

## デジタルインクの一圧縮・復元方式

内藤敦士 中川正樹

naitoh@cc.tuat.ac.jp nakagawa@cc.tuat.ac.jp

東京農工大学 工学研究科 電子情報工学専攻

〒184 東京都小金井市中町2-24-16

電子タブレットから入力された筆跡は、一定時間間隔でサンプリングされた筆点座標列である。これをデジタルインクと呼ぶ。デジタルインクは符号化されたデータより膨大にメモリを消費するため、これを保存する研究は今まであまりなされなかった。そこで、本稿ではデジタルインクの一圧縮・復元方式を提案する。これは筆点座標列から特定の座標だけを抜粋するものである。また、抜粋された座標列に直線補間とBスプライン補間を組み合わせることで適用させることにより復元を行う。この手法により、データ量を元の約20%に圧縮できた。

## A METHOD OF COMPRESSION AND RESTORATION OF DIGITAL-INK

Atsushi Naito and Masaki Nakagawa

Department of Computer Science, Faculty of Technology,

Tokyo University of Agriculture and Technology

Naka-cho 2-24-16, Koganei, Tokyo, 184, Japan

Handwritten data from a tablet is made up of coordinates sampled from a pen trace at regular intervals. We call this "Digital-Ink". There have been few studies on saving Digital-Ink, because it needs much more memory than coded data. This paper proposes a method of compression and restoration of Digital-Ink. This method extracts only specific coordinates from a pen trace, with this being restored by combining straight and spline interpolation. Experiments showed with that pen traces can be compressed to approximately 20% their original size with this method.

## 1 はじめに

日常生活において、情報の入力、蓄積、及び伝達的手段として、もっとも自然に用いられてきたのが紙とペンである。近年になり、PDA (Personal Digital Assistant)、ペンコンピュータといったペンを用いた計算機も出現している。手書きの利点には次のものがある。

- ・幼い頃からものを表現する手段として慣れ親しんでいる
- ・自然な行動として訓練をしなくても受け入れられる
- ・省スペース
- ・思考を妨げることがない

しかし、計算機環境では上記最終項目が達成されていないと考える。なぜなら、デジタルインクを文字としてユーザが入力した場合、多くの計算機では、即座にそれを認識して符号化しようとする。この場合、認識待ちの時間や誤認識文字の訂正を余儀なくされ、思考を妨げられるからである。個人的メモや友人へ送る手紙などは、符号化された文字より肉筆の文字の方が親しみがあり適している。今後は、キーボードが扱えないユーザの増加が予想される。そのとき、計算機環境においても文字や図形が紙と同様に扱えれば、大いに役立つと考えられる。

入力中に思考を妨げないためには、即時認識をしなければよい。必要に応じて認識を行う方式（オンデマンド認識方式）が最適である。[1] しかしながら、これまでオンデマンド認識方式はあまり採用されていない。それはオンデマンド認識をするには、デジタルインクの保存が必要不可欠だからである。デジタルインクの保存は以下の理由で困難とされる。

- ・データ量が膨大である
- ・データの基本形式を変えると認識不可能になる
- ・共通フォーマットがない

そこで本稿では、上記の問題の解決を目的として、デジタルインクの一圧縮・復元方式を提案する。この方式を用いることにより、デジタルインクの保存に要する資源の浪費を押さえることができ、資源を有効に使える。特に PDA のような少量のメモリしか持たない計算機や、通信分野においてその効果が発揮できる。

## 2 アルゴリズム

### 2.1 概要

デジタルインクは、一見普通の紙の上に描いた線のように見えるが、実際には一本の連続した線ではなく、図1(a)のようにいくつかの座標点を線で結んだ構成になっている。ここで、この座標点をある規則にしたがって分類する。図1(b)に分類結果を示す。中黒の点は他の白抜き点の点に比べて重要な点であるといえる。なぜなら、白抜き点の点は図1(c)のように中黒の点を結ぶことにより、再現できるからである。このようにデジタルインクに含まれる多くの座標点は、その効果を他の点で代行できる。本稿で提案する手法は、これを利用する。つまり、パタンの特徴を示す重要な点(以下、特徴点)だけを保存することが圧縮法であり、保存された特徴点同士を補間して、取り除かれた点を補うことが復元法である。ただし、本稿で提案する圧縮・復元方式は、圧縮されたものが完全に元の状態には復元されないため、不可逆な方式である。

特徴点となるのは、端点、屈折点、交差点、曲線部分の膨らみが最大の点などが挙げられる。従来の手法では、特徴点同士を補間する際には直線だけを用いていた。直線だけで補間する場合には図2(a)のように曲線部分に多くの特徴点を与えなければならない。これに対して、曲線部分を図2(b)のように滑らかな曲線で補間することができれば、特徴点の数をより少なくすることが可能となる。そこで、本稿で提案する手法は、デジタルインクの形状が直線に近い部分は直線補間を用い、曲線に近い部分には曲線補間を用いる。つまり、一本のストローク(ペンダウンからペンアップまでに採集される筆点座標列)に対して、直線補間と曲線補間を組み合わせた補間を行う。次に、復元法、圧縮法の順に詳細説明をする。

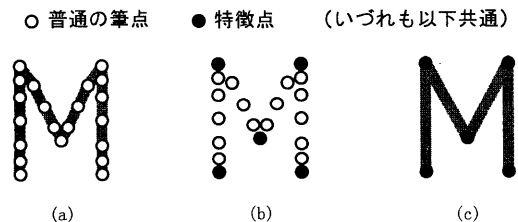


図1: 圧縮の概念



図2: 補間の種類

## 2.2 復元処理

### 2.2.1 補間法

前述のとおり、直線部分を補間するには直線補間を用いる。直線補間とはもっとも単純な補間法で、点と点を直線で結ぶ手法である。

曲線部分を補間するにはBスプライン補間を用いる。Bスプライン補間は座標点列を滑らかなラインでつなぐ補間法である。滑らかなラインを引くときには、与えられた座標点の近くを近似して補間していくのではなく必ず与えられた座標点を通過する。したがって、筆点の再現性が高い補間法と言える。また、Bスプライン補間の中にも数種類あるが、ここでは3次曲線補間を選択した。これは、与えられた  $n$  点  $(x_i, y_i)(i = 0, 1, \dots, n-1; x_0 < x_1 < \dots < x_{n-1})$  を通る区分的3次式で補間する。端点の処理は、両端点で2次導関数を0として処理する。適当な媒介変数を使って刻み幅を調整を可能にしておく。また、X、Y座標を切り分けて別々に処理する方法を用いる。

### 2.2.2 補間法の切分け

一つのストローク中に直線とBスプライン曲線を混在させるためには、互いに適した部分を区別する目的で特徴点に情報を持たせる必要がある。そのために、直線補間のためのフラグ（以下、直線フラグ）、Bスプライン補間のためのフラグ（以下、スプラインフラグ）を用意する。この二つのフラグの示す意味は次のとおりである。

- ・直線フラグと直線フラグが並ぶ区間に直線補間を施す
  - ・直線フラグ、一つ以上のスプラインフラグ、直線フラグという組み合わせからなる区間にBスプライン補間を施す
- なお、図3に補間例を示す。

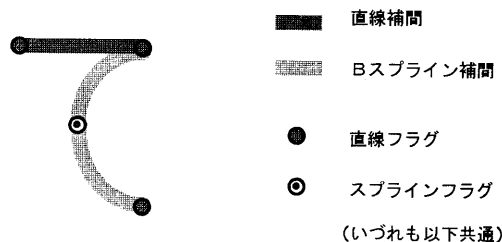


図3: 補間例

### 2.2.3 圧縮データ列

前述の補間法で復元を行うためには、補間法に沿ったデータ列に圧縮処理を行う必要がある。そこで、圧縮データ列の性質を次に示す。

- ・ストローク単位で時系列順に並んでいる
- ・始点、終点の二点が必ず存在する
- ・スプラインフラグが始点、終点となることはない
- ・直線フラグ、スプラインフラグは、その数に制限がない

### 2.2.4 圧縮データ列からの復元

Bスプライン関数には、あらかじめ補間したい区間の座標点列が格納された配列を引数として渡す必要がある。直線補間だけの場合のように単純にはいかない。圧縮データ列から直線補間部分とBスプライン補間部分に切り分けるため、ストロークを復元させるアルゴリズムは少し複雑になる。方針としては、圧縮データ列から座標を一つずつ取得する。座標の属性が直線フラグの場合は、そのまま直線補間する。スプラインフラグの場合は、次に直線フラグが現れるまで、座標点列をバッファへ格納していき、直線フラグが現れた時点でバッファをBスプライン関数へ渡して補間処理をする。この操作をストローク終点まで行えばよい。

## 2.3 圧縮処理

本稿で提案する圧縮処理とは、デジタルインクの中で特徴となる点、例えば、始点、終点、交差点、屈折点、曲線部分でもっとも膨らんでいる点だけを抽出し、直線フラグまたはスプラインフラグをセットすることである。これらの抽出はストローク単位で行う。このうち、始点、終点は無条件に直線フラグの特徴点にする。交差点を特徴点とする理由は、経験的に人間が位置を把握する際

には他のものとの相対位置を基準にすると考えられるからである。なお、すべてのデジタルインクとの交差点を算出するとその効果の割に処理時間が長いので自己交差だけに留めてある。また、他の特徴点とは性質が異なるため、単独に処理を行う。その他の特徴点の抽出は圧縮処理においてメイン処理であり、フラグセットとも絡んだ処理を行う。以下、交差点判定、特徴点抽出、フラグセットの詳細説明をし、最後に全体の手順を示す。

### 2.3.1 交差点判定

交差点判定は特徴点抽出においてのメイン処理ではないため、高速に行わなければならない。そのため、自己交差だけを対象とし、自己交差可能性がない場合には即座に処理を終了する。そこで、次のように処理を行う。

次のいずれかの条件が成り立つとき、ストロークの自己交差はありえないため処理を終了させる。

- ・筆点のX座標値が単調に増加（減少）する
- ・筆点のY座標値が単調に増加（減少）する
- ・連続する筆点を結ぶベクトルの方向成分について最大最小差が 180 度以下

次にストロークをX座標の極大・極小及びY座標の極大・極小で一つのブロックに分割する。これをセグメントと呼ぶことにする。あるセグメントが交差するならば、そのセグメントとは違うセグメントと交差するはずで、そのとき、そのセグメントともう一つのセグメントは、X座標への射影、及びY座標への射影の両方において、重なる部分があるはずである。逆に、どのセグメントとも、X座標への射影あるいはY座標への射影において重なる部分があれば交差しないことになる。図4にセグメントとその射影の関係を示す。図中の番号はセグメント番号および射影番号である。セグメント1については射影との位置関係を点線で示した。セグメント1とセグメント4のように、射影に重なりがある場合だけ、交差判定と交点座標の算出を行う。

こうして、算出された交点座標は実際のストローク上の座標点列中に存在することはまれである。そこで、交差点にもっとも近いストローク上の筆点を交差点として選択する。

### 2.3.2 特徴点抽出

デジタルインクは様々な形状をしているため、始点、終点、交差点以外の特徴点は、特別な情報を含んでい

ない。この中から、残りの特徴点を抽出するために次の手法を用いる。

図5のデジタルインクを例にとり説明をする。まず、黒丸の部分が既に特徴点として抽出された点である。圧縮処理開始時においては、この点が始点と終点になる。図5(a)において点aから点qまでを特徴点抽出区間とした場合、まず、特徴点と特徴点に直線を引く。次にこの直線からの垂直距離がもっとも離れているものを点aから点qの中から選択する。点iがこれにあたり、新しい特徴点とする。仮に点aから点qの中に交差点が存在した場合には、距離差に関係なく交差点を新しい特徴点とする。新しい特徴点が決まった場合には再帰処理に入る。したがって、図5(b)のように、新しい特徴点との間に直線を引く。それに伴い、特徴点抽出区間は点aから点hまでとなる。同様に距離差が最大となる点は点eとなり、これが次の新しい特徴点となる。以上のように処理を再帰的に繰り返してゆく。再帰の終了条件は、すべての点に対する直線との距離差がある一定のしきい値より小さい場合である。このしきい値を特徴点しきい値(単位:ドット)と呼ぶことにする。[2]

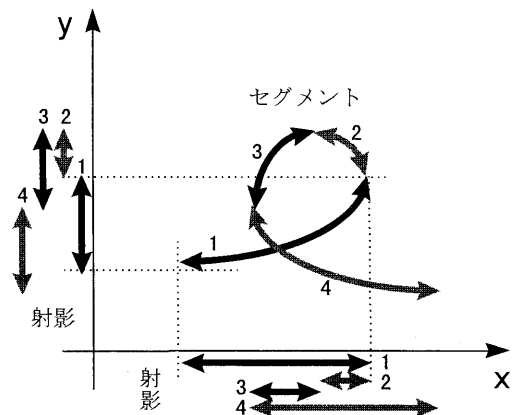


図4:セグメントと射影

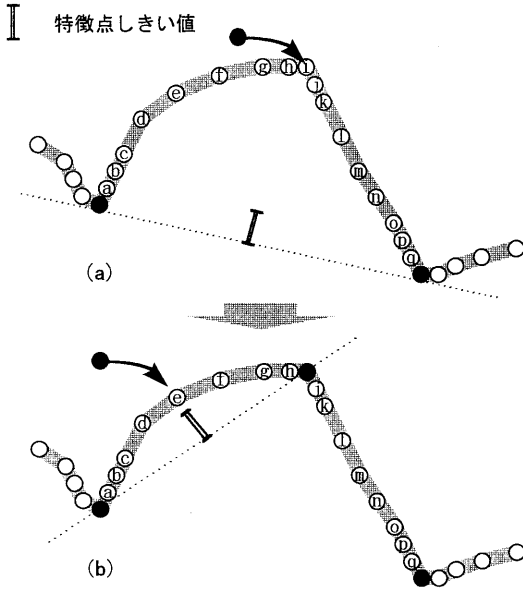


図5: 特徴点抽出

### 2.3.3 フラグセット

復元補助目的のフラグである直線フラグとスプラインフラグのセット方法を説明する。全体としては、初期状態においてすべての特徴点をBスプライン補間する点であると仮定(スプラインフラグのセット)して、直線に近い部分を見つけた場合に直線部分として登録(直線フラグのセット)し直す。そこで問題となるのが、直線に近い部分の判定である。本圧縮手法では二つの判定法を用いる。

一つ目は、特徴点と特徴点を直線で結び、特徴点間にある点と直線との距離差を用いる方法である。これを、距離差によるフラグセットと呼ぶ。図6(a)のように直線から点列までの距離差が離れている場合は曲線部分候補として何もしない。図6(b)のように距離差が小さい場合に直線部分と判定をする。ここで直線と判定された場合には直線の両端にある特徴点に直線フラグをセットする。また距離差は特徴点抽出における距離差と同一のものであるため計算量を節約できる利点がある。

二つ目は、特徴点自体の屈折度を用いる方法である。これを屈折度によるフラグセットと呼ぶ。図7のように特徴点に対して、その前後一点とのなす角度で判定する。角度が小さい時に直線フラグの点であると判定する。ここで判定された場合には、その特徴点だけに直線フラグをセットする。

以上の二方法を用いてフラグセットを行うが、直線部分判定にそれぞれしきい値を用いる。前者を距離差しきい値(単位:ドット)、後者を屈折度しきい値(単位:度)と呼ぶことにする。

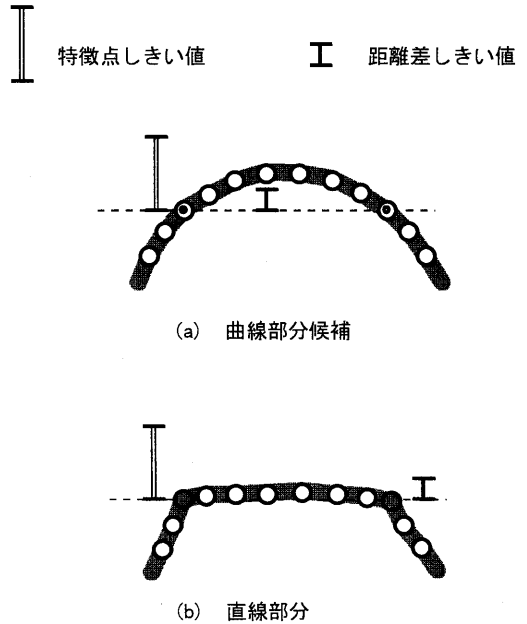


図6: 距離差によるフラグセット

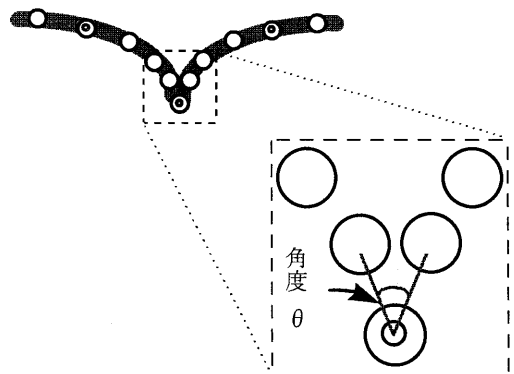


図7: 屈折度によるフラグセット

### 2.3.4 圧縮手順

処理はストローク単位で行い、ストローク中にある二つの特徴点に注目して処理を行う。このうち、時系列順において前に位置する特徴点を始特徴点、後に位置する特徴点を終特徴点と呼ぶことにする

- ①ストローク上の自己交差点判定
- ②始点を始特徴点、終点を終特徴点とし、直線フラグセット。
- ③始特徴点と終特徴点を直線で結び、ストローク上にある始特徴点-終特徴点間の座標点列を処理区間に設定
- ④処理区間に交差点がある場合、交差点を新たな特徴点とする(⑦へ)
- ⑤直線と処理区間にある点との距離差を計算し、その中で距離差が最大の点を抽出
- ⑥距離差がしきい値を越えている場合、その点を新たな特徴点とする(次へ)  
そうでない場合(⑨へ)
- ⑦新たな特徴点にスプラインフラグをセット
- ⑧新たな特徴点を終特徴点にし、再帰処理(③へ)
- ⑨再帰処理を一つ抜ける
- ⑩再帰処理がすべて終了した場合、特徴点以外の点を削除(次へ)  
そうでない場合、終特徴点を始特徴点にし、次の区間へ処理を移動(③へ)
- ⑪スプラインフラグがセットされている特徴点に対して屈折度フラグセット処理をする
- ⑫圧縮データ列完成

## 3 実験と評価

本章では、これまで提案してきたデジタルインクの圧縮効果を、実際のデジタルインクに適用し、有効性を評価する。実験には、文字として書かれたデジタルインクだけを用いた。本来なら図形として書かれたデジタルインクも実験対象にするべきである。しかし、現在データが充実している図形データベースが存在していないため、図形を対象とした実験は行っていない。

### 3.1 実験環境

実験用データには IPDB(Indexed Page Data Base)を用いる。IPDB とは東京農工大学工学研究中川研究室において作成されたオンライン手書きパターンデータベースであり、一人につき約一万文字分のデジタルインクが集められている。このデータベースを用いることで再現性のある実験をすることができる。しかし、現行バージョン

(Version 3.0)では同一座標点を重複記録しない仕様のため、圧縮率の測定において得られる結果は実際の結果より低い値になることが予想される。[3]

実験では IPDB から無作為に六人分のデータ(mdb0005, mdb0010, mdb0015, mdb0020, mdb0025, mdb0030)を選択した。

また、認識には同研究室作成の認識エンジンを使用した。これは、時間線形伸縮マッチングという手法で認識を行う。[4]

実験における各しきい値は予備実験を参考にして次のようにした。

- ・特徴点しきい値: 3ドット
- ・距離差しきい値: 1ドット
- ・屈折度しきい値: 120度

なお、本実験は、メモリを 16Mbyte 実装し、クロック 90MHz の Pentium を CPU とする PC 上で行った。

### 3.2 圧縮率測定実験

一人分のデータにつき、元パタンの合計筆点数、圧縮処理後の合計筆点数から次の式で圧縮率を算出する。

圧縮率 =

$$(\text{圧縮処理後の筆点数} / \text{元パタンの筆点数}) \times 100(\%)$$

これを、六人分について実験した結果を図8に示す。

圧縮率は、人により大きな違いはなく、平均して 20% 前後の安定した結果となった。圧縮率としては十分その有効性を示すことができる数値といえる。

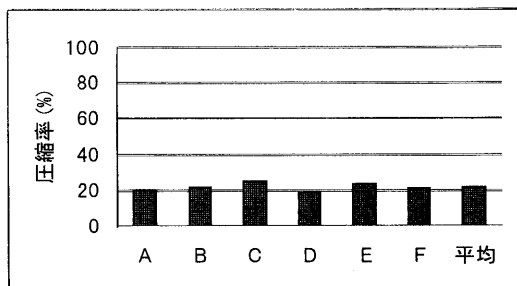


図8: 圧縮率の測定

### 3.3 復元の正確さ実験

復元したパターンと元のパターンとの見た目の類似度は、定量的に測定するのは困難である。そこで、同一の認識エンジンを用いて、元パターンを認識したときの認識率、一連の圧縮及び、復元処理を施したパターンの認識

率を測定し、前者に比べて、どれだけ認識率の低下が発生するかを調査することで類似度の測定を行う。以上の処理を6人分に対して行った結果を表1に示す。

実験により、圧縮・復元処理を行うと認識率の低下が発生することが確認された。しかし、その低下は平均で1.58%であった。この数値は百文字認識したとき、圧縮処理の弊害が原因で一文字から二文字分余計に誤認識することを示している。類似度という点に関しては、文字の特徴を大きく崩してはいないといえる。

表1: 認識率の差

認識率	元パターン	圧縮パターン	差分
A	90.36	89.15	1.21
B	86.15	85.1	1.05
C	84.72	82.39	2.33
D	83.40	81.60	1.80
E	84.03	82.54	1.49
F	84.82	83.22	1.60
平均	85.58	84.00	1.58

### 3.4 総合評価

図9に実験を行ったデジタルインクとその圧縮・復元の実例を示す。同じ文字が上下に並んでいる。上が元データであり、下が圧縮・復元を施したデータである。上下に並べた場合、形状の違いにあまり気付かないことが確認できる。しかし、上下のデータを重ねれば、多少のずれが発生する。しかし、この多少のずれはアプリケーションの工夫次第で解消できる問題であると考えられる。総合的に見た場合、本デジタルインクの圧縮・復元手法は有効性があると判断する。

## 4 おわりに

本稿では、デジタルインクの一圧縮・復元手法を提案した。その結果、圧縮においては元のパタンの筆点数を平均20%程度まで減らすことができ、その形状もほぼ同一にみせることに成功している。提案した手法は、あくまで、筆点の絶対数を削減する手法である。そのため、メモリ効率を考慮した他の圧縮手法との組み合わせにより、圧縮率をさらに高めることが可能である。

デジタルインクの圧縮手法の確立は、これからのペンインタフェースに変化を与える可能性が高い。特にPDAにおいては、デジタルインクの保存が可能なることで、即時認識の必要性がなくなる。これにより、これまでのPDAではあまり例がない、速記メモ及びそのメモの

符号化が可能となる。また、デジタルインクの共通フォーマットが確立されることで、あらゆる計算機間でのデジタルインクの共有や、これを発展させて手書きメールも現実のものとなる。

今後は、本手法を組込んだアプリケーションを試作して、更に処理を検討するとともに、アプリケーションの使用の中でペン入力に対する人間工学的評価を進めていきたい。

遺症はなお、この地  
 遺症はなお、この地  
 域に影を落とす。下  
 域に影を落とす。下  
 BSの放送技術局に  
 BSの放送技術局に

実例1 (mdb0005)

入出米。七年前か  
 入出米。七年前か  
 ら娘の腫さんも手伝  
 ら娘の腫さんも手伝  
 うようくなり、人件  
 うようくなり、人件

実例2 (mdb0010)

字で表記した早い時  
 字で表記した早い時  
 期の例としては一六  
 期の例としては一六  
 の〇年にあるが、「  
 の〇年にあるが、「

実例3 (mdb0015)

図9: 圧縮・復元の実例

## 謝辞

本研究は、情報処理振興事業協会の創造的ソフトウェア育成事業の補助による。

## 参考文献

- [1]M.Nakagawa, K.Machii, N.Kato, T.Souya : Lazy Recognition as a Principle of Pen Interfaces, "INTER CHI\*93 Adjunct Proc. pp.89-90(1993)
- [2]Ishigaki.K, Morishita.T : A Top-down On-line Handwrittern Character Recognition Method via the Denotation of Variation, 1988 Proc. International Conference on Computer Processing of Chinese and Oriental Languages, Toronto, pp.141-145 (1988)
- [3]東山孝生,山中由紀子,澤田伸一,中川正樹:手書き文字時系列筆跡パタンの一解析と今後の計画,公開シンポジウム 人文科学とデータベース,pp.37-42(1995)
- [4]秋山勝彦,中川正樹:ストロークのつながりに寛容なオンライン手書き文字認識,画像の認識・理解シンポジウム (MIRU'94), I ,pp.67-74(1994)
- [5]曾谷俊男,福島英洋,高橋延匡,中川正樹:遅延認識方式を用いた手書きユーザインタフェースの基本設計,情報処理学会論文誌,34,1,pp.158-166(1993)
- [6]奥村晴彦:C言語による最新アルゴリズム事典,技術評論社,pp.129-132(1991)
- [7]松原 敦, 道本健二:ペン・コンピュータ普及元年,日経バイト, 108, pp.176-215(1993-01)
- [8]佐賀聡人, 牧野宏美, 佐々木淳一:手書き曲線モデルの一構成法:ファジースプライン補間法,進学論(D-II),J77-D-II,8,pp.1610-1619(1994-08)