

## 手書き認識による数式入力のための誤認識修正インタフェースの 設計と試作

徳田創<sup>†1</sup> レ・ドゥック・アイン<sup>†2</sup> 中川正樹<sup>†2</sup>

本稿では、手書き数式認識における修正インタフェースの設計と試作について述べる。数式入力には、数式記述言語による方法、数式エディタによる方法、手書き数式認識による方法がある。数式が入力できると、児童・生徒が選択形式で答えを選ぶのではなく、数式で解答する問題に利用でき、児童生徒の考える力を問うことができる。しかし、前者2つは児童・生徒が学習過程に利用するには困難があり、手書き数式認識には誤認識の問題がある。手書き数式認識は、シンボル分割、シンボル認識、そして構造認識からなり、それらのすべてで誤認識が発生しうる。誤認識を減少させ、誤認識が発生しても簡便に修正できるインタフェースが提供できると、児童・生徒はより自然に数式を入力できる。本稿ではシンボル分割修正、シンボル認識修正、構造修正のユーザインタフェースを提案し、その試作を報告する。

キーワード：インタフェース、手書き数式認識、オンライン認識、

## Design and Prototyping of Error Correction Interface for Inputting Mathematical Expressions by Handwriting Recognition

SO TOKUDA<sup>†1</sup> ANH DUC LE<sup>†2</sup> MASAKI NAKAGAWA<sup>†2</sup>

This paper describes design and prototyping of a user interface for correcting misrecognitions after inputting mathematical expressions by handwriting recognition. There are three major methods for inputting mathematical expressions, by mathematical script language, by mathematical formula editor and by computer recognition of handwritten mathematical expressions. If mathematical expressions can be input, students can answer questions by inputting mathematical expressions rather than selecting answers, which let them think more deeply and enable teachers to grasp student's true understanding and thinking ability. The first two methods mentioned above, however, have difficulty for students to use in learning process, especially in question answering process. On the other hand, computer recognition of handwritten mathematical expressions has the problem of recognition errors. It consists of symbol segmentation, symbol recognition and structure analysis, each of which may cause errors. Through reducing recognition errors and providing a user interface with which users can easily correct recognition errors even if they happen, mathematical expressions can be more naturally input. This paper proposes design and prototyping of a user interface to correct errors in symbol segmentation, symbol recognition and structure analysis.

Keyword: User Interface, Handwritten Mathematical Expressions, On-line Recognition

<sup>†1</sup> 東京農工大学大学院工学府産業技術専攻  
Department of Industrial Technology and Innovation,  
Tokyo University of Agriculture and Technology(TUAT)

<sup>†2</sup> 東京農工大学大学院工学府電子情報工学専攻  
Department of Electronic and Information Engineering, TUAT

## はじめに

近年、タブレットの低価格化と高速化により、普及が進み4人に1人が持つほどになった。これに伴い手書きでの入力の実用的になってきた。タブレットとスタイラスペンや指による入力は、キーボードとマウスのように習熟を必要としないため、紙と鉛筆のように使えるインターフェースとして研究されている。

数式の入力には、タッチインターフェースを用いた手書き入力に加え、キーボードを用いる数式記述言語[2]、数式エディタ[3]などで入力する方法がある。しかし、記述言語、あるいは、数式エディタではメニュー選択や操作の学習が必要になる。手書き入力による数式認識では、そのような学習が必要もなく、手書きのためキーボードに慣れていない生徒や児童にも使用可能である。

教育現場でのタブレットの導入も進んでいる。その際に算数や数学の課題の出題方式を選択式でなく、数式の筆記問題にすることができれば、児童・生徒の考える力や深い理解を問うことができる。そして、それを自動採点できれば、迅速に採点することができ、児童・生徒の記憶が新しいうちに採点をフィードバックできる。また、自動採点は、児童生徒の自習にも活用できる。この実現のためにも手書き数式認識が必要である。

当研究室では手書き数式認識を研究[1]しており、小中学生の数式を対象に認識エンジンの開発を行っている。認識率は85.15%である。まだ改良の余地があり、誤認識された際に正しく認識されるまで再度数式を手書き入力する必要がある。手書き文字認識では修正機能があり[4][5]、書き直さずに正しい認識結果に修正することができる。より長い数式であるほど数式の誤認識は起こりやすく、書き直すコストも高く手書き数式認識が実用的でない一つの要因である。したがって手書き数式認識では、誤認識の際に修正することが可能にすることでより実用性のある手書き数式認識システムを構築できる。

本稿では手書き数式による数式入力のための誤認識修正インターフェースの設計と試作を報告する。

## 2. インタフェースの設計

本章では誤認識修正インターフェースの設計について述べる。

### 2.1 認識の流れ

手書き数式認識の処理の流れには3つの処理がある。まず手書きで入力されたストローク（筆画）をシンボルという単位に分割するシンボル分割という処理である。シンボルとは数式を構成するすべての要素を指し、数字、記号、演算子などが含まれる。2番目の処理は、このシンボルという単位に分割された筆記ストロークを文字認識するシンボル認識という処理である。この認識されたシンボル間の位置関係、そして数式の文法を考慮して解析する構造解析を経て数式は認識される。この3ステップのいずれでも誤認識は起こる。

起こりえる誤認識の一覧は以下の3とおりである。

- シンボルの分割位置
- シンボルの誤認識
- 位置関係の誤認識

これらのすべてを修正することが可能なシンボル分割修正機能、シンボル認識修正機能、そして構造解析の修正機能が必要となる。

### 2.2 誤認識の修正機能の設計

各修正機能の設計について述べる。

#### 2.2.1 シンボル分割の修正

手書き数式を入力した際に誤認識された際に、入力した数式が正しく分割されていない場合に修正する機能である。

正しく分割がなされないかぎり、正しく認識ができる可能性が低いため、ユーザに分割が間違っていることをわかりやすく表示する必要がある。表示方法は図1のとおりである。



図 1 分割位置の表示

分割結果は分割された手書き入力ストロークの外接矩形で表示する。外接矩形を塗りつぶし、その上にストロークを表示する。後述するシンボル認識の結果をその上に表示し、各部分の視認性を上げるために赤、黒、緑の色を使用する。

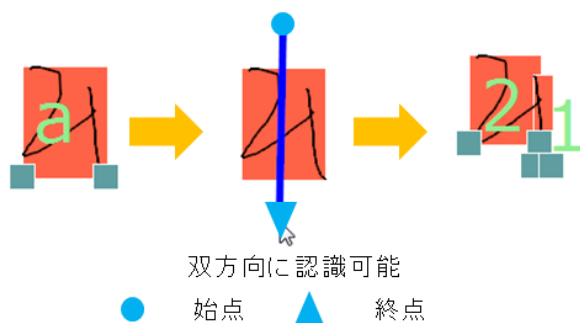


図 2 シンボル分割ジェスチャ

分割機能は認識の際に本来分割すべき場所で分割していない場所を分割する機能である。図 2 では、本来 21 と 2 つのシンボルを入力したものが 1 つのシンボルとして分割されている。分割機能を利用するには図 2 のように目的の矩形を縦に貫通するようにジェスチャを入力する。双方向にジェスチャは入力可能である。

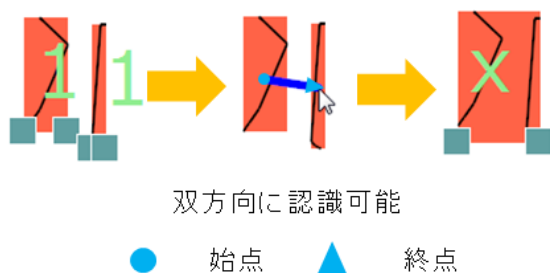


図 3 分割統合ジェスチャ

反対に分割統合機能は分割の際に分割しすぎてしまった場合に、非分割として分割されたシンボルを統合して 1 つのシンボルとする機能である。図 3 では 4 とシンボル 1 つを入力したつもりのものが、分割されてしまい 2 つのシンボルとして分割されている。

分割修正後には再認識を行う。シンボル認識、構造解析を行うことで数式文法的にも矛盾のない認識結果が出力される。このため、分割修正位置以外の認識結果にも変更

が起こることがある。

また、その結果再び同じ分割結果にならないように、分割の際には必ず分割、統合した場合には必ず非分割になるように分割スコアを変更する。

## 2.2.2 シンボル認識の修正

シンボルが誤認識された場合に修正する機能である。シンボル分割がすべて正しく分割されている前提での修正機能である。手書き入力されたストロークを正しく分割できているが、シンボルを誤認識していることをユーザにわかりやすく伝える必要がある。



図 4 シンボル認識結果表示

分割を表示する際に矩形を使う。その矩形内に認識されたシンボルを表示する。分割されたストロークと認識されたシンボルをリンクすることで直感的にどう認識されたかがわかる。また、外接矩形の下に、認識候補を切り替える時にタップする領域を表示する。

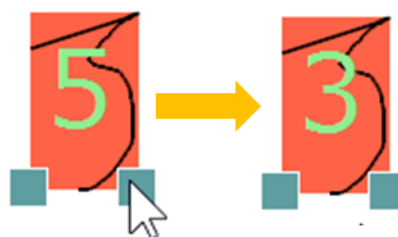


図 5 シンボル認識修正 次の候補

シンボル認識を行うことで最大 5 つの認識結果を得る。図 5 のように青い領域をクリックすることで認識候補を切り替えていくことができる。右の領域が次の候補、左の領域で前の候補に切り替える。

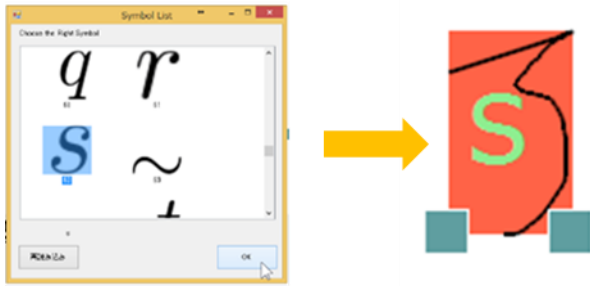


図 6 認識候補以外からの選択

認識候補は最大で5つあるが、候補に目的のシンボルがない場合がある。認識可能なシンボルは95種類ある。この複数のシンボル修正候補から簡単に選べるようなインターフェースが必要となる。図6のリストを作成し95種類のシンボルを表示する。そこから選ぶことで認識結果に反映される。このリストは5回以上右の領域をタップ、または第1候補時に左の領域をタップすることで表示する。修正された矩形のシンボル候補の第1候補は修正後に再び表示されないように尤度を0にする。

### 2.2.3 構造解析の修正

構造修正機能は、構造解析での誤認識を修正する機能である。構造修正機能はシンボル分割、シンボル認識で誤認識が起こっていないもしくは修正している前提で適用する機能である。

前述のシンボル分割、シンボル認識の結果は筆記画面で確認が可能である。修正内容も筆記面にすぐに反映される。しかし、構造の誤認識はシンボルの位置関係や文法による間違いとなり筆記画面での出力よりも認識結果と目的として数式の違いによる比較になる。修正後も数式の認識結果での確認となる。

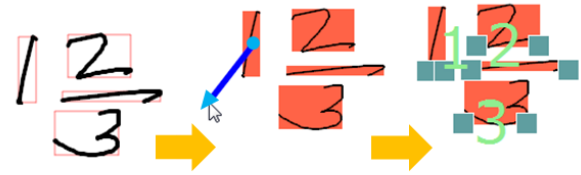
小中学生の扱う数式の中で起こりえる構造誤認識としては、構造的に特異な分数や累乗などの指数がある。他の数式表現では横に展開することが多いなか、これらは縦に展開する数式表現である。

ジェスチャ入力後に再認識により結果を出力する。しかし、文法にない場合は修正できない。

#### (1) 帯分数

分数の場合に起こりうるのは帯分数における構造的な

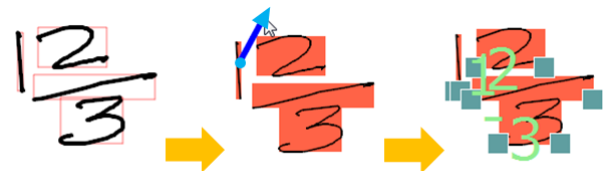
誤認識である。帯分数が仮分数として認識されてしまう、または仮分数が帯分数として認識されてしまう問題である。



● 始点 ▲ 終点

図 7 仮分数修正機能

仮分数の修正対象となる矩形は、分子における第1画である。第1画の矩形に対して中心から左下へジェスチャ(図7中央)を入力することで修正する。図7の修正画像では1に対してジェスチャを入力することで修正している。



● 始点 ▲ 終点

図 8 帯分数修正機能

図8左のように帯分数と誤認識されてしまった場合への修正ジェスチャ(図8中央)は帯分数の左の1画へ入力する。分子側へという、イメージが付きやすい中心から右上に入力する。

#### (2) 指数

指数における誤認識は前の数字と横並びに認識されてしまうことである。これは特に数字同士に起こりえる。

修正ジェスチャは、矩形の上下の辺に対してのジェスチャ入力を行っている。

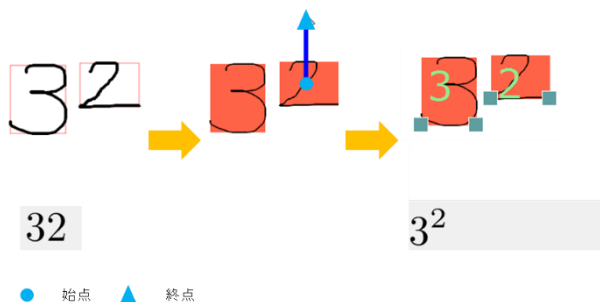


図 9 指数化のジェスチャ

図 9 のように 32 として認識されてしまった場合に、修正対称となるのは冪指数となるシンボルである。その矩形に対して中心から上の辺を通るジェスチャ（図 9 中央）に角度を付けずに真上へ入力することで修正する。

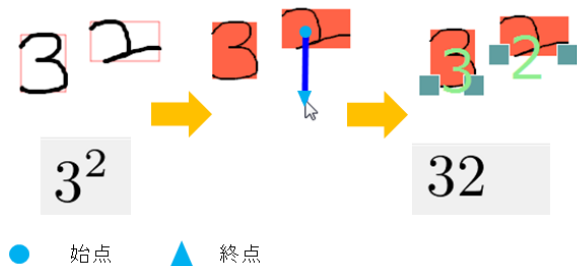


図 10 横並び化

反対に指数として誤認識された場合には、中心から下の辺を通るジェスチャ（図 10 中央）を入力することで修正する。

### 3. インタフェースの試作

インタフェースの試作について述べる

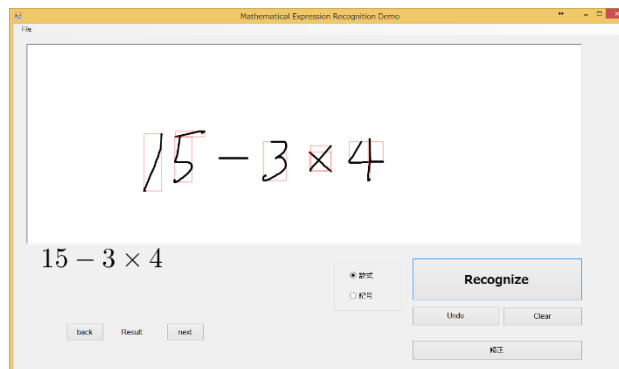


図 11 数式認識インタフェース

数式認識システムを利用し、Windows 環境で実装した。筆記画面を上認識結果やボタン類を下に配置する。

筆記面で間違えた場合にはボタン[Undo]を押すことで最新の筆画を 1 画消すことができる。ボタン[Clear]を押すことですべての筆画を消すことができ、書きなおす際に使用できる。

筆記画面で数式を入力後にボタン[Recognize]を押すことで認識する。認識すると筆記画面の左下に認識結果が出力される。認識結果の下のボタン[next][back]は認識結果の次の候補や前の候補に切り替えることができる。

修正を利用するには右下に配置されている修正ボタンを押す。修正ボタンを押すことで矩形や認識結果が筆記画面に出力される。図 11 では修正前の認識時のインタフェース画面である。修正前には修正機能で利用する矩形情報等は表示されない。

修正ボタンを押すことで修正ジェスチャの入力が可能になり、各修正機能を利用できる。修正後にボタン[Recognize]を押す必要はない。

### 4. おわりに

今回手書き数式認識における修正インタフェースの設計と試作について報告した。手書き数式認識の精度は開発途上のため、誤認識の問題がある。これまでは誤認識時に数式を書き直さなければならなかった。しかし、修正機能によって書き直すよりも簡単に修正が行えるようになれば、誤認識によるストレスが軽減され、手書き数式認識の実際の認識率以上の実用性が得られることが期待できる。

手書き数式認識を応用した自学自習のシステムの研究はいくつも行われている[7][8]. 修正機能を搭載することで、学習システムの開発の課題であった誤認識の対処を解決できる.

タブレットを用いた学習はこれからの一つの学習スタイルであるが、十分でない認識率によって本来の目的である学習の効率が損なわれることはあってはならない. 手書き数式認識の認識精度の向上とともに、修正機能の使い勝手を高めることでより頑健な認識システムとなる.

今後、修正機能の有効性を確かめるべく評価を行う予定である.

## 謝辞

当研究室の手書き数式認識班に深謝する.

## 参考文献

- 1) Anh Duc Le, Van Phan T, Nakagawa M, "A System for Recognizing Online Handwritten Mathematical Expressions and Improvement of Structure Analysis", 11th IAPR International Workshop on Document Analysis Systems, pp.51,55, 7-10 April 2014
- 2) 「LaTeX – A document preparation system」  
<https://www.latex-project.org/>, アクセス日時:2016年1月18日
- 3) 「Word 文書に数式を挿入する方法」  
<https://support.microsoft.com/ja-jp/KB/882544>, アクセス日時:2016年1月18日
- 4) 坂東 宏和, 福島 貴弘, 加藤 直樹, 中川 正樹: 枠なし手書き文字列認識における誤認識訂正インタフェース, 情報処理学会論文誌, Vol.43, No.6, pp.1996-2005 (2002.6).
- 5) 中川正樹, 依藤充範: ペンインタフェースによる自由形式文字列入力のための訂正ジェスチャ, ヒューマンインタフェースシンポジウム 2005 論文集, vol. 1, pp. 43-46 (2005. 9).
- 6) 森重湧太, 中川正樹: 手書き数式認識を用いた計算過程の正誤フィードバック, 情報処理学科研究報告, Vol. 2014-CE-126, No.3, pp.1-6(2014,10)
- 7) 千葉智史, 中川正樹: 手書き数式認識を利用したタブレット PC 上での数学 e ラーニングシステムの試作, 情報処理学会研究報告, Vol. 2014-CE-127, No. 10, pp.1-5 (2014.12).