

ペン入力による ドリル答案作成／採点システムプロトタイプ

曾谷俊男

日本アイ・ビー・エム (株) 東京基礎研究所

近年、記憶媒体の大容量化や通信技術の進歩により、高度な教育シナリオを持つエンドユーザ向け CAI システムが実現可能となった。しかし、生徒が直接触れるユーザインタフェース(UI)は、キーボードとマウスに頼るものがほとんどである。本論文は、計算機そのものとの関係のない内容を学ぶエンドユーザ向け CAI にはペンによる UI が最適との考えに基づき、ドリル形式による答案作成／採点ヘルパシステムを提案する。本システムでは、遅延認識と暗黙の記入枠の考え方を採用した。これにより、認識結果や記入枠に注意を奪われることなく、示された問題だけに集中できる環境を提供する。問題の形式は選択問題などに限定されることがなくなる。

A Prototype of a Drill Answering / Scoring System through Pen Input.

SOUYA Toshio.

Tokyo Research Laboratory, IBM Research.

Recently, CAI systems with high quality scenario becomes realizable. But most of these systems employ user interface (UI) systems through a keyboard and a mouse. This paper presents a drill answering / scoring helper system through a pen interface that is the most suitable for these CAI systems. This system employed "lazy recognition scheme" and "tacit answering field" to enable users concentrate on only on questions but user interface problem. Questions are allowed to have not only answer form of selection but also vary forms.

1. はじめに

従来、熟や家庭教師などで行なわれてきた一般生徒向け教育を、コンピュータを用いて行なうことが現実的となってきた。例えば CD-ROM と通信を組み合わせた在宅教育などが商品化されている。

今後、さらなるパーソナルコンピュータの普及により、このような在宅教育が伸びてゆく下地ができてくると言えよう。

ところが、このような商用システムでは、生徒が直接触れるユーザインタフェース (UI) を、専らキーボードとマウスに頼っている。このため、次の問題がある。

- (1) キーボード操作自体に慣れが必要である。
- (2) 2次元配置による解答（分数、数式、傍線など）の入力が困難である。
- (3) キーボード操作の負担を低減するため、選択式解答に解答形式が偏る。

まず、(1)は、慣れてしまえば問題ないが、慣れるまで学習効率が低下することは否めない。(2)は本質的な問題である、キーボードが数式などを入力するのは、困難が伴う。結果として、(3)の問題が生じる。これで、入力作業自体は楽になるが、選択式問題に偏るのは、教育上疑問が残る。

これらの問題を解決することを目的とし、ペン入力をういた研究が、主に数式入力を中心に行なわれている[1,2]。

これらの方法では、数式入力用の特殊な UI を使用する。これでは、ペンによる直接指示のメリットはあるが、慣れを要するという意味ではキーボードによる方法とさほど変わらない。

特殊な UI なしに数式を入力するための研究もあるが、数式を入力するだけの域を出ない[3,4]。

筆者は、ペン入力の特徴は、慣れを要せず、結果として入力作業自体に何ら注意を要しないことと考えた。この特徴を生かすため、遅延認識方式[5]を採用したドリル答案作成システムを考案した。さらに暗黙の記入枠を利用することによって、見掛け上まったく記入枠のないドリル答案の採点ヘルプ処理を考案した。以下で、その思想とプロトタイプについて述べる。

2. 本システムの基本思想

学習方式として、伝統的なドリルを想定した。元来ドリルは、紙面上に印刷された問題を同一紙面上で解いていく方式である。この方式を計算機上にペン入力システムとして実現することは、元々鉛筆と消しゴムで行なっていた作業をそのまま再現でき、違和感がない。また、ペン入力の直接指示の恩恵を受けやすい。

2.1 遅延認識

認知心理学の教えるところによれば、ある例外を除いて、人間があらゆる作業（思考、操作、認識など）を行なうためには、「注意」が必要である。ある1時点の「注意」には限界があるため、複数の作業を同時に行なえば「注意」は分散し、各々の作業効率は低下する。

さて、まったく注意を必要としない作業（上で述べた「ある例外」）もある。それは、過度に練習された作業で、これを「自動化」された作業という。

ここで、手書き作業について考える。文字を書く作業は日常生活で日々使用されているもので、それ自体に注意を必要としない。つまり自動化された作業と言える。したがって、文字を書く作業と同時に他の作業を行なうことができる。

逆に、キーボードによる日本語入力は変換時に確定操作が必要であり、ここの判断のために「注意」を要する。マウスによる選択の場合でも、何を選択するか判断しなくてはならず、ここに「注意」を要する。したがって、これらの作業は、極めて自動化されにくく、もう一つの作業と並行して行なうと、その作業効率の低下を引き起こすことになる。

さて、多くのペン入力システムでは、1文字から数文字書かれた時点で認識処理結果が示される（認識処理を行わないものを除く）。リアルタイムに認識結果を表示し、確認を促すのである。この確認作業は、仮名漢字変換の候補選択作業と同じで、「注意」が必要であり、思考作業を中断させる。

この思考の中断をなくすため、遅延認識では、筆記作業中に一切認識結果を表示しない。筆記中は、インキング（画面上にインクデータをそのまま表示すること）とインクデータの蓄積だけである。認識結果を表示するのは、ユーザの思考が自発的に途切れた後、ユーザが指示した時になる。認識結果が遅延して表示されるため、遅延認識と呼ぶ。

この遅延認識の利点は、

- (1) ユーザの思考を乱さない
- (2) リアルタイムにインク文字や構造を切り出す必要がないため、自由に筆記できる
- (3) 筆記終了後、認識処理を行なうので、大域的特徴を使用できる。

の3点である。

この遅延認識は、ドリル答案作成／採点ヘルプシステムに、そのまま適用できる。答案作成と採点を、筆記と認識処理に対応させるのである。答案作成中は、見掛け上インキング処理だけが行なわれる。答案作成終了後、認識処理を伴う採点ヘルプ処理が行なわれる。つまり、遅延認識を応用すると、

- (1) 答案作成時に生徒を問題に集中させることができる。
- (2) UIの制限なく、紙によるドリルと同様に、自由に答案を作成させることができる。
- (3) 大域的特徴を用いて、採点処理をできる。ということになる。

2.2 暗黙の記入枠

前節に述べたように、遅延認識を応用することによって、自由に答案を作成させることができる。しかし、本質的にドリルの答案では、予め紙面上に用意された領域に期待された答えを書くことを要求されている。これによって正答とみなされるわけである。この予め答えを書くことが期待されている領域を本稿では「暗黙の記入枠」と呼ぶことにする。

暗黙の記入枠によって、ドリルの設問それぞれに対応する書き込みを決定することができる。インクデータは、オフラインのビットマップデータと異なり、ストローク（ペンダウンからペンアップまでの筆跡）単位に分割されているため、設問との対応付けは容易である。暗黙の記入枠は、設問の内容や、その配置そのもので決定される。

もちろんヒントを与えるために明示的な記入枠を示すこともできる。例えば、計算問題の解答が、

分数、帯分数、整数になる可能性がある時、文字ごとの記入枠によってその配置を示せば、ヒントとなる。もし、暗黙の記入枠だけを用いて出題すれば、何らヒントを与えることはない。このように、専ら出題意図によってだけ、生徒を答えに導くことができるため、高い教育効果を期待できる。

2.3 採点ヘルプ処理

ユーザ（生徒の指導者。ゆくゆくは生徒自身）によって、採点ヘルプ処理が起動される。採点ヘルプは認識処理を応用して自動採点処理を行ない、採点の手間を軽減するものである。インクデータが暗黙の記入枠によって設問に割当てられ、正誤判定処理が行なわれる。

正誤判定処理における、誤りは

(タイプ1) 正答を誤答とする

(タイプ2) 誤答を正答とする

の2通りがあり得る。このうち、タイプ2を排除することで、誤答と判定された設問だけについて、確認すればよいことになり、採点作業の手助けとなる。これが採点ヘルプ処理の目的である。

3. ドリル答案作成／答案ヘルパ プロトタイプ

図1にプロトタイプの画面例を示す。

本プロトタイプでは、まず算数ドリルを対象教材として、作成した。その理由は、

(1) 分数、等号で結ばれた式など、従来方式では入力が困難な解答形式を含むこと

(2) 使用される文字が数字と数個の数学記号に限られるので認識系の負担が少ないことである。

今回は、答案作成システムと、採点ヘルパの機能を一つのプログラムとして実現した。

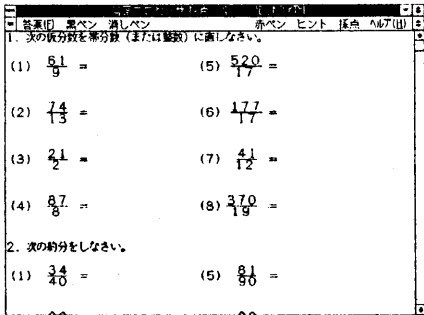


図 1 プロトタイプ画面例 1 (答案作成前)

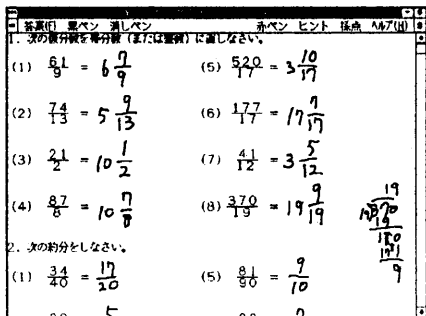


図 2 分数, 帯分数の答案例

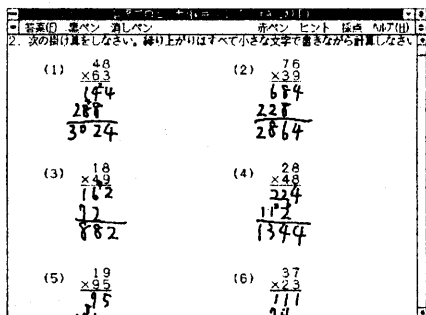


図 3 縦形掛け算の解答例

3.1 答案作成 UI

従来の紙上のドリルとまったく同じように画面
上のドリルに答案を書き込む。もちろん筆算の中
間結果なども許容される (図 2, 図 3)。

筆記には、鉛筆と消しゴムのメタファである、黒
ペンと消しペンの 2 つの記入モードを切り替えて
用いる。これの切り替えは画面上段のメニューを
タップ (ペンで 1 度触れること。マウスのクリッ

クに相当) することで行なう。黒ペンモード時に
は、画面上に黒でインキングされる。消しペン時
には、消しゴムのように、画面上のインクを消す
ことができる。実際は、背景色でインキングする
ことで、消えたように見せている。混乱を避ける
ため、モードによりカーソルの形が変るようにし
た。

画面の物理的大きさに制限があるため、設問数の
多いドリルでは、スクロールが必要である。プロ
トタイプでは、スクロール用トラックボールを用
意し、「紙をずらす感覚」でスクロール操作がで
きるようにした。もちろん、スクロールバーをタ
ップしてもよい。

3.2 算数の採点ヘルプ処理

採点ヘルプは図 4 に示す順に処理を行なう。

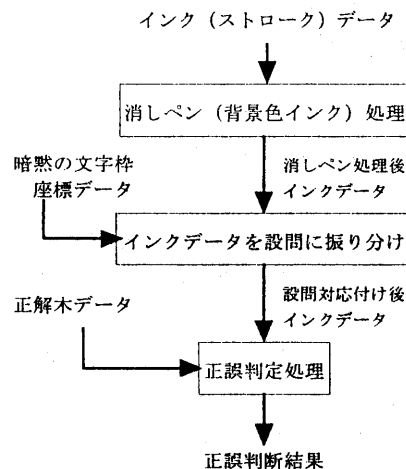


図 4 採点ヘルプ処理

以下、インクデータ中のペンダウンからペンアッ
プまで (いわゆる「画」に相当) をストロークと
呼ぶ。まず、消しモードによる背景色ストローク
(背景色のストローク) で、黒ストローク (黒ペ
ンによるストローク) を消去する処理である。黒
ストロークとそれより (時間的に) 後で書かれた
背景色ストロークとで、外接矩形が重なっている

組み合わせを探す。この組み合わせに関して、ビットマップ上に順にインキングを行なう（表示はしない）。黒ストロークを構成するサンプルポイントが背景色に消されているかを検査し、消されている場合はそのサンプルポイントを除去する。黒ストロークの中間に消されたサンプルポイントがある場合、黒ストロークを分割する。

背景色ストロークの処理後、黒ストロークを設問に振り分ける。各ストロークの重心（サンプルポイントの X,Y 座標値の平均）位置が、どの暗黙の記入枠（ドリル上の矩形領域）に入っているかによって、判断する。暗黙の記入枠のどれにも入っていないストロークは、無視される。

振り分けられた各設問で、正誤判定処理を行なう。プロトタイプの前誤判定処理では、専ら正答か誤答かを判断する。意味的な内容は用いず、予め用意された正答として解析可能かにより判断する。

3.2.1 解答形式

ドリル画面上の解答の配置を形式的に分類し、これを解答形式と呼ぶ。主に2次元的な配置を表わし、これを単位として、正誤判定処理を行なう。プロトタイプでは、次の解答形式を実現した。

- (1) 左から右へ横書きの数字列(負号を含む)
- (2) 分数（分数線の書き足しに対応）
- (3) 帯分数
- (4) 解答線（水平方向の直線）の下に書かれる解答（縦形筆算の掛け算、足し算用）
- (5) 余り記号（「…」）で結ばれた解答（縦形筆算の余り付き割り算用）
- (6) 「+」、「-」、「=」で結ばれた解答
- (7) 右側余白に解答の一部でないインク（余白演算など）がある解答

この他に、解答の一部を無視するための、ワイルドカードを用意した。認識不用な部分や、当面認識不能な部分について、これを用いる。

これらの解答形式を用いて、木構造に正解データを表現する(以下、正解木と呼ぶ)。例を図5に示す。大きな部分に渡る構造から順次細かい部分の

構造を表わすよう、木を構成する。算数ドリルの場合、木構造の末端は常に数字列(1)となる。その他の構造が枝別れを発生し、下位構造を含む。解答形式を組み合わせて正解木を構成する。

3.2.2 正誤判定処理

正誤判定処理では正解木をたどりながら、トップダウンに構造の解析を行なう。正解木のトップの形式を用いて、ストロークデータ全体を解析し、正解木の下位に割り当てるストロークを決定する。こうして順次、下位の形式にストロークを割当て、解析する。最下位の構造である数字列を認識するために、既開発の認識エンジン[6]を用いた。すべてが正解木の通りに解析されると、正解と判断される。

各形式の解析はルールベースで実現した。

分数の形式を例とする。分数の解析では、まず分数線を見つける。これは、外接矩形の縦横比で判定し分数線の候補とする。（書き足しに関する説明は略）。構造は大きなものからトップダウンに記述されるので、分数線の候補中で、一番大きなものを分数線とする。これが決定したら、その上下にストロークデータを分け、下位構造に渡す。

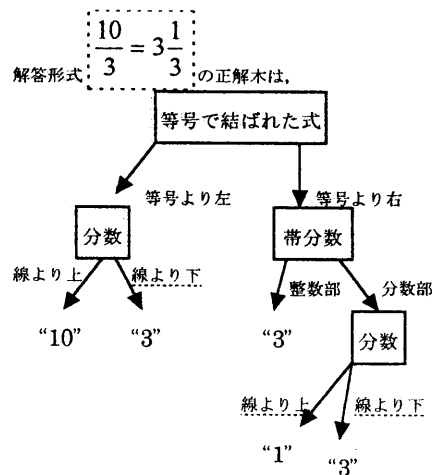


図5 正解木の例

数字列を認識するためには、まず、左右方向にノートする。数字は、たかだか2画なので、単独インク、および隣接するインクを組み合わせ、正解数字列として認識できるかを、判定する。判定結果例を図6、図7に示す。

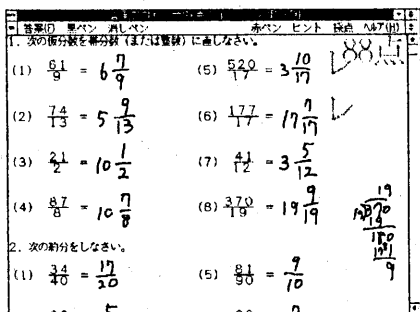


図6 採点ヘルパ処理例1

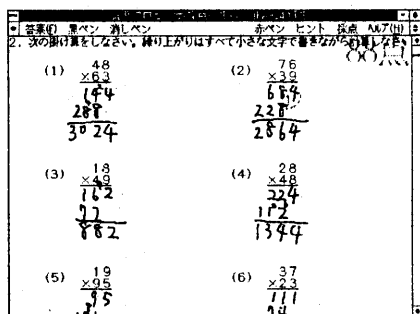


図7 採点ヘルパ処理例2

4. 試用実験

簡単な評価として、10名被験者に試用してもらい観察した。総問題数は118問で、内容は、筆算(掛け算、足し算、割り算)、通分、約分、1元一次方程式である。その結果、

- (1) 簡単な説明(ペンの選択方法、スクロール方法)後、即座に解答を始めることができた
- (2) ほぼ全員が30分で答案作成を終えたの2点が観察された。

まず(1)は、何れ紙上と違いないUIが自然に受け入れられた結果と言える。また、全員が集中力を

を持続させ、休憩もせず一気に問題を解いた結果が(2)に現れていると考えられる。

採点ヘルパの採点精度については、概して良好であるが問題の複雑さによって、採点精度が影響を受けることが確認された。改善の余地がある。

5. おわりに

本稿では、遅延認識と暗黙の解答枠による、ドリル答案作成/採点ヘルパシステムの思想を示した。算数ドリルを内容例としたプロトタイプを作成し、その具体的な内容を示した。

UIについては良好な結果が得られた。採点ヘルパについては、さらなる改良が今後の課題となる。プロトタイプでは、トップダウンの解析を行なっているが、この方法では一つの誤りが全体に波及する。柔軟なレイアウト解析を検討したい。

また、教科についても対象を広げてゆきたい。国語の書き取りなど、ペン入力ならではの題材が散見される。

参考文献

- [1] 栗原,他,“分数計算を指導する知的CAIシステム—手書きユーザインタフェースの試作—”,信学研報,ET94-112(1994)
- [2] 和田,“手書き数式入力システムにおける数式の認識戦略”,信学研報,ET95-31(1994)
- [3] 松本,他,“手書き数式入力システムと手書き数式エディティング”,信学研報,ET93-26(1993)
- [4] 村瀬,他,“オンライン手書き数式認識システム「METAH」の実現”,第46回情処全大,4H-5, pp.5-127(1993)
- [5] 曾谷,他,“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”,情処論,vol.34, No.1, pp.158-166(1993)
- [6] Toyokawa, et al. "An On-Line Character Recognition System for Effective Japanese Input", IEEE, Proc. of 2nd ICDAR, pp.208-203(1993)