

ペーパインタフェースによる文書編集システムの基本設計

前田 陽 二[†] 中川 正 樹^{††}

本論文では、ディスプレイおよびキーボード、マウスによらず、紙をインタフェースとしてコンピュータと対話する方式(以後ペーパインタフェースと呼ぶ)について考察し、この考察に基づいて、本方式による文書編集システムの検討を行い、その基本設計およびプロトタイプの開発を報告する。ペーパインタフェースによる文書編集システムを実現するためには、編集すべき文書に手書きで記入された記号や文字を切り出して認識する必要がある。我々は、文書の方を手書き記号・文字と区別しやすい形で出力すること、すなわち、編集すべき文書をドット集合(以後ドット文字と呼ぶ)で印刷することを提案する。そして、モルフォロジーによる手法を用いて、ドット文字を消して手書き記号・文字を検出し、ドット文字を強調することで記入した記号とその対象の特定を行う手法を開発した。この手法は、比較的軽い処理で、ユーザに記入制限を課すことなく、また、特別な色のペン、および、これを読む特別なスキャナなどの使用を前提とすることなく実現できる。そのプロトタイプの開発を行い、数人のユーザに試行してもらったところ、紙に訂正を記入して秘書が直してくれる感覚を与え、また、ドット文字による文書は見にくいということもなく、印象は良いものであった。この方式の定量的な評価、改良は今後の課題とする。

A Design of a Paper Based User Interface for Editing Documents

YOJI MAEDA[†] and MASAKI NAKAGAWA^{††}

This paper considers user interface with real papers rather than display, keyboard and mouse and then presents a design and prototyping for document editing. We call this UI as "Paper based User Interface". In order to realize this UI, we need a technology to detect and recognize correction marks written on printouts of documents. We propose a scheme to print out documents in a form that eases detection of handwritten marks rather than restricting the way to write correction marks or forcing the users to use a pen of a specific color and a specific scanner. As such a scheme, documents are printed out with dotted characters. By using methods categorized in "Morphology", they can be easily dropped out to detect correction marks and also they can be easily emphasized to detect where each correction mark is applied. We made a prototype system on a PC and asked several people to try this system. They have expressed good impressions on this system such as "The system works like a secretary", "There is no difficulty to read dotted characters", and so on. Evaluation and improvement remain to be done.

1. ま え が き

1979年に仮名漢字変換技術による日本語ワードプロセッサ(以後ワープロと呼ぶ)が初めて世に出てから20年が経過し、キーボードにより仮名を入力するこの漢字入力方法が日本語文書入力の主流になっている。この技術の出現により、キーボードで文書作成を行う

欧米の文化に合わせて発展したパーソナルコンピュータ(以後パソコンと呼ぶ)のソフトウェアの日本語化をスムーズに行うことができたといえる。

しかしながら、キーボードを利用する歴史のなかった日本においては、新たな技能として修得する必要があった。これはユーザが機械に歩み寄りて実現している側面は無視できず、キーボード操作を苦痛とする人々も多くいることも事実である。また、英語のように容易にスペルチェックを行える文書と異なり、打ち間違い以外に仮名漢字変換誤りの多い現状では、つねに入力結果をディスプレイで確認しながら文書作成を進める必要があり、特に中高齢者にとってディスプレイを見ながらの長時間の操作は目に対する負担は大きく、健康上の問題は無視できないものとなっている¹⁾。

[†] 東京農工大学大学院工学研究科電子工学専攻
Division for Research of Technology, Department of
Electronics and Information Engineering, Tokyo Uni-
versity of Agriculture & Technology

^{††} 東京農工大学情報コミュニケーション工学科
Department of Computer, Information and Commu-
nication Sciences, Tokyo University of Agriculture &
Technology

急速なコンピュータ社会の発展とともに、今後ますます多くの人がコンピュータを用いて文書の作成を行うことを考えると、ここで文書作成における人と機械との関係について見直すことが必要であろうと考えている。

そこで、我々はキーボードとディスプレイを使わずに、編集すべき文書に編集のための記号と文字を記入し、これをスキャナで読み取らせることによって文書編集を行うペーパーインタフェースによるシステムの検討を行っている²⁾。

ペーパーインタフェースに関する初期の研究としては、原稿上に文字の消去や組み替えを指示する修正記号を書き加え、この原稿をファクシミリで読み取らせた文書画像において、修正記号に従って文書画像の切り張りを自動的に行うシステムがある³⁾。ただし、このシステムは、画像上で修正処理を行うものであるため、文字コードなどの文書ファイルを変更するものではない。また、修正記号の切り出しを容易にするため、修正記号を文書の文字に接触することなく記入する必要があった。ペーパーインタフェースによって文書ファイルを変更するものとしては、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ で作成された文書ファイルのプリントアウトに修正記号をカラーインクで記入することにより修正記号の切り出しを容易にし、かつ、この修正記号に対応した修正処理を文書ファイルに対して行うシステムが開発されている⁴⁾。

ところが、できる限り記入制限のないペーパーインタフェースを実現するためには、インクの色を変えることなく文書上に重ねて記号や文字を記入できる必要があり、これはかなり困難な技術とされている⁵⁾。この困難を回避するために、文献 3) では文書上の文字に接触しないで訂正記号などを記入することが求められ、記入者に精神的負担を強いることになっている。また、文献 4) では、特定の色のカラーインクを用いて記入することが求められ、文書を修正する際にはユーザは修正作業用のペンに持ち替えなければならず、また、特定の色のカラー文字だけを検出する機能のあるイメージリーダーが使うことができない。紙の良さは外出先でもどこでも朱入れが可能であることなのに、特定のペンを持っていないと朱入れができないことになる。

我々は、人々がこれまで行ってきたのと同じような方法で、文書の校正、推敲を進めることができ、かつ、インターネットなど通信インフラがますます使いやすいものになってきていることから、文書入力的手段としてほとんどのファクシミリやスキャナを利用し、処理はサーバで集中して行うことにより、ユーザが新たに専用の機器を購入せずに利用できることを目指して

いる。具体的には、編集すべき文書をドットの集合(以後ドット文字と呼ぶ)で印刷し、モルフオロジー⁶⁾によって切り出す手法を考案した。この手法は、比較的軽い処理で、ユーザに記入制限を課すことなく、また、特別な色のペン、および、これを読む特別なスキャナなどの使用を前提とすることなく、実現できるといった長所がある。さらに、そのプロトシステムの開発を行い、この方式の可能性と安定性の確認を行った。

2. ペーパーインタフェースによる文書編集システムの設計方針

2.1 ペーパーインタフェースの必要性

現在、流通する文書にほとんど手書きの文書はなくなっており、ワープロ出力された文書が一般化している。また、会社などにおいては、ネットワーク上で文書が管理され、流通するようになり、文書の電子化はますます一般化しつつある。このような中で、ほとんどの人が、何かしらの形で文書の電子化にかかわらざるをえなくなってきており、だれでもが簡単に使える日本語入力装置の出現が待たれている。

簡単に使える日本語入力装置の条件とは以下のとおりである。

- (1) 簡単な教育で使えるようになり、一度教育を受ければ、長期にわたって再教育の必要がないこと
- (2) 個々の利用者個人で高価な装置を買う必要がなく、さらに、いつでもどこにいても利用することができること

ワープロが出現する前は、文書を活字出力するには、まず原稿を作成してその原稿を専門職であるタイピストに渡し、タイプできたものを手書きで推敲して再びタイピストに渡して修正させていた。この方法は、タイピストにとっては、負担が大きいものであったが、入力を依頼する側にとっては、特別な教育を必要とせず、どこでも作業ができるといったメリットがあった。ワープロが一般的に使われだすと、タイピストに原稿を渡すのではなく、自分でタイプすることが要求されるようになってきた。しかし、ワープロを一般的に使うようになって、多くの人は、文書の内容や構造を考えたり、推敲や再構成の段階では、紙とペンを用いて作業している。この理由は⁷⁾

- (a) どこでも作業ができること
- (b) 広い範囲を一度で見渡せること

ドット文字は従来ドットプリンタで印字された文字の意で用いられた経緯があるが、ここではプリンタの方式によらず、印字された結果が図 7 に示すように互いに接しない独立したドットの集合として印字されている文字を指す。

- (c) CRT や液晶面より紙出力の方が見やすいこと
 - (d) 書き込みが自由, 手軽であること
 - (e) 落ち着いて作業ができること
- をあげることができる.

この紙の特長をヒューマンインタフェースに取り込むための技術開発は, オンライン文字認識技術の分野, および文書読み取り技術の分野で進められている.

オンライン文字認識技術の分野では, 上記 (d), (e) の特長を取り込む検討が行われた⁷⁾. しかしながら, 上記 (a), (b), (c) の比較項目においてペン入力装置が紙の使いやすさをしのぐことは考えにくく, また, メーカーごとにインタフェースが異なるといった問題もあり, 当面, 紙とペンに代わるものにはならないと予測される. もう一方の文書読み取り技術の分野では, インタフェースは紙にして, 従来タイピストが行ってきた作業をコンピュータにさせる考えがある. 手書き, あるいは活字の文書をそのまま電子コードに変換する研究の発表は多くあり^{8),9)}, 文書読み取り装置は広く利用されはじめている. しかしながら, その読み取り性能は完全に正しく読み取れるまでは至っていない. 特に日本語の文章の場合, 文字の読み取り誤りだけでなく, 1つの文章の中でも漢字, アルファベット, 数字など大きさが異なる文字が混在する場合があるため, 文字の切り出し誤りも多い. そのため, 文書読み取り装置を利用した場合, 文字の読み取り結果の修正は必須であり, 現在, 多くの文書読み取り装置では, 出力の修正はユーザがディスプレイに表示された読み取り結果を見て, キーボードとマウスを用いて修正している. この場合, ユーザは修正の操作方法を学習する必要があり, ユーザにとってかなりの負担となる.

以上の検討より, キーボード入力による仮名漢字変換に代わり, より自然なインタフェースを実現するためには, 文書の認識方式の研究に加えて, 読み取り出力の訂正を紙の上で行えるようにすることが必要である.

インフラの面でも, ネットワークが発達し, サーバ技術も向上していることから, ユーザは個人的にはまったく何の認識装置も持たずに, ファクシミリやスキャナから原稿を入力し, これを認識サーバに送って処理を受け, ファクシミリやプリンタから出力された読み取り結果, あるいは修正結果をチェックしながら推敲していくことも可能な状況になってきている.

2.2 基本機能の検討

文書の編集においてどのようなペーパインタフェースを実現すれば良いかを検討する. まず, このシステムで何ができなければいけないかをあげる.

表1 ワープロの主要機能

Table 1 Basic functions of a word processor.

分類	機能
文書の入力	文字の入力と消去, 文字列の移動やコピー, など
文書のレイアウト指示	文字フォントや文字サイズの指示, 文字数/列数の指示, 段組の指示, 図表の挿入の指示, など
文書ファイルの管理	文書ファイルの検索と格納, ファイル名やフォルダ名の指定, 印刷の指示, など

表2 文書読み取り装置の主要機能

Table 2 Basic functions of a document reader.

分類	機能
文書読み取り	読み取り領域の指定, 読み取りフォントの指定, など
読み取り結果修正 (誤読処理)	文字消去, 正解文字入力, 候補文字からの正解文字の選択, など
読み取り結果修正 (再処理)	単語処理の指示 結合指示による再認識 分離指示による再認識

(下線は実現すべき機能)

現在流通しているワープロの基本的な機能は実現する必要がある. これらは, 文書の入力, レイアウトの指示および文書ファイルの管理に関するものであり表1に示す.

また, 一般的な文書読み取り装置で実現している機能および実現すべきと我々が考えている機能を表2にまとめる. この表の中の再処理について説明を追加すると, 単語処理(認識結果の情報と, 単語としてありうる組合せであるかの情報により, 単語として認識結果を出力する処理)は, 単語部分を指示して再度単語処理を行うもので, 自動的に単語を切り出せなかったときに効果がある. 結合指示は, 1つの文字を2つの文字として切り出してしまった場合であり, 1つの文字として再認識する. 分離指示は, 逆に2つ以上の文字が1つの文字と判断された場合で, 再度文字切り出しを行った後に認識する.

表1, 表2に示す処理を行う場合, 文字の修正, 追加など, 文書に直接書き込んで指示することで行える処理と, 印刷の指示や文書ファイルの検索など直接文

書に書き込むのではなく、処理内容を指示するための帳票（以後指示帳票と呼ぶ）を用いた方が良い処理がある。指示帳票を使って、文書ファイルを管理、利用する方法は、これまでもいくつかの試みが行われている^{10),11)}。これは、帳票に記入する指示内容と記入方法などの帳票設計ができれば実現可能であり、ここでは、修正すべき文書に直接書き込んで修正の内容を指示するための記号（以後修正記号と呼ぶ）について焦点をあてる。

2.3 修正記号に関する考察

修正すべき文書上に記入する修正記号のセットについて検討する場合、オンライン文字認識の研究成果が参考となる。オンライン文字認識におけるジェスチャ（修正記号と同様のもので修正のための操作）についての検討はこれまで多く行われており^{12)~14)}、ジェスチャの数は、ユーザに記憶の負担をかけないために5個から多くても10個以下であろうと推測される。また、修正記号の形状は、人の直感に合うもので、かつ、互いに類似した記号がないことが望ましい。

表3に本システムの修正記号の一覧を示す。JISのZ8208の印刷校正記号を基本として、新たに認識結果の修正に関する記号を検討して追加した。なお、修正記号は、次のように分類される。

- (1) テキスト編集：文書入力、修正を行うものである。削除は、認識の安定を高めるため2本線とされている。また、“変更”として、認識誤りを修正するために認識候補文字の次候補と変更する記号を加える。
- (2) レイアウト編集：文書レイアウトの指示を行う。
- (3) 認識結果修正：文字の切り出しミス、単語の切

り出しミスに対応する。

3. ペーパーインターフェースによる文書編集システムの基本設計

ワープロを使わない人が文書を作成する場合、まず原稿を作成し、この原稿をタイピスト（秘書）などに渡して入力を行い、この結果を推敲していくことにより最終的な文書を作成する。我々は、この文書作成の作業におけるタイピストなどの役割をネットワーク上のサーバに置き換えることを提案する。ユーザは、原稿用紙、あるいはプリンタの出力に書き込み、これをファクシミリなどを使ってサーバに転送することによりキーボード操作なしに文書を作成することができる。システムイメージ図を図1に示す。

3.1 システム構成

3.1.1 処理の流れ

全体のシステムのイメージをつかむために、既存の文書を読み取って、読み取り誤りを修正する場合を例にとって操作フローを図2に示す。操作は既存文書の読み取り、修正記号の記入、修正処理の順に行う。

(1) 既存文書の読み取り

指示帳票に文書を読み取るための読み取り領域、文字フォントなどの指示と、読み取り結果を記憶するファイル名を記入し、この指示帳票と読み取るべき既存の文書をスキャナやファクシミリで電子データに変換してシステムに送る。システム側では、その文書イメージより文字を切り出して認識し、その結果をプリントアウトすると同時に文書イメージなどとともに指示された文書ファイル名に記憶する。

(2) 修正記号の記入

出力されたプリントアウトに、必要に応じ表3に示す修正記号をペンなどで書き込む。

表3 修正記号
Table 3 Correction marks.

		修正記号	
テキスト編集	削除		対象文字の上に2本線を引く
	変更		対象文字に縦線を1本引く
	挿入		挿入位置を指示する
レイアウト編集	改行		行を起こす位置を指示
	行の接続		行の終わりと先頭をつなぐ
認識結果修正	単語処理		単語部分を囲む
	分離		対象文字の上に記入
	結合		対象文字を囲む

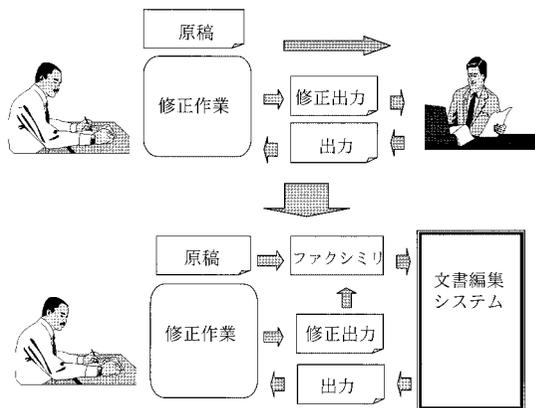


図1 システムイメージ
Fig. 1 System image.

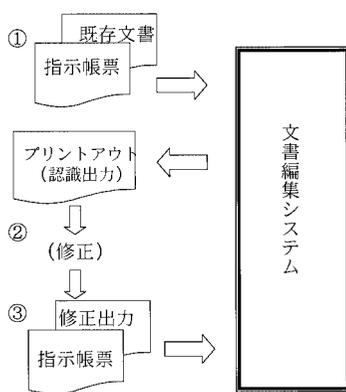


図2 操作フロー
Fig. 2 Operation flow.

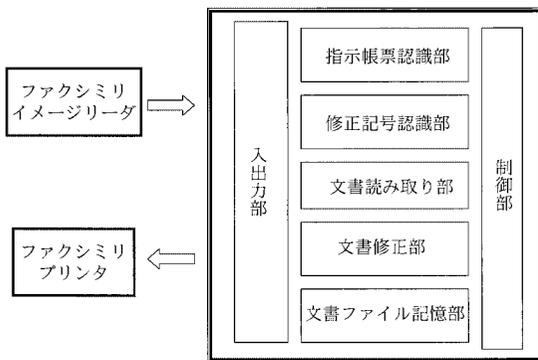


図3 システム構成
Fig. 3 System architecture.

(3) 修正処理

指示帳票に修正すべき文書ファイル名を記入し、修正記号を記入したプリントアウトとともにスキャナやファクシミリで電子データに変換する。システム側では修正記号を認識して指示された文書ファイル名の文書の修正を行う。

3.1.2 システム構成

システム構成を図3に示す。

(1) 指示帳票認識部

ファクシミリやスキャナによって読み込まれた指示帳票のあらかじめ決められたフォーマットに記入されている文字とマークを認識する。

(2) 修正記号切り出し部

ファクシミリやスキャナによって読み込まれた修正入りのプリントアウトイメージから、修正記号・文字を切り出して認識する。修正記号の切り出し方式については3.2節で詳しく説明する。

(3) 文書認識部

ファクシミリやスキャナによって読み込まれた文書

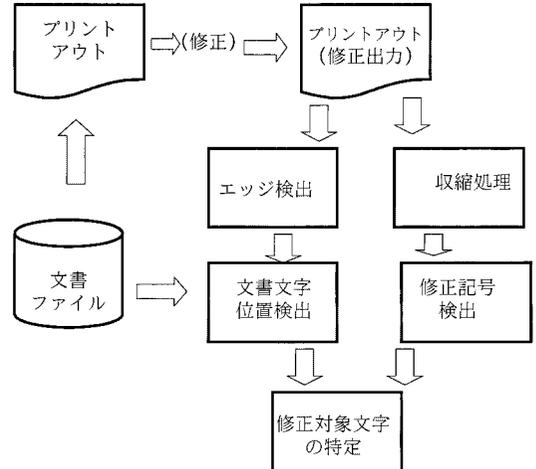


図4 修正対象文字の特性
Fig. 4 Identification of target characters.

イメージから文字を切り出して認識する。なお、活字文書読み取りについては OCRKIT¹⁵⁾を利用した。一方、手書きの読み取りエンジンは、Liuらによる。このエンジンは、ETL-9Bのデータベースに対して認識率97.5%、認識速度25文字/秒を Pentium-II 300 Mhz 上で達成している^{16),17)}。

(4) 文書修正部

修正記号切り出し部で切り出された修正記号に従って、文字の削除、追加などの文書ファイルの変更を行う。

(5) 文書ファイル記憶部

文字認識部での処理結果である読み取り結果(候補文字も含む)と文書イメージを記憶する。

3.2 修正記号の検出

このシステムを実現するための重要な課題として、文書上に重ねて記入された文字、記号の分離抽出技術の開発がある。それは、ユーザに負担を強いたり、特別なハードウェアを必要としたりしない方式でなければならない。我々は、文書の出力に工夫をすることで、この問題を解決する着想を得た。具体的には、現在ほとんどのプリンタでドット文字の出力指示を行うことができることに注目し、修正対象の文書をドット文字で出力することにより、これを実現した。

修正記号などの切り出しフローを図4に示す。この処理においては、文書のイメージを取り込むときに傾いて取り込まれる可能性があることから、修正記号の

プロトタイプシステムを構築するにあたり、活字文字認識ツール(OCRKIT)と手書き漢字認識システムを利用させていただいた、三菱電機(株)と Cheng-Lin Liu 博士にそれぞれ深謝いたします。

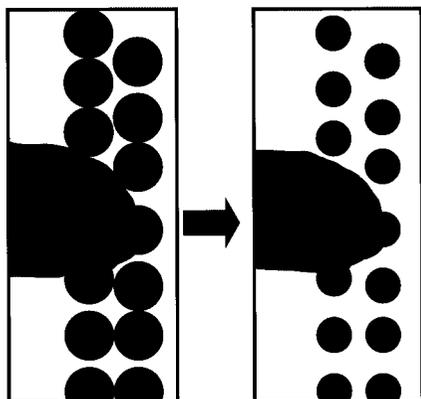


図5 収縮処理(概念図)
Fig. 5 Contraction process.

切り出しだけでなく、修正記号の修正対象文字の特定を行う必要がある。

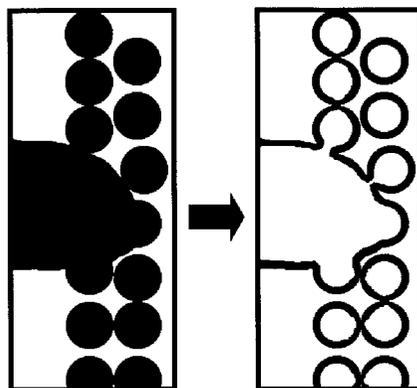
処理は大きく3つの項目からなる。1つは、プリントアウト上に記入された手書きの修正記号などを切り出す処理である。すなわち、プリントアウトされたドット文字を除いて修正記号だけを切り出し、認識し、その位置を求める。もう1つは、逆にドット文字でプリントアウトされた文字情報を強調して文字位置を検出する処理である。最後が、修正記号の位置と文書上の文字位置から、修正記号が文書上のどの文字を対象としているかの対応付けを行う処理である。以下具体的な方法について述べる。

(1) 修正記号の切り出し

まず、入力画像に対して収縮処理を行い、各ドットをできるだけ分離する。手書き文字はドットに比べて太く記入されるため、収縮処理をしても影響は少ない。収縮処理の概念を図5に示す。次に、連結する画素ごとに画素数を数え、閾値以下のものを除く。この処理により記入された記号と挿入文字だけを残すことができる。次に、修正記号と挿入文字を切り出して認識する。認識結果はその位置情報とともに記憶する。

(2) ドット文字位置の検出

文字位置は、横方向の周辺分布により行位置を求め、次に行ごとに縦方向の周辺分布をとり文字ごとに切り出して求めることが多い。このシステムの場合、修正記号や文字が文書に重ねて記入されるため、従来の方法では、この修正記号や文字がノイズとなり、図6(b)に示すとおり縦方向の周辺分布をとっても行位置や文字位置が検出できないなど、ドット文字の切り出しを妨げることが考えられる。そこで、記入された記号や文字の影響を極力減少させるため、入力イメージ全体



(a) 概念図

(b) 原イメージ

(c) エッジイメージ

図6 エッジ検出処理

Fig. 6 Edge detection process.

にエッジ検出をまず行い、それから文字位置検出を行う。エッジ検出によりドット文字部は大量のエッジを生成し、反対に記入された記号はその輪郭分のエッジしか生成しないために、記入された記号の影響を薄くすることができる(図6(c))。エッジ検出処理の概念を原イメージとエッジ処理後のイメージとともに図6に示す。次に、文書ファイルの情報により、行数や各行の文字数と文字の大きさなどの情報をあらかじめ知ることができ、この情報を使って、ドット文字の位置を確実に検出する。

(3) 修正対象文字の特定

入力イメージ上の修正記号位置とドット文字位置の対応付けを行って、修正記号がどの文字を修正対象としているかの特定を行う。

4. プロトタイプ

文書認識結果の修正を目的として、表3に示す記号の中から4種(削除, 変更, 分離, 結合)の修正記号に限り、認識および修正処理を行うプロトタイプシステムを作成した。以下、各段階における処理結果について説明する。なお、本プロトタイプは挿入記号を含んでいない。したがって、挿入文字の誤認識などの問題は発生しない。

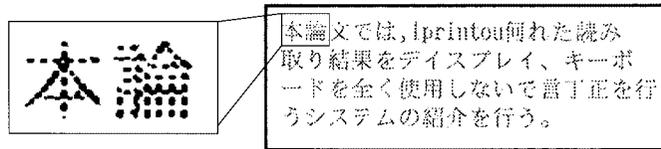
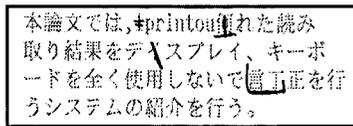
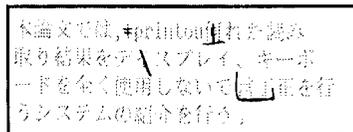


図7 ドット文字例

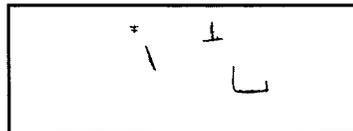
Fig. 7 Dotted characters.



(a) 入力イメージ



(b) 収縮処理 (1回)



(c) 微小点の除去

図8 修正記号の切り出し

Fig. 8 Extraction of correction marks.

評価に用いた文書は、10ポイントのMS明朝体をドット文字で出力したものであり、その濃さは見やすさと処理のしやすさを考慮して選んだ。ドット文字の例をその一部を拡大して図7に示す。次に、修正文字の切り出し例について図8に示す。修正出力の入力イメージに1回収縮処理を行うことによりほとんどのドットを微小な孤立点にすることができる。また、ドット文字位置の検出に用いるドット文字の輪郭画像の例を図9に示す。この画像と文書ファイルの行ごとの文字数や句読点などの検出しにくい文字の出現位置(行の何文字目にあるか)の情報によりドット文字位置を安定して検出することができる。対応付けの結果について図10に示す。

プロトタイプを6人の被験者に修正文字を記入するなどして試行してもらったところ「おもしろい」「使ってみてほしい」「紙に訂正を記入して秘書が直してくれる感じがする」などの意見が得られた。また、ドット文字による文書は見にくいかとの質問に、全員が否定した。被験者に示したドット文字の文書例(部分)を図11に示す。総じて印象は良かった。

なお、プロトタイプはアイデアの基本的な部分だけを実現したものであり、具体的な文書入力・訂正な

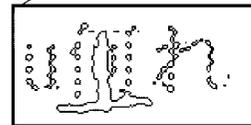
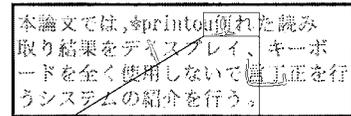
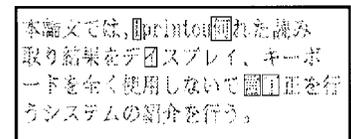
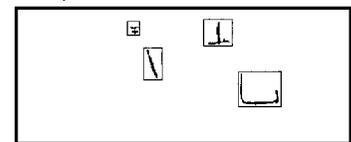


図9 輪郭画像

Fig. 9 Edge image.



(a) 文字位置情報



(b) 記号位置情報

図10 対応付け処理

Fig. 10 Merging information.

どのタスクを課してのユーザビリティ評価は行っていない。

実現手段の予備の評価としては、記入される修正記号が少ない場合には、安定して切り出せることは確認した。しかし、長い文章が挿入される場合など、記入される記号や文字数が多い場合については、エッジ画像を使っても記入された文字の影響が残る可能性がある。また、図10の削除記号の検出結果にも見られるように、ドット文字と修正記号が接触したまま切り出される場合があり、修正記号の種類を増やす場合には安定して認識できるようにすることを考慮しなければならない。この方式の定量的な評価、改良は、今後の課題とする。

近年パソコン等の飛躍的な発達により、文書作成作業は容易になったと言われている。しかしながら、これはユーザが機械に歩み寄って表現している側面は無視できず、機械操作を苦痛とする人々も多くいることも事実である。

図 11 ドット文字による文書例 (10ポイント, 明朝体)

Fig. 11 A document with dotted characters.

5. おわりに

ヒューマンインタフェースにおいて、これまであまり検討されることのなかったペーパインタフェースについて検討を行い、この中で、文書の入力だけではなく、文書認識によって認識された結果の修正を含めてペーパインタフェースで対応するシステムの検討を行った。次に、キーボードとディスプレイを使わずに、編集すべき文書に編集記号や文字を記入し、これをファクシミリやスキャナで読み取らせることによって文書編集を行うペーパインタフェースによる文書編集システムの基本設計を行った。さらに、このシステムを実現するうえで困難な課題とされていた文書に重ねて記入された記号や文字を認識するために、編集すべき文書をドット文字で印刷することを提案した。こうすることにより、簡単な画像処理技術でドット文字を消すことができ記号や文字を検出し、また、ドット文字を強調することで記入した記号の対象文字の特定を行う方式を開発した。さらに、そのプロトシステムの開発を行い、この方式の安定性を確認した。

今後、修正記号の種類拡大と、挿入などで記入された自由手書き文字の切り出し、認識への対応など、実用化へ向けての方式の拡張、モルフォロジーを用いたパターン処理手法の処理時間や安定性の評価、およびパターン認識方式としての評価と安定化を進めていきたい。また、レイアウト指示に関するペーパインタフェースについても、実用化の検討を進めたい。さらに、ネットワーク上のサーバに実装して、その操作性についても評価を行っていきたいと考えている。

参考文献

- 山田尚勇：タイプライタの歴史と日本文入力、情報処理，Vol.23, No.6, pp.559-564 (1982).
- Maeda, Y. and Nakagawa, M.: A paper based user interface for editing documents, *Proc. HCI International'99*, Vol.1, pp.243-246 (1999).
- 末永康仁：ファクシミリを用いた原稿自動編集システム，電子通信学会論文誌，Vol.J63-D, No.12, pp.1072-1079 (1980).
- Bunke, H., Gonin, R. and Mori, D.: A tool for versatile and user-friendly document correction, *Proc. 4th ICDAR'97*, Vol.1, pp.433-438 (1997).
- 小川英光(編著)：パターン認識・理解の新たな展開—挑戦すべき課題，電子情報通信学会 (1994).
- 小畑秀文：モルフォロジー，コロナ社 (1996).
- 曾谷俊男，福島英洋，高橋延匡，中川正樹：遅延認識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，Vol.34, No.1, pp.158-166 (1993).
- Maeda, Y., Yoda, F., Matsuura, K. and Nanbu, H.: Character Segmentation in Japanese Hand-Written Document Images, *Proc. 8th. IAPR*, pp.769-772 (1986).
- 黄瀬浩一：文書画像理解の目指すもの，信学技報，PRMU97-248, pp.55-62 (1998).
- Johnson, W., Jellinek, H., Lotz, L., Pao, R. and Card, S.: Bridging the Paper and Electronic Worlds: The Paper User Interface, *Proc. INTERCHI*, pp.507-512 (1993).
- Rao, R., Card, S.K., Johnson, W., Klotz, L. and Trigg, R.H.: Protofoil: Storing and Finding the Information Worker's Paper Document in an Electronic File Cabinet, *Proc. CHI'94*, pp.180-185 (1994).
- Carr, R. and Shafer, R.: *The Power of Penpoint*, Addison-Wesley (1991).
- Microsoft: Windows for Pen Computing Version 1.0a ユーザーズガイド，マイクロソフト (1992).
- Nakagawa, M., Oguni, T. and Yoshino, T.: Human Interface and Applications on IdeaBoard, *Proc. IFIP TC13 Int'l Conf. on Human-Computer Interaction*, pp.501-508 (1997).
- 三菱電機：ToolMASTAR OCRKIT (PC版) 操作説明書 (1997).
- Liu, C.-L., Kim, I.-J. and Kim, J.H.: High Accuracy Handwritten Chinese Character Recognition by Improved Feature Matching Method, *Proc. ICDAR'97*, pp.1033-1037 (1997).
- Liu, C.-L. and Nakagawa, M.: Prototype Learning Algorithms for Nearest Neighbor

Classifier with Application to Handwritten Character Recognition, *Proc. ICDAR '99*, pp.378-381 (1999).

(平成 11 年 10 月 30 日受付)

(平成 12 年 4 月 6 日採録)



前田 陽二 (正会員)

1972 年早稲田大学理工学部電気通信工学科卒業。1974 年同大学大学院修士課程修了。同年三菱電機(株)入社。以来、主にパターン認識、特に文字認識・文書認識の研究開発に

従事。1998 年東京農工大学博士後期課程入学、在籍中。電子情報通信学会会員。



中川 正樹 (正会員)

1977 年東京大学理学部卒業。1979 年同大学大学院修士課程修了。同在学中、英国 Essex 大学留学 (M.Sc. with distinction in Computer Studies)。1979 年東京農工大

学工学部助手。現在、教授。日本語計算機環境、文書処理、オンライン手書き文字認識、手書きインタフェース、情報教育等の研究・教育に従事。理学博士。