りが多くある．一方，処理(A)を追加した図 13(b)では，町域部の文字が正しく切り出されている．また，従来手法では横書きデータの文字間の間隔が小さすぎるため，2．3．2にある観測スケールを決定するための仮定を満足せず，図 14(a)の様に複数の文字を一つの文字に切り出す誤りが発生する．本手法では，図 14(b)に示す様に，この失敗を防ぐことができた．一方，一つの文字を二つに分解してしまうケースも発生した．

次に処理(B)について検討する．処理(B)とは，文字列の接続線に基づいて文字同士の統合を行う処理である．実験結果から従来手法に処理(B)を追加すると，縦書きのデータに対し 1.0%向上し横書きのデータに対しては文字切り出し率がほとんど変わらなかった．漢字は辺と旁で構成されている場合が多い．そのため，横書きのデータに対して文字列の方向に文字が分割しやすく，文字切り出しの修正が難しくなる．

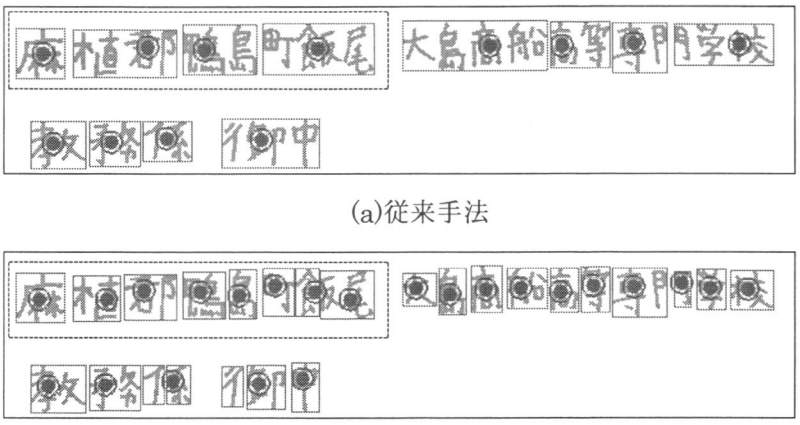
処理(A)と処理(B)を組み合わせることにより，縦書きのデータに対し 5.2%，横書きのデータに対しては 50.2%上回っている．これは，処理(A)の文字サイズの変動に対する効果と処理(B)の文字列による訂正効果が現れたためである．

次に処理(C)を検討するため，従来手法に処理(A)+(B)を追加したときの結果と処理(A)+(B)+(C)を追加したときの結果を比較する．処理(C)とは，複数の文字を誤って統合してしまったときの切り出しの修正を行う処理である．(A)+(B)+(C)の文字切り出し率は(A)+(B)の切り出し率に対し，縦書きのデータで 1.0%，横書きのデータで 4.6%，それぞれ向上している．

更に，処理(C)の有効性を詳しく検討するため，文字切り出しに失敗した文字を観察し，文字切り出しの失敗原因を

誤り 1：一文字が複数文字に分割される
誤り 2：複数文字が一文字に統合される
誤り 3：その他

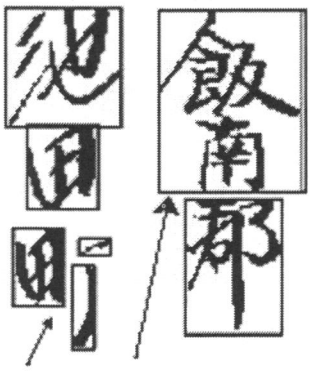
の 3 通りに分類した．表 3 に処理（C）の無い処理(A)+(B)の場合と，処理（C）のある場合における，各誤り個数を示した．また，図 15 には誤り 1 と誤



(a)従来手法

(b)提案手法

図 14　横きに対する文字抽出例



誤り 1　　誤り 2

図 15　文字切り出し失敗原因

表 3　切り出しに失敗した
原因とその文字個数

(A)+(B)			(A)+(B)+(C)		
誤り 1	誤り 2	誤り 3	誤り 1	誤り 2	誤り 1
96	463	72	315	136	72

表 4　文字列抽出結果

対象データ	抽出率(%)
横書き	96.3
縦書き	96.0

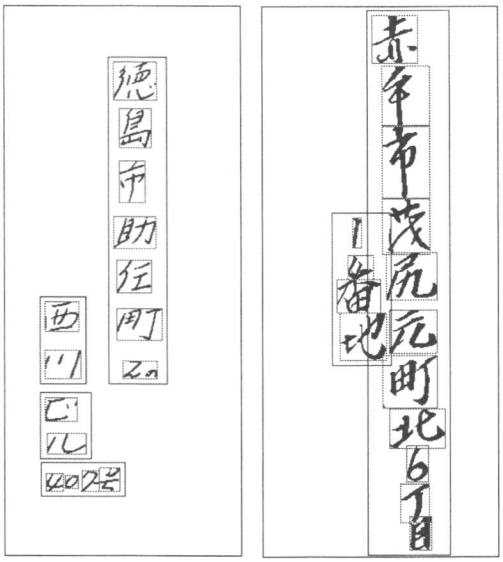
り 2 の例を示した．

文書理解の文字認識では，切り出した文字領域を組み合わせて認識することが一般的である．従って，誤り 1 はこの組合せの部分で解決することができると考えられる．それに対して，複数の文字を統合してしまった誤り 2 は，文字認識部で解決することが困難である．そのため，文書理解システムの文字認識部の前段階として，誤り 2 の減少が望ましい．表 3 より処理(C)を導入することにより誤り 2 の文字数が大きく減少していることがわかる．従って，文書理解の文字認識部の前段階として，処理(C)が有効であると考えられる．

一方，処理(C)の問題点として，誤り 2 の文字数が少なくなる一方，誤り 1 の文字数が大きくなっている．また，処理(C)だけでは誤り 2 を完全に解決することはできない．文字切り出し部においては，以上の問題点が残されている．

4.3.2　文字列抽出

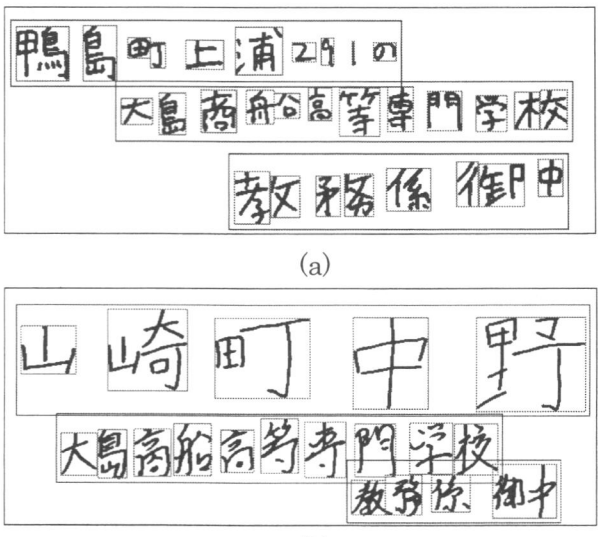
縦書き及び横書きデータに対する文字列抽出率を表 4 に示す．文献[7]では文字列抽出を行っていないため表 4 には文献[7]に対する抽出率は載せていない．文字列抽出の実験の結果，縦書きデータを対象にしたとき文字列抽出率 96.3%，横書きデータを対象にしたとき文字列抽出率 96.0%を得た．図 16 に縦書き文字列，図 17 に横書き文字列の例を示す．図 16(a)，図 17(b)に示すように文字列同士のオーバーラップが発生していても文字列の抽出に成功した．また，図 16(a)に示すように一つの画像に対し横書きと縦書きの文書が含まれている場合でも，文字列抽出に成功した例があった．これは，文字列の抽出では画像全体として文字列の方向を仮定して文字列を抽出するのではなく，文字列を構成する文字同士の接続状況を用いるからである．



(a)　(b)

図 16 縦書き文字列抽出

一方，文字列抽出に失敗するデータのほとんどは，文字切り出しにおいて失敗している．文字切り出し部で異なる文字列に所属する二つの文字を統合してしまうと，必ず文字列抽出に失敗する．この様な文字切り出し部の失敗が，文字列抽出に影響を与えない様にすることが今後の課題である．



(a)

(b)

図 17 横書き文字列抽出

5. まとめ

本論文では画素密度を用いた手書き文字切り出し方式の問題点を考慮し，文字切り出し率を向上するための対策を提案した．提案手法は，画素密度を用いた手書き文字切り出し手法をもとに，文字位置検出に用いる観測スケールの可能な範囲を設定した．更に，文字切り出しの結果から文字列を抽出し，それに基づいて文字切り出しの修正を従来手法に追加した．

計算機シミュレーションでは，画像の全領域を対象にしたとき，縦書きのデータに対し 89.4%，横書きのデータに対し 85.7%の精度で文字を切り出すことができた．更に，従来手法との比較実験により，提案手法の有効性を明らかにした．また，提案手法において文字切り出しに失敗した文字を分析し，文書理解の前段階として解決すべき課題について考察した．

文字列抽出の実験では，縦書きのデータに対し 96.3%，横書きのデータに対し 96.0%の精度で文字列を抽出することができた．特徴として，縦書きと横書きの文字列が混在する画像に対しても，文字列抽出ができるという点にある．

今後の課題として，複数の文字を統合してしまった誤りを少なくするため，文字列抽出部と文字切り出し部の協調動作，文字ストロークの各角度に基づく文字切断処理などが挙げられる．

謝辞

実験に用いた手書き漢字データベース IPTP CD-ROM2 を作成，提供して頂いた旧郵政省郵政研究所の関係諸氏に感謝致します．更に，横書き漢字データベースの作成に協力して頂いた大島商船高等専門学校情報工学科の皆さんに感謝致します．

参考文献

[1] 美濃導彦, "手書き画像処理の現状と動向",信学誌，Vol.76, No.5, pp.502-509, 1993.

[2] 堤田敏夫,城戸 賛,太田一浩,木村文隆,岩田 彰, "文字認識研究の新たな展開に向けて-郵便番号データにみる手書き数字認識の現状-", 信学技報, PRU 96-190,1997.

[3] 若村哲史, "手書き文字認識の高性度化に関する研究", 名古屋大学工学部学位論文，論工博第1333号, June, 1997.

[4] F. Kimura, K.Takashina, S. Tsuruoka, and Y. Miyake, "Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition", IEEE Trans. of Pattern Analysis and Machine Intelligence, Vol.9, no.1, pp.149-153, 1987.

[5] C. Liu, M. Koga, H. Fujisawa, "Lexicon-Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading", IEEE Trans. of Pattern Analysis and Machine Intelligence, Vol.24, no.11, pp.1425-1437, 2002

[6]福島俊一，下村秀樹，森義和，“手書き文字列読みとりのための単語列検索アルゴリズム－文字タグ法－”，情報処理学会論文誌，Vol.37,No.4,1996

[7] 佐長康久，岡村健史郎，浜本義彦，”画素密度検出エージェントによる手書き文字切り出し", 信学技報, PRMU 2000-222,pp.1-8,2001.

[8] 馬場口登，塚本正敏，相原恒博, "手書き日本文字列からの文字切り出しの基礎的考察", 信学論(D), Vol.J69-D, No.9, pp.1292-1301, Sept.1986.