

ペンストロークの時間間隔を用いた解答停滞箇所の検出

飯山 将晃*, 中塚 智尋**, 森村 吉貴***, 橋本 敦史****, 村上 正行****, 美濃 導彦*

Detecting Answer Stuck Point Using Time Intervals of Pen Strokes

Masayuki IYAMA*, Chihiro NAKATSUKA**, Yoshitaka MORIMURA***, Atsushi Hashimoto****, Masayuki MURAKAMI***** and Michihiko MINOH*

1. はじめに

近年, 学校現場における ICT 活用基盤の構築や活用が進められている. その中のひとつにタブレット PC の導入があり, これによって授業中, 家庭学習など場所を限らず学習ログデータを取得することが可能となった. 著者らも 2014 年より「京都 ICT 教育モデル構築プロジェクト」⁽¹⁾を実施しており, 現在, 中学 3 年生を対象に 3 クラス 120 名の生徒と教員にタブレット PC を配備して学習ログの取得を行っている.

タブレット PC を活用する際の利点として, 解答記述における時間や筆跡といった記述の過程を表す時系列データの学習ログが収集できることが挙げられる. 教師が適切な学習指導を行うためには, 生徒の習熟度の低い箇所を把握することが重要であり, 答案の解答過程を分析することになる. その際, 習熟度が低い箇所とは, 「誤りが生じる箇所」あるいは「時間がかかる箇所」のどちらかであると考えられるが, 時系列データの学習ログを分析すれば, 従来の紙と鉛筆による学習では行なうことのできなかつた「時間がかかる箇所

を把握することが可能となる. また, このような分析結果を学習者にフィードバックすることによって, 自分自身で誤りや弱点を把握し, 学習方法を振り返る手がかりにもなると考えられる.

本研究では, 教師の指導や生徒自身の学習の振り返りに活用することを目的として, 生徒がつまづいた部分を適切に把握できるようにするために, ペンストロークの時間間隔を用いた答案の解答停滞箇所の検出手法を提案する. 具体的には, 数学を対象として同一問題に対する複数の答案間で解答過程の同等箇所について時間比較を行うことにより, 停滞箇所を検出する.

従来, タブレット PC やデジタルペンを用いて入力された情報から, 文章の書き写し行動の時間分析⁽²⁾や解答停滞箇所の可視化⁽³⁾など, ストロークの時間間隔を利用した研究がある. これらの研究が他の筆記との対応をとらず分析対象となる答案のストロークのみを用いて停滞箇所の検出などの処理しているのに対し, 本研究では他の答案との比較によって停滞箇所を検出する. 他の答案との比較により, 「皆が解答に時間がかかる箇所」を停滞箇所として検出することなく, その生徒が苦手として解答に時間を要した箇所のみを検出

* 京都大学学術情報メディアセンター (Academic Center for Computing and Media Studies, Kyoto University)

** 京都大学大学院情報学研究所 (Graduate School of Informatics, Kyoto University)

*** 京都大学高等教育研究開発推進センター (Center for the Promotion of Excellence in Higher Education, Kyoto University)

**** 京都大学大学院教育学研究科 (Graduate School of Education, Kyoto University)

***** 京都外国語大学マルチメディア教育研究センター (Research Center for Multimedia Education, Kyoto University of Foreign Studies)

することができる。教師の生徒ごとへの指導や生徒自身の学習の振り返りに活用することを考えた場合、生徒全体の傾向ではなく生徒毎の苦手箇所を検出できる点で本手法は有効である。

2. 提案手法

2.1 ペンストロークデータ

本研究では学習ログデータの中でペンストロークデータを研究対象として用いる。ペンストロークデータの例を図 1 に示す。答案用紙には下書きのためのスペースも含めた解答過程を記述するための空白の欄が設けられており、生徒はその空白の欄内にスタイラスやデジタルペンを用いて解答を記述する。

ペンストロークデータにはある生徒がその空欄内に記述したすべてのペンストロークが時系列順に記録されている。ここで、「ペンストローク」とは記述における一筆のことであり、ペンダウンの時刻、ペンアップまでの経過時間、ペン先の二次元座標列で表現されるペン先が通った軌跡 (図 2) などからなる。このようなペンストロークデータからは、紙の答案から得られる静的な情報に加え、動的な解答の過程も得ることができる。

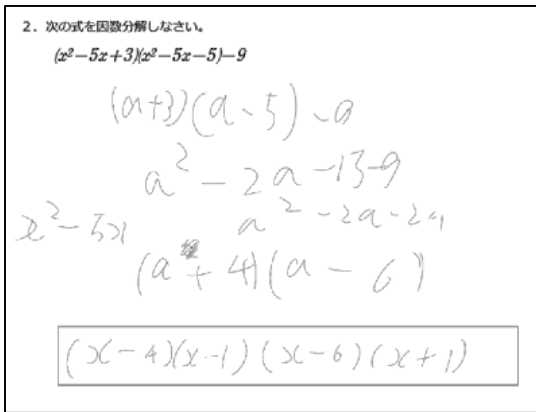


図 1: ペンストロークデータ

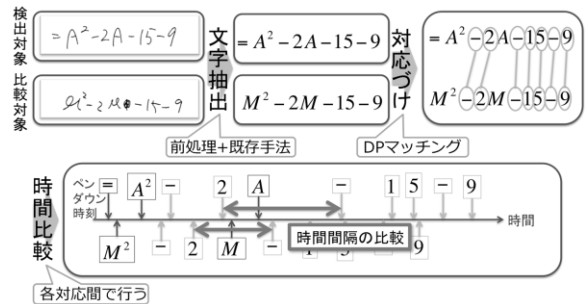
2015/11/21-13:55:19.896 MODE_ANSWER_STYLUS;FF000000;1,1 2 292,380;293,3AE;294,3AC;294,3AB;296,3AB;299,3AG;29D,3AA;2A0,3A2;2A3,3A0;2A7,39F;2AA,39E;2AD,39E;2B0,39E;2B3,39F;2B5,3A1;2B8,3A3;2BA,3A5;2BB,3AB;2BC,3AB;2BC,3AC;2BB,3B1;2BA,3B5;2BB,3B6;2B7,3B6;2AD,3CC;2A9,3D2;2A5,3B8;2A1,3DE;294,3ED;28B,3FA;280,401;27F,402;27D,404;27C,405;27C,406;27E,406;281,405;283,404;298,3FE;284,3FA;2C9,3F7;2D6,3F5;2D8,3F4;2D8,3F3;2D0,3F2;2Df,3F1;2E0,3F0;2E1,3EE;2E3,3EC 242
2015/11/21-13:55:22.777 MODE_ANSWER_STYLUS;FF000000;1,1 2 555,390;554,394;553,390;552,39C;551,3A5;550,3AD;54F,386;54C,3D2;54A,3E9;549,3F6;549,3F7;549,3F8;549,3F9;549,3F9;549,3F8;54B,3EA;54C,3E4;54F,3D0;552,3D7;554,3D4;556,3D2;559,3D6;55C,3CF;55F,3CE;562,3CF;564,3D6;566,3D2;567,3D6;566,3D9;565,3D3;563,3Df;55F,3E3;55B,3E6;557,3E9;552,3EC;54D,3F0;546,3F3;548,3F6;53C,3F7;538,3F7;535,3F8;534,3F6;535,3F4;537,3F3;538,3F2;538,3F0;53D,3EE;53E,3ED;548,3ED 1D1

図 2: ペン先の座標列データ

2.2 手法の概要

このようなペンストロークデータから得られる生徒の解答過程より、解答の停滞が見られた箇所を自動的に検出する。これにより、習熟度が低い箇所を把握できるようになることが期待できる。解答の停滞が見られた箇所に基づいて生徒の理解度や思考の過程を分析することを想定した場合、単に解答に時間を要した箇所を検出するのではなく、答案を他の複数の生徒の答案と比較することによって、相対的な解答時間の遅れを検出できることが重要である。その際、この比較は答案間における解答過程内の同一ステップにあたる箇所 (同等箇所と呼ぶ) ごとに行われるべきであり、答案間の同等箇所がペンストロークデータ上で対応している必要がある。これを実現するために以下の手順にて答案の対応付けと解答停滞箇所の検出を行う (図 3)。

1. ペンストロークデータからの文字列抽出: 生徒間で筆跡の違いが存在する場合にもロバストに対応付けが可能となるよう文字認識を行い、答案を表す文字列を生成する。
2. DP マッチングによる答案文字列同士の対応付け: 抽出された文字列には認識誤りが含まれ、また、同等箇所でも生徒ごとに異なる数式の表記が用いられている。これらの違いに対してロバストになるよう DP マッチングによる対応付けを行う。
3. 解答停滞箇所の検出: 対応づけられた同等箇所ごとに、記述に要する時間の生徒間の平均値と各生徒の平均からの乖離を算出し、これを可視化する。



複数の比較対象に対して行い、遅れの平均を算出平均が一定以上の箇所を解答停滞箇所として検出

図 3: 手法の概要

以降、答案 A のペンストロークデータを S_A と表記し、 N_A 個のペンストローク $s_{1A}, \dots, s_{N_A A}$ から構成されているものとする。ここで、 s_{iA} はペンストロークデー

タ S_A の*i*番目のペンストロークであり、文字や記号の一面に相当する。各ペンストロークは軌跡およびペンダウン時刻からなり、ペンストローク s_{iA} の軌跡を p_{iA} 、ペンダウン時刻を t_{iA} と表記する。軌跡は点座標の順序付き集合として、

$$p_{iA} = \{(x_{iA}^{(1)}, y_{iA}^{(1)}), (x_{iA}^{(2)}, y_{iA}^{(2)}), \dots, (x_{iA}^{(n_{iA})}, y_{iA}^{(n_{iA})})\}$$

と表す。ここで n_{iA} は軌跡 p_{iA} を構成する点の数である。

2.3 ペンストロークデータからの文字抽出

本研究で扱うペンストロークデータは数学の答案を表すものであるが、数式は一般的な文章と異なり、文字が左から右に整列しているだけでなく、文字が上下に配置されている場合もある。その一方で、数学の答案は式が上から下に行ごとに展開されていく。そのため、文字抽出の前にまずペンストロークデータを一行分ずつに分割する。

ペンストロークデータは時系列データであり、ペンストロークが書かれた順に記録されている。先頭の時刻から順にペンストロークの座標を走査し、ストロークの座標の変化に基づいて改行を検出する。これによりペンストロークを行単位に分割する。なお、座標系を図4のように定める。

改行には図4に示す2つの場合が考えられる。図中の①では改行前後でストロークのx座標の値が改行前の行の幅だけ減少し、y座標の値が文字の縦幅だけ増加する。図中の②では改行前後でx座標の値が段落間の幅だけ増加し、y座標の値が何行かの高さの分だけ減少する。つまり改行時には、毎回多少の違いはあるが、同じ行を書いている間には見られないような、一定以上の変化がストロークのx座標y座標共に生じることがわかる。この変化量は答案中の文字の大きさに依存しているため、まずペンストロークデータから文字の大きさを算出し、それを改行の検出に用いる。

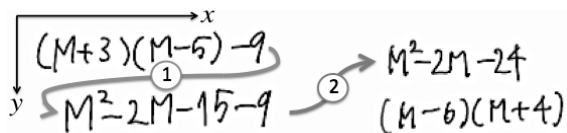


図4：改行

文字の大きさを算出する際、各ペンストロークのx

座標幅、y座標幅を用いて中央値を算出してこれを文字の大きさとし、改行の検出に用いる。

$$\text{width}_i = \max(x_{iA}^{(k)}) - \min(x_{iA}^{(k)})$$

$$\text{height}_i = \max(y_{iA}^{(k)}) - \min(y_{iA}^{(k)})$$

fontsize

$$= \text{median}(\text{width}_i, \text{height}_i, \dots, \text{width}_j, \text{height}_j)$$

これにより、ペンストロークに含まれる点やマーカなどの線、または誤入力といった文字以外を表すものに影響されず文字の大きさを求めることができる。

答案Aのある行が、ストローク s_{iA} から始まり s_{jA} までを含むことが既に判定できているものとする。このとき、次のストローク $s_{(j+1)A}$ もこの一行に含まれているか、あるいは改行を挟んでいるのかどうかを以下の条件で判定する。

$$X_{\max} = \max(\max(x_{iA}^{(k)}), \dots, \max(x_{jA}^{(k)}))$$

$$Y_{\min} = \min(\min(y_{iA}^{(k)}), \dots, \min(y_{jA}^{(k)}))$$

$$x_{\min} = \min(x_{(j+1)A}^{(k)})$$

$$y_{\min} = \min(y_{(j+1)A}^{(k)})$$

とおき、

$$|X_{\max} - x_{\min}| > \text{fontsize}$$

かつ

$$|Y_{\min} - y_{\min}| > \text{fontsize}$$

を満たすとき s_{jA} と $s_{(j+1)A}$ の間には改行があると判定する。

2.3.1 分割されたペンストロークデータからの文字抽出

分割されたペンストロークデータから文字抽出を行う。文字抽出には Alvaro らが提案した手書き数式認識手法(SESAT)⁽⁴⁶⁾を用いる。数式の認識には大きくわけて、記号の認識と数式の構造解析の二つの行程が必要になる。数式の構造は使われている記号に依存するので、"SESAT"では記号の認識結果と数式の構造解析結果を組み合わせるとき最適となる結果を選択して出力している。ペンストロークデータを入力に与えると、その認識結果の文字列と各文字を構成しているペンストロークの座標情報が得られる。前節の手法により、ペンストロークデータは複数の行に分割されて

いるが、次節でのペンストロークデータの対応付けでは、分割されたペンストロークデータから得られた文字抽出結果を連結させて、ひとつの文字列として処理を行う。

2.4 ペンストロークデータの対応付け

比較するふたつの答案から得た文字列を、DP マッチングにより対応付ける。本研究では一般的なDP マッチング手法⁶⁾に、数式のマッチングに適した重み付けを加えて用いる。

ふたつの答案から得た文字列 $text_1$, $text_2$ から図 5 に示されるようなペナルティ表を作成する。ペナルティ表内の各ペナルティは左から右、上から下の順に計算され、文字列 $text$ の i 番目の文字を $text(i)$ と表すと、ペナルティ表の i 行 j 列のペナルティ $P(i, j)$ は、次式に従って算出される。

	a	b	4	c
b	5	7	19	26
4	17	16	8	20
c	24	23	20	9

図 5: ペナルティ表

$$P_w(i, j) = \begin{cases} 0 & text_1(i) = text_2(j) \\ \alpha & text_1(i) \neq text_2(j) \text{ (共にアルファベット)} \\ \beta & \text{それ以外} \end{cases}$$

$$\begin{aligned} left(i, j) &= P(i-1, j) + 2 \\ up(i, j) &= P(i, j-1) + 2 \\ leftup(i, j) &= P(i-1, j-1) + 1 \end{aligned}$$

として、

$$P(i, j) = P_w(i, j) +$$

$$\min(left(i, j), up(i, j), leftup(i, j)) \quad (1)$$

式1では、ふたつの文字が異なっていた場合のペナルティを設定している。一般的な文字列同士のマッチングをDP マッチングによって行う場合、ふたつの文字が同一であればペナルティを0に、異なっていれば文字種にかかわらず一定値のペナルティを与える。それに対して、答案の数式を表す文字列同士のマッチングを行う場合、ふたつの文字が'x'や'a'など、異なるアルファ

ベットだったとしても、両者は表現方法が異なるだけで意味的には同じという可能性がある(例えば、 $x^2 + 1 = 0$ と $a^2 + 1 = 0$ など)。一方、ふたつの文字がxと1のようにアルファベットと数字であるような場合は意味的にも異なると判断される。以上のことを考慮して、ふたつの文字が同一である場合、異なるが共にアルファベットである場合、異なりかつ文字種も異なる場合の3つの場合に対してそれぞれ異なるペナルティを設定している。

また、式では上下および左右の移動経路はどちらかの文字列の伸縮を表しており、文字列を伸縮させないなめ移動の経路よりも大きいペナルティ($left(i, j)$ と $up(i, j)$ は+2, $upleft(i, j)$ は+1)を設定している。

図 5 に示したペナルティ表は、2 つの文字列"ab4c","b4c"に対して、 $\alpha = 5, \beta = 10$ としたときの例である。このように作成したペナルティ表の最下段のうち最小値のマス(最小値が複数ある場合、その最右マス)から、左、左上、上のうち最小値を辿って最上段の最左マスに至る経路を得る。経路が (i, j) のマスを通り $text_1(i) = text_2(j)$ であるとき、この両文字は対応付いていると判定する。この文字間の対応関係から答案間の対応付けを得る。

2.5 答案間の同等箇所の対応付けと解答停滞箇所の検出

ペンストロークの対応付けに基づいて答案のストロークごとに解答時間の比較を行う。答案Aの各ストロークについて他の答案での解答時間の平均からの遅れ d_{iA} を算出する。

答案Aの m_i 番目のストローク s_{m_iA} と答案Bの m'_i 番目のストローク $s_{m'_iB}$ が対応付き、答案Aの m_k 番目のストローク s_{m_kA} と答案Bの m'_k 番目のストローク $s_{m'_kB}$ が対応づいているとする。答案Aにおける、 s_{m_iA} から s_{m_kA} までの筆記時間を $T_A(m_i, m_k)$ 、答案Bにおける $s_{m'_iB}$ から $s_{m'_kB}$ までの筆記時間を $T_B(m'_i, m'_k)$ とすると、答案Bに対する答案Aの m_i 番目から m_k 番目までの各ストロークの遅れは、

$$d_{iA}^B = \frac{\max\{T_B(m'_i, m'_k) - T_A(m_i, m_k), 0\}}{m_k - m_i + 1}$$

で与えられる。ここで、 $m_i \leq j \leq m_k$ である。

答案Aの比較対象として用いる答案 B_1, \dots, B_N を用

いて同様に各ストロークの遅れを計算し、次式にて遅れを算出する。

$$d_{iA} = \frac{1}{N} \sum_{b=1}^N d_{iA}^{Bb}$$

2.6 解答停滞箇所の提示

教師に解答停滞箇所を提示する手段として以下の2つの提示方法を考案した。提示にあたっては、答案から解答停滞箇所が視覚的に把握できることが重要である。

ペンストロークデータから答案を再現することができるが、このときペンストロークを直前のストロークの遅れ d_{iA} に応じて着色する。解答停滞箇所が色によって確認できる上に、それ以外の部分、例えば人よりも素早く進めた箇所などを確認することもできる。

また、ペンストロークデータには解答過程の時間情報も含まれているため、その解答過程を動画再生することも可能である。ふたつの答案を、対応付いている箇所ごとにタイミングを同期させながら再生することで、各箇所でもちがいがどれだけ停滞しているかがわかる。各箇所にかかった時間をそれぞれの答案ごとに棒グラフで表示する。棒グラフを見ながら、解答停滞箇所として検出されたペンストローク付近の箇所には実際どれくらいの時間がかかっているのか、またその時間は他者の答案と比べてどうなのかをより詳細に見ることができる。図6に提案する提示法による動画の1フレームを示す。

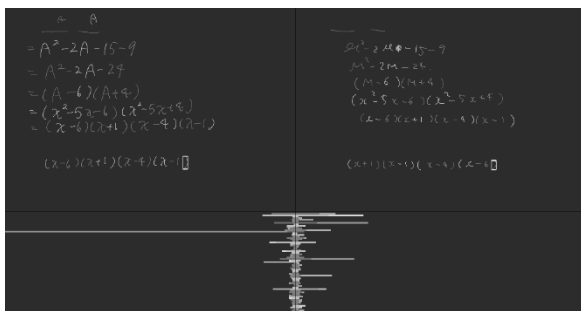


図6：動画による解答停滞箇所の可視化

3. 実験

3.1 ペンストロークデータの収集

提案手法を実際の教育現場で得られたペンストロークデータに適用する実験を行った。用いたデータは中学校3年生86名が数学の因数分解を中心とした課題

7題に解答した際に得られたものである。この602個のデータの中から消しゴムとしてのストロークが含まれたものを除き、211個の答案データを用意した。

なお、実験にあたっては、氏名・出席番号等個人を特定する情報を削除して匿名化処理を行ったものを利用した。

3.2 文字抽出と対応付け結果

ペンストロークデータからの文字抽出結果を図7に示す。74文字中10文字の抽出に失敗していることが確認できる。以降の実験では211個の答案のうち、同様の解答ステップを経たと考えられ、かつ、7割以上の対応がある比較対象答案が存在する93個の答案を実験に利用する。なお、各答案に対する比較答案は最小1個、最大36個、平均14.2個である。93個の検出答案に対して平均14.2回比較を行うことから、比較の組は1320組存在する。このうち、答案の内容がほぼ一致していた組と、一部の式の項の順序が入れ替わった組の2組について対応付けの精度を調べたところ、適合率、再現率が前者は94.6%、89.7%、後者は76.6%、73.5%であった。前者の対応付けの結果の例を図8に示す。

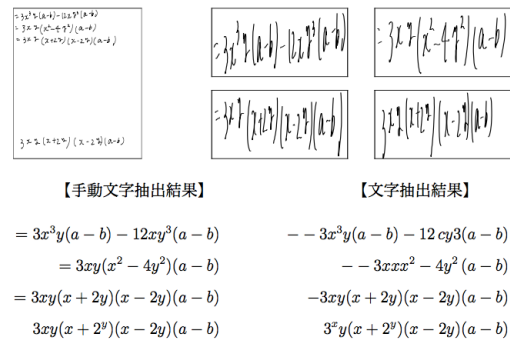


図7：文字抽出結果

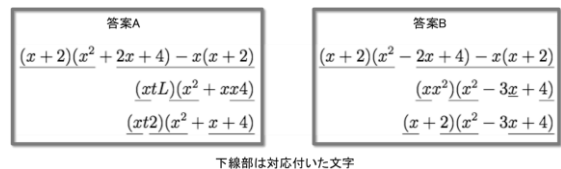


図8：対応付け結果

3.3 解答停滞箇所の検出結果

解答停滞箇所の検出閾値は、人目で停滞と感じられ

る長さを元に 2 秒と定めた。つまり全答案の平均より 2 秒以上遅れている箇所を解答停滞箇所として検出する。この結果、93 個の答案のうち 14 個の答案から計 298 箇所の解答停滞箇所が検出された。このときひとつの答案からは最大で 93 個検出された。検出結果を図 9 および図 10 に挙げる。直前のストロークが全答案の平均より 2 秒以上遅れている箇所を解答停滞箇所として赤く、遅れが 0 に近づくほど青く着色している。図 9 において、 $(a + b)(a^2 - ab + b^2) = (a^3 + b^3)$ の公式を利用する際に他の答案よりも時間を要したことが確認できる。また図 10 において、式展開後に項ごとにまとめる計算に時間を要し、また中かっこを記述する際に少し時間を要していることが確認できる。

なお、文字抽出から対応付け、停滞箇所の可視化は 93 個の答案に対し 5 分程度で処理可能である。

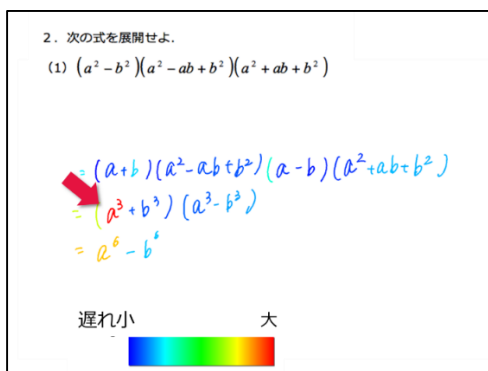


図 9：解答停滞箇所の検出結果(1)

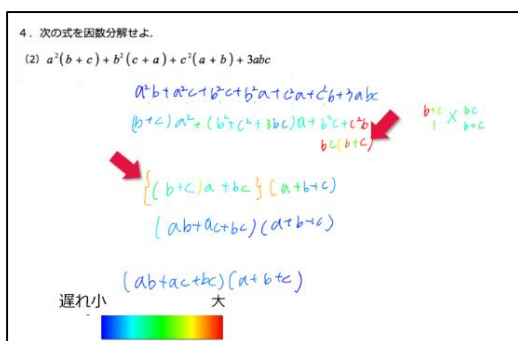


図 10：解答停滞箇所の検出結果(2)

4. おわりに

本研究では、教師の指導や生徒自身の学習の振り返り

に活用することを目的として、生徒がつまづいた部分を適切に把握できるようにするために、ペンストロークの時間間隔を用いた答案の解答停滞箇所の検出手法を提案した。

提案手法で対応可能なものは計算問題のように数式や文字のみで構成され、ある程度答案のパターンが限定された答案であり、数式の証明問題のようなものには対応可能であるものの、図形問題のように図を描画したり書き込みを行ったりするような問題には対処できない。今後の課題としてこれらの問題に対応できるように画像特徴など文字列以外の特徴を用いた答案の対応付けなどに取り組む必要がある。

参考文献

- (1) マイクロソフト：“京都 ICT 教育モデル構築プロジェクト”
<https://news.microsoft.com/ja-jp/2015/06/02/150602-education/> (参照 2016.5.31)
- (2) 平林 ルミ,河野 俊寛,中邑 賢龍：小学生の視写における書字行動プロセスの時間分析, 特殊教育学研究 48(4), 275-284 (2010)
- (3) 浅井洋樹, 野澤明里, 苑田翔吾, 山名早人：オンライン手書きデータを用いた学習者のつまづき検出, DEIM Forum 2012 A8-4 (2012)
- (4) Alvaro, F., Sanchez, J.-A. and Benedi, J.-M.: Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks, Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, pp.2944-2949 (2014).
- (5) Alvaro, F., Sanchez, J.-A. and Benedi, J.-M.: An integrated grammarbased approach for mathematical expression recognition, Pattern Recognition, Vol. 51, pp. 135-147 (2016).
- (6) Baeza-Yates, R., Ribeiro-Neto, B. et al.: Modern information retrieval, Vol. 463, ACM press New York (1999).