CenterBand を用いた数式構造解析の安定化

村上 玄生† 鈴木 昌和‡

†九州大学大学院数理学府 ‡九州大学大学院数理学研究院
〒812-8581 福岡市東区箱崎 6-10-1
E-mail: Suzuki@math.kyushu-u.ac.jp

あらまし
昨年発表された江藤‐鈴木による数式構文解析は、文字認識の影響を受けにくいものであったが、まだ十分ではなく文字の誤認識があった場合に大きく数式構造をとり間違うことがある。そこで、文字認識の結果を統計的にしか使わない「Center-Band」を利用する方法を加えたことで、文字の誤認識の悪影響を吸収し構造解析の精度の更なる向上が見られたので、その方法と結果を報告する。

キーワード OCR, 数式認識, Center-Band, ネットワーク

Improvement of Mathematical structure analysis by Center-Band

Makoto MURAKAMI†, and Masakazu SUZUKI‡

† Faculty of Mathematics, Kyushu University
‡ Graduate School of Mathematics, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581 Japan
E-mail: suzuki@math.kyushu-u.ac.jp

Abstract
The mathematical formula recognition presented by “Eto and Suzuki” last year is robust against errors of the character recognition to some extent. However, there are some cases causing a fatal error of structure analysis. We propose a method by Center-Band obtained by using the result of the character recognition only statistically. It considerably improves the structure analysis. We report our new method and the result of the experiment.

Key words OCR, Mathematical formula recognition, Center-Band, network

1. はじめに

近年、自然科学分野において学術雑誌の電子 Journal への移行が急速に進んでいる。しかし、科学技術分野の文献では数式が含まれているため、電子化には多大なコストがかかる。科学的分野の論文誌、書籍の電子化に要するコストを軽減するために、数式を含む文章に対応できる OCR システムの開発の重要性が増している。

1.1 数式構造解析の現状

これまでにも数式認識の研究は行われているが、実行速度と認識精度の両方において実用化に適したアルゴリズムとして発表された最初のものは岡本の手法（[1][2]）と、Fateman（[3]）によるものであると考えられる。特に、岡本による数式認識の研究は、多くの文献に引用され、実用化の可能性を持つ手法として国際的によく知られている。[4][5]による手法も、岡本や Fateman と同様にトップダウンの解析手法になっており、その流れをくむものである。しかしながら、これらの手法では文字の誤認識や大きさの異なる類似文字等による影響を受けやすく、局所的な誤認識が数式の全体の構造認識を大きく崩してしまうことが少なくない。精度が向上しているとはいえ OCR において文字の誤認識は避けられない。そのため、文字認識の結果にあまり依存しない安定した解析手法が求められる。現在、鈴木研究室では昨年発表された江藤－鈴木手法（[6][7]）を用いて、文字認識の結果にあまり依存せずに数式構造解析を行うよう工夫している。これによってかなり精度の高い解析を行うことに成功しているが、未だ大きく間違うこともあるのが現状である。

数式構造解析を誤る主たる原因としては以下の二点がある。一点は、文字認識を誤り、候補中に正解と同じ大きさの文字がない場合である。この場合はほとんど間違ってしまう。もう一点は、文字認識が正しくてもサイズや中心位置の関係が後で述べる散布図の正しい領域内になかった場合である。どの領域にもなかった場合、どのような親子関係を結べばよいか分からないので強制的に水平に接続しているため、本来添え字関係であるならば構造を誤ってしまう。また散布図の誤った領域にしかなければ必ず間違ってしまう。

1.2 今回提案する手法

これらの構造解析の誤りの中には、本来は構造をもたない単純な数式部分や更にはテキスト中の文字であるにもかかわらず添え字構造を持ってしまうものも多くある。これらは、接触・分離文字を含め大きさの違う文字に誤認識された文字やスモールキャピタル、類似文字（c,C や s,S 等、大きさは違うが形が同じため一文字認識では区別がつかない文字）が含まれているものがほとんどであり、認識結果を元にその行における大きさや中心位置を比較すると添え字と判断されてしまうことがあるからだ。

このような添え字でない通常の文字を添え字と誤ってしまう誤りは、複雑な数式における添え字上での誤りと比べ人間から見れば明らかである。しかしなぜ人間は明らかに間違っていると判断するのか。それを"Center -Band"というものを用いることで説明できると考えた。

図 1：Center-Band

Center-Band とは一行中の文字のアセンダー部やディセンダー部を除いたセンター部分をちょうど覆うような幅の水平な帯（図 1）のことをいう。ベースライン上の文字・記号類のほとんどはこの Center-Band を覆っているかその中に入っている。ベースライン上の文字とは、添え字位置でない通常の位置にある文字のことを言う。また逆に添え字上の文字・記号類が Center-Band を覆うようなことはほとんどない。人間は Center-Band を覆っている、または Center-Band に覆われているものをベースライン上にあると判断し、そうでないものを添え字と判断しているのではないかと考えた。

このことを利用し、Center-Band 上の文字は親文字には添え字としてはリンクしない、Center-Band 上の文字が添え字位置にあった場合にペナルティを与える、という条件を従来からの手法に加えることでベースライン上の文字・記号類が添え字となってしまわないようにし、数式構造解析の向上を図った。

ここで重要なのは Center-Band を取得するには英数字、ギリシャ文字の文字認識結果を統計的に用いるが、個別の文字間の水平・添え字判定には文字認識結果を使わないということである。それにより誤認識から受ける影響が抑えられるように工夫した。

これに関しての実験を行ったので、その実現方法

と実験結果を報告する。

2. 数式構文認識の概要

本節では前提となる数式構文の認識手法として、仮想リンクネットワークを用いた手法（[6][7]）について述べる。

2.1 前提条件

数式構文認識を行う際に使用する正規化サイズ、正規化中心、またそれを利用した(*H*,*D*)を定義する。

今回使う一文字認識エンジンは、複数個の候補を返し、その候補毎に類似度を与えるものとする。

また点類・アクセント記号類は最初は取り除いて構文認識を行い、全ての処理が終わったあとに適切な場所に挿入する。

正規化サイズ(Nsize)

文字はアセンダー部やディセンダー部を持つなど文字種によって大きさが違う。そのため一列に並んでいる文字列でも外接矩形のみからでは、でこぼこに並んでいるように見える。そのため同じラインに並んでいる文字は同じ大きさを持つように補正する必要がある。その補正された大きさを正規化サイズと呼ぶ([1])。ここでは、図 2 に示すアセンダー部分 *x* とディセンダー部分 *z* をあわせたサイズ *x*+*y*+*z* とする。ただし *xyz* 比は 28:51:21 とする。これは 32 冊の雑誌、教科書から取得したデータの平均値である。また、*xyz* の区別が明確でない記号については、文字の高さをそのまま正規化サイズとする。ただし、演算子については高さと幅の大きいほうを正規化サイズとする。

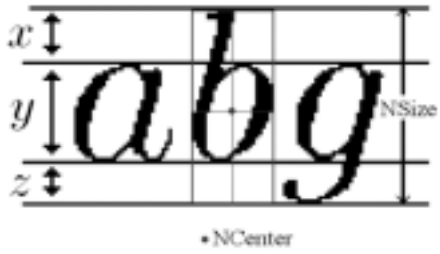
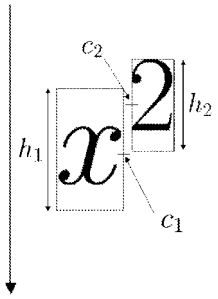


図 2：*x*,*y*,*z* と正規化サイズ・正規化中心

正規化中心(NCenter)

また文字の中心に関しても同様の理由で隣り合う文字の外接矩形の中心の高さが異なってしまう。それを同じ高さの中心位置を持つように中心位置を補正したものを正規化中心と呼ぶ([1])。ここでは、正規化した矩形の中心を正規化中心とした。



$$H = \frac{h_2}{h_1} \times 1000$$

$$D = \frac{c_1 - c_2}{h_1} \times 1000$$

図 3：サイズと中心位置の比較

このを計算し、実際に測定した値を散布図（図 4）に示す。この散布図は文字種ごとに作成した。ほとんどの場合は(*H*,*D*) をこの散布図に当てはめることで、隣接する二つの文字の関係が水平、上添え字、下添え字のうちいずれの関係かを区別できる。

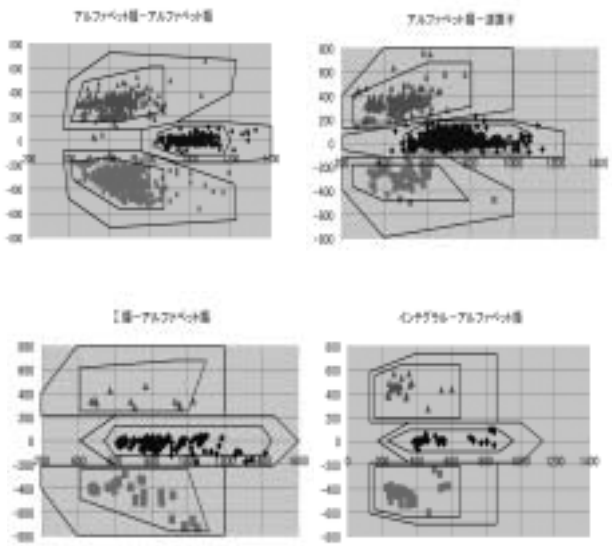


図 4：サイズと中心位置の散布図

2.2 同一ライン判定

数式領域においてさまざまな大きさと位置をもって並んでいる文字・記号を、同じサイズ・中心位置をもつと考えられる文字・記号のクラスに分類することを同一ライン判定と呼ぶ。ここでは、数式領域中のすべての認識候補を対象として同一ライン判定を行い、サイズとラインのラベルをつける。

同一ラインクラスへの分類には前提条件で述べた水平の関係にある文字どうしの散布図を用いる。各同一ラインクラスには基準の正規化サイズと中心をもたせ、それと判定しようとする候補文字の(*H*,*D*)を計算する。(*H*,*D*) が図に示した狭領域に入るときに

同じクラスとし、どのクラスにも入らないときに新しいクラスを作る。ただし、標準文字サイズより大きな括弧類・矢印・分数線に対しては、中心位置 D のみで判定をおこない、D の値が $-100 \sim 100$ のときに同一ラインと判定する。

判定後、同じクラスの文字に同じラインラベルをつける。また、クラスごとの基準の正規化サイズが小さい順にサイズラベルをつける。サイズラベルは異なるクラスであっても正規化サイズが近いクラスには同じラベルをつけるようにする。

2.3 ネットワークの構成

数式中の文字や記号のあいだに親子関係を定めることができると考え、数式全体を向きを持った木として表現する。その実現方法として、文字や記号を頂点（ノード）とし、可能性のあるすべての親子関係を有向辺（リンク）として持つネットワークを構成する。

ノードは外接矩形と認識結果の候補を持っており、リンクは（親候補、子候補、接続の種類、リンクコスト）の組を表す。このリンクは各ノードの第一候補と類似度の高い候補に対し(H,D)から散布図を用いて接続の種類、コストを算出して作る。成分が一つでも違えばそれは異なるリンクとみなす。また、c,C や s,S のような文字は一文字認識では区別がつかない。そのため必ず正規化サイズが異なる同形の文字は数式認識候補に加えるようにしている。

2.4 無矛盾数式構文木の取得

構成したネットワークから数式構文として矛盾のない全域木を取得する。その一つ一つを無矛盾数式構文木と呼ぶことにする。ここで矛盾のない数式構文木とは、各矩形に対し認識結果がひとつに定まった連結な木で、矩形はそれぞれ各接続方向に高々１つの子を持ったものである。ビームサーチを用いてリンクコストの合計が低い無矛盾数式構文木を複数個求める。

2.5 大域的コストによる再評価

取得した無矛盾数式構文木に対して各矩形のリンクコストの和を木のコストとし、それに数式構文木全体の構造を反映させた大域的なコスト付けによって再評価を行い、その結果、コストが最小のものを最適な数式構文木とし、それを認識結果とする。

コスト付けは以下の条件である、まず各矩形 K に対し添え字領域の部分木の最大サイズラベルが K のサイズラベル以上のときコストを上げる。ベースライン上の候補文字と同じラインラベルの文字が添え字領域にあるときコストを上げる。ベースライン上のアルファベット類が異なるサイズラベルを持つときコストを上げる。

これらのコスト付けを行っても最小コストのものが複数存在する場合は文字認識結果の類似度の和が最も大きな数式構文木を最適な構文木とする。

3. Center-Band の取得とその利用

本節では Center-Band の求め方、および数式認識への利用法を述べる。

3.1 前提条件

Center-Band の求めるために、以下のことを条件とする。

- 認識にかける画像は文章（数式を含む）の領域と図表の領域は分割されているものとし、本論文は文章領域のみを対象とする。
- 文章領域はブロック分割が行われている。ブロック分割とはヘッダやフッタのように文字サイズの違うものを別のブロックに分けることである。
- 行の切り分けが正しく行われているとする。2 行以上が混合されているものは Center-Band が一つではないため、対象としない。
- 各文字は外接矩形と一文字認識の結果を持っているとする。ただし、一文字認識の精度は高いものとするが誤認識は含まれていてもよいとする。
- 一文字認識の結果を利用して、各ブロック・各行の標準文字サイズ(x,y,z)、および添え字文字サイズ(x,y,z)がそれぞれ求められているとする。一文字認識の精度が高いことから文字数が多ければ標準文字サイズはほぼ正確に取得できるが、文字数が極端に少ないブロック・行に関しては信頼性が低い。
- 画像の傾きは少ないものとする。

3.2 Center-Band の取得方法

Center-Band を取得する方法を述べる。操作は各行ごとに行う。

点類・アクセント記号類は Center-Band を利用した方法では判定ができないので、以後無視するものとする。また、一般的に分母・分子の文字サイズはベースライン上の文字サイズと同じである。そのため Center-Band を誤らせる可能性がある。それを回

避するために、分数線の上下にある文字は対象外とした。

画像はあまり傾いていないことが前提であるが、現実的にはスキャニング時に傾いたり印刷状態が悪かったりと、全く傾いていない画像データを作るということは困難である。そのため、更に傾きを補正する必要がある。その方法は後で述べることにする。以後、矩形位置は傾き補正されているとする。

3.2.1 外接矩形から Center-Band を得る

まず、行中の文字の認識結果と外接矩形からセンター部 (y) の上下の座標を y_1, y_2（図 5）とし、それぞれのヒストグラムを作る。

ここで対象とする文字は英数字またはギリシャ文字と認識されたものである。アセンダー部分やディセンダー部分をもつ文字の場合、外接矩形からは直接センター部の座標は分からないので、そのブロックの標準文字サイズ xyz の値を使って推定する。

図 5：センター部分の座標 y_1, y_2

また、外接矩形の高さが標準文字サイズと異なっている場合、それは文字認識が誤っている、添え字である、特殊なフォントである、等ということが考えられるが、いずれにせよヒストグラムを乱す恐れが高いので対象としない。今回の実験では、外接矩形の高さが標準文字サイズより 10%以上大きいか小さい時に対象外とした。

得られたヒストグラムの最も大きな山となるところを Center-Band の座標として採用する。平均ではなくヒストグラムを使うことで統計的に誤認識の影響は無視される。しかし、多少のゆがみ等は吸収される必要がある。そこで各度数 k_n に（n は y 座標）付近の値に重みつきで足した値 k'_n でヒストグラムを作り直した。今回の実験ではその計算を

$$k'_n = k_n + \sum_{\substack{i=-3 \\ i\neq 0}}^{3} \frac{k_{n+i}}{|i|}$$

とした。

3.2.2 例外的な処理

以上の方法では Center-Band を正しく得られないことがある。例えばブロック分割を誤ってしまった場合である。脚注やリファレンス等が本文と同一の領域にあった場合、その行にはブロックの標準文字サイズと一致する文字がないため、ヒストグラムが作られない。誤認識により、誤ってヒストグラムの対象となる文字が現れることはあるが、それを信用しては当然 Center-Band を誤ってしまう。これを回避するために、行の標準文字サイズを使用して前述と同様にヒストグラムを作り、Center-Band を再取得する。今回の実験では再取得の条件を、ブロックの標準文字サイズを使ったときの対象となる文字数が、その行のテキストの文字数の半分以下ならば行うとした。ここでテキストの文字数を使った理由は、テキストの半分も Center-Band の文字の対象となる文字がないということは標準文字サイズの方に問題があると考えられるからで、テキストの文字数がある程度あれば行の文字サイズも信頼できるからである。

また、分数線と演算子のみからなる数式においても Center-Band を取得できない。そのようなことを想定した Center-Band の取得方法を以下のようにした。

その行の中から最も長い分数線を探し出す。見つかればその外接矩形の bottom に標準文字サイズのセンター部 y の半分の値を足し引きし、その 2 つ値を Center-Band の上下の座標とする。ここで添え字上の分数線が候補にならないように分母・分子には標準文字サイズに近い文字が含まれていることを条件とした。今回の実験では標準文字サイズと添え字文字サイズの平均よりも大きい文字が含まれていることとした。またこの方法はヒストグラムの対象文字が極端に少ないときにのみ行う。今回の実験では対象となる文字が 2 文字以内の場合とした。

これらによっても Center-Band が得られない場合は現在のところ Center-Band の取得は行わないこととした。その行に関してはこれ以降の処理は行わない。

3.2.3 Center-Band によるベースライン判定

得られた Center-Band を使ってベースライン判定を行う。ベースライン上であると判定されたものにはフラグを立てることにした。ここでは文字認識の結果は使わず外接矩形のみから判定することを原則とする。文字認識結果に左右されないためである。し

かし、判断が難しい場合に悪影響を防ぐためにのみ使う。

各文字・記号類の外接矩形と Center-Band の上下の y 座標を比較する。もし外接矩形が Center-Band を覆っている場合、フラグを立てる。ただし、誤差やゆがみ等を考慮し、ゆとりを持たせる。今回の実験では Center-Band の上から 1/9 と下から 1/7 は覆われていなくてもよいこととした。また、Center-Band に外接矩形が覆われている場合にもフラグを立てる。ただし、第二添え字等が混入することも考えられるのでいくつかの条件を科す。まず、認識結果を使って英数字・ギリシャ文字だった場合は標準文字サイズ、添え字文字サイズと比較し、添え字文字サイズに近ければフラグは立てない。また、Center-Band 内における外接矩形の位置があまりに上だったり下だったりしてもフラグは立てない。今回の実験では外接矩形の top が Center-Band の中心位置より下にある場合、bottom が Center-Band の上から 30%のところにある場合とした。

ここでフラグが立った矩形をベースライン上の文字とみなす。ベースライン上の文字が誤認識や接触などで正規化サイズや正規化中心がずれてしまっても、ここで拾い上げることができる。しかし、フラグが立っていないからといって必ずしも添え字であるとは限らない。特に古い文献ではしばしば図 6のような特殊なフォントが使われており、添え字でなくともこの条件でははねられてしまうからである。

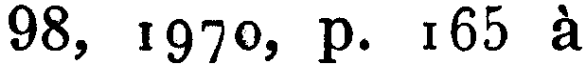


図 6：特殊な数字フォント

3.3 数式認識への利用

ベースライン判定の結果を 2 節で述べたネットワークの作成、無矛盾数式構文木の取得の際に利用する。

ネットワークを作成する際、仮に添え字としてのリンクの可能性があったとしてもフラグが立っている矩形ならばそのリンクを削除する。このことによって水平な接続、もしくは自身がルートになることしかできない。しかし、水平に接続したとしても親となる矩形が添え字となるかもしれない。そこで大域的コストによる再評価を行う際に、フラグが立っている矩形が添え字上にあった場合はその部分木に大きくコストを与えることとする。

この処理によってフラグが立っている矩形が添え字領域に行かないようにする。また、ネットワークの辺を削除することで組み合わせの数の増加を抑えることにも効果がある。

3.4 傾きの補正

ここで傾きの補正の方法について述べる。

まず、ある程度長い行に関して各行の傾きを求める。短い行では誤差による悪影響が考えられるためである。今回の実験ではそのブロックの幅の 90%以上の長さが必要とした。行の傾きの求め方は、以下のようにした。行の両端の矩形を使ってそれぞれ Center-Band を求める。実験では左右の端からブロック幅の 20%にある矩形を使用した。傾きがあれば左右の Center-Band の高さがずれている。その差をその行の傾きとする。

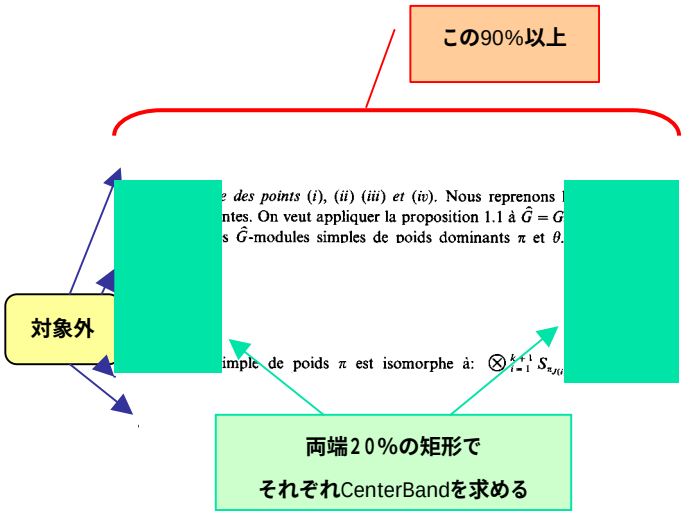


図 7：傾き取得の対象

対象となる全ての行から傾きを求め、Center-Band 取得時と同様に重みをつけたヒストグラムを作る。その最も高い山となるところをブロックの傾き s とし、それをブロックの幅 W で割ったものを傾き係数 S とする。今回の実験では左右 20%ずつの矩形を使ったため、傾き係数 S を以下の式で与える。

$$S = \frac{s}{0.8 \times W}$$

ヒストグラムを使うため、多少傾きを正確に取れない行があっても統計的に無視される。ただし、対象行が少なく極端に山が低い場合は、誤っている可能性もあるので傾き係数 S は 0 とし、特に傾き補正は行わない。今回の実験ではその閾値を度数が 1 以

下のときとした。

傾き S と外接矩形の中心 x 座標 x_c から以下に示す式によって与えられる値 C を

$$C = |x_c - x_0| \times S$$

と定義し、y 座標に加えることで水平に補正されるようにした。

4. 実験と考察

Center-Band の利用が数式構造解析に与える影響について実験したので、その方法、結果と考察を述べる。

4.1 実験環境

今回の実験では、12 の論文を 600dpi でスキャンし、その画像を認識にかけた。論文誌は 1970 年代初頭のものと 1990 年代のものを使った。Center-Band を使用した結果、Center-Band を使用しなかった結果、の 2 つを正解と比較し、数式内での位置付けを誤った矩形の数を数えた。位置付けを誤るとはベースラインから見たリンクの深さ、種類が間違っている物の数である。以下の例（図 8）では下線の引いてある 6 つを誤りとして数える。

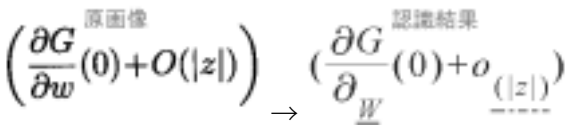


図 8：誤りの数え方

ただし点類・アクセント記号類に関しては、本手法と関係がなく数式構文解析後に挿入するという点から数値の対象外としている。

また、接触・分離が起こっている文字に関してもその位置が正しいかどうかの定義が難しいため数値の対象外とした。

4.2 結果と向上した例

実験の結果は表 1のようになった。

‘page’は認識にかけたページ数、‘line’はその行数である。ただし、行切り出しを間違ったものはカウントしていない。‘Character’は数式構文解析にかかった矩形のうち比較の対象となった文字数である。Center-Band を使わなかった場合の位置付けを誤った矩形の数を‘Before’、使った場合の位置付けを誤った矩形の数を‘After’とし、比較した。また、（ ）の中は Character に対する誤りの割合である。

	page	line	Character	Before	After
1990 年代	83	2758	36810	359(0.98%)	200(0.54%)
1970 年代	88	2665	37395	1046(2.80%)	355(0.95%)
合計	171	5423	74205	1405(1.89%)	555(0.75%)

表 1：出版年による比較

この結果からも分かるように、古い雑誌の方が構文解析は難しい。それは文字がかすれていたり接触したりするため、大きさの違う文字に誤認識をおこしやすいからである。今回の実験から、本手法は、最近の文献はもちろん、ある程度古い文献に対しても強力に作用し、構造の誤りを減らすことができるといえる。

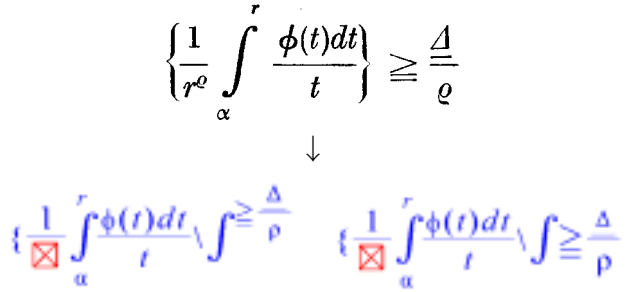


図 9：実験の結果

図 9は 1970 年の論文の一部である。上が元の画像、左下が Center-Band を使用しなかった場合の認識結果、右下が使用した場合の認識結果である。ここでは「}」がかすれて二つに分離し、上部が「＼」、下部が「∫」と認識された。以前はこのような場合、「≧」は必ず上付き添え字となり、それ以降も全て添え字となっていた。それに Center-Band を利用することで強制的に添え字にならないようにし、構造の誤りを修正することができた。

ここで最も注意すべきはこの手法による悪影響である。今回の実験では、Center-Band が原因で添え字領域のものが誤ってベースライン上になったというものは見つからなかった。

種類ごとに分析した結果を以下に示す。

数式領域

	Character	Before	After
ベースライン	36449	190(0.52%)	20(0.05%)
添え字	13961	533(3.82%)	467(3.34%)
数式領域合計	50381	723(1.44%)	487(0.97%)

表 2：数式中の誤りの数

表 2は、数式中の矩形について正解がベースライン上である矩形と添え字である矩形に分けてカウントしたものある。これより特に文字に対して大きな効果が得られたことが分かる。しかし、本手法を組み入れたならばベースライン上の矩形は全て正解となってほしいところである。なぜ誤ったか、その原因は以下の二つである。

一つはフォントが特殊だった場合である。その例を図 10 に示す。上が元の画像で下がその解析結果である。「2」が小さいためベースライン判定の条件からもれてしまう。それだけでは添え字になるとは限らないが、前の矩形との関係(*H*,*D*)を計算し散布図に当てはめると「2」は「*f*」の添え字であるとなってしまう。このような特殊なフォントへの対応が課題である。

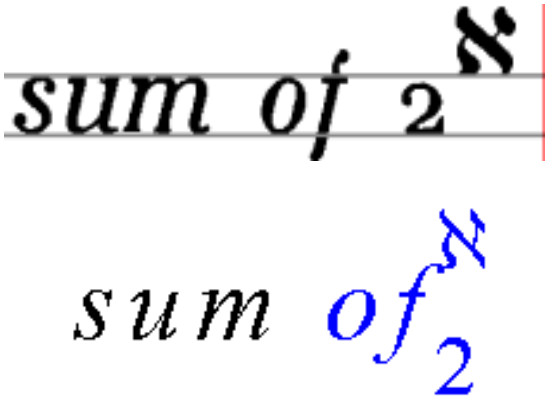


図 10：フォントが特殊なため間違えた例

もう一つは Center-Band 自体が取得できなかった場合である。Center-Band が取れないのはブロック分割を誤り、脚注等の文字サイズの違う行が本文と同じブロックになってしまったときに起こりやすい。文字の大きさがブロックの標準文字サイズと同じものがなくヒストグラムが取れない。そのようなことも考慮にいれて行の標準文字サイズを使うようにしている。しかし、文字数が少ない場合や数式しかない場合は行の標準文字サイズはあまり信用できないため、今回はそのような場合は Center-Band の取得をあきらめている。また、本文中でも文字数が少ないと誤認識によってヒストグラムが取得できずに Center-Band が取得できないことがある。閾値や条件をゆるくすることで取得できるようになるかもしれないが、誤った Center-Band を取得しかねない。その調整とともに安定して行の標準文字サイズを取れるようにすることも今後の課題である。

また、添え字領域に関しては直接何も施してないが、結果が向上している。それは、今回の数え方ではある矩形の位置が間違っている場合はその添え字も間違えていると数えるので、親となる文字が正しく判定されることで自分自身も正しい位置にいくからである。

テキスト領域

	Character	Before	After
テキスト領域	23825	682(2.86%)	68(0.29%)

表 3：テキストの誤りの数

表 3は、本来は数式ではなくテキストであるが誤って数式認識にかけられた矩形の数と、さらに誤って構造をもってしまった矩形の数である。テキスト領域にも数式領域のベースライン上を同様に大きく向上が見られた。この誤りのほとんどは特殊な数字フォントであった。

5. まとめ

本論文では Center-Band を使うことで構文解析の精度、特にベースライン判定の精度を大きく上げることができることを示した。

しかし、文字が少ないときなどに Center-Band が得られないことがある。より確実に Center-Band を取得するよう改良の余地がある。また、古い文献などに多く見られる特殊なフォントには対応していない。何らかの方法を考えるべきである。これらが課題として残っている。

文　献

[1] 岡本 正行、トワキョンド ムサフィリ ハシム、“ 周辺分布特徴を用いた数式構造認識 ”、信学論、J78-D-II、No.2、pp366-370(1995-2)

[2] 岡本 正行、東 裕之「記号レイアウトに注目した数式構造認識」、信学論、J-78D-II、No.3、pp474-482(1995-3)

[3] R. J. Fateman, T. Tokuyasu, B. P. Berman and N.MitchellOptical Character Recognition and Parsing of Typeset Mathematics,Journal of Visual Communication and Image Representation vol 7 no. 1 (March 1996), pages 2-15.

[4] K.Inoue、R.Miyazaki、M.Suzuki：「Optical recognition of printed mathmatical documents」、Proceedings of the Third Asian Technology Conference in Mathematics、pp280-289、(1998-8)

[5] 中山 優幸、福田亮治、鈴木昌和、玉利文和：「数学記号の特徴を用いた数式の水平分割による数式構造解析」、信学技報(2001-03)

[6] 江藤裕子、笹井真樹、鈴木昌和、“ 仮想リンクネットワークを用いた数式構文認識 ”、信学技報、PRMU2000-202 (2001-03)

[7] Y.Eto, M.Suzuki. Mathematical Formula Recognition Using Virtual Link Network, Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, IEEE Computer Society Press (2001) 430-437