社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

文字画像の実時間クラスタリングを用いた文書認識と修正作業の効率化 －英文数学書の場合－

安藤英里子†　　鈴木　昌和††

† 九州大学大学院数理学府
〒 812–8581 福岡市東区箱崎 6–10–1
†† 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　認識実行時に対象文書中の文字・記号画像についてクラスタリングをおこなう。そして、クラスタごとに多数決をとり、文字認識の結果を確定する。これにより、文字・記号の認識率の向上も見られた。また、クラスタの情報を使用し、クラスタ単位での修正をおこなうことで、誤認識修正作業の効率化を図る。その方法としては、認識結果を修正する際、ある修正を行なうと同じクラスタのものもすべて修正する自動修正とクラスタリングを行ない、文字認識結果の確定後、クラスタごとに認識結果の確認、修正をした後に数式構文解析をすることでほぼ誤りのない認識結果を得る方法の 2 つを提案する。

キーワード　OCR、クラスタリング、修正作業の効率化

Document recognition by real-time classifications of character images and reduction of correction labor of recognition results

Eriko ANDO† and Masakazu SUZUKI††

† Faculty of Mathematics, Kyushu University
6–10–1 Hakozaki, Higashi–ku, Fukuoka, 812-8581 Japan
†† 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　In the process of the recognition of documents, we perform a real-time classification of character images. After the classification, the result of character recognition is determined by majority in each class, so that the character recognition rate is improved. We propose two methods of reduce the labor to correct the recognition results and report the experimets to estimate the performance. One of the method is the automatic correction method for the caracters belonging to a same class. When a user correct a character, then all the other characters belonging to the same class are corrected simultaneously. In the second method, after the classification of character images for all the pages of the volume to be recognized, the system lists the classification results of the characters and present a method of cluster-wise correction of the recognition results, before performing the structure analysis of mathematical expressions, to improve the efficiency of digitization work of mathematical documents.

Key words　OCR, Clustering, reduction of correction labor of recognition results

1. はじめに

1.1 研究目的

インターネットの急速な発展に伴い、電子ジャーナルへの移行が進む中で、現存する大量の数学論文誌を電子化する動きが活発化している。これが実現されれば、インターネットで結ばれた電子ジャーナルの環境の中に、過去に出版された論文誌も統合し、どこからでも見たいものを検索・閲覧することができるようになる。更に参考文献等のリンクも可能になり、有効な情報を簡単かつ迅速に取得することができる。現存する数学論文誌は、仮に 1960 年以後の主要な雑誌 100 程度に限っても、全体で約 1000 万ページに及ぶ。これらの大量の数学文書を電子化するためには、数学文書に対応した OCR システムの開発が重要となる。

しかし、現在の技術では OCR において誤認識を避けて通ることはできない。現在の一般的な OCR では印刷された文書を 400dpi〜600dpi 程度の高解像度でスキャンした画像に対しては、性能のよいものは 99% 以上ある。それでも数学雑誌の 1 ページ平均文字数は約 1500 程度あり、認識率 99% では各ページが 15 個程度の誤認識を含むことになる。しかも、通常の OCR は数学雑誌に対応していないものがほとんどであり、数式の存在に影響されてテキスト部分の認識率までもが低下する。近年、我々が開発してきた数学書認識システムは、数式部分も認識率が向上し ([2])、数式領域とテキスト領域の分離が高い水準で切り分けられるようになった。昨年度の段階でテキスト部分の認識率 99.3% 程度を実現している ([1])。しかし、それでも各ページ平均数個から 10 個近い誤認識を含む。

誤りを探し出した上で、修正を行なうのは非常に大変な作業である。しかも OCR は同じ種類の誤りがたくさんあり、同じ修正を何度も繰り返し行なわなくてはならない。同じ作業の繰り返しは大きな負担となる上、それだけ労力とコストがかかる。これは電子化プロジェクトを進めるときの大きな問題点となる。これを回避するためには認識率を向上させることも重要である。しかし、ノイズなどの影響により認識しにくいものもあり、完全な認識は困難である。よって、修正作業は必要不可欠である。

そこで、少ない回数の修正で正解が得られることに重要な意義がでてくる。同じ種類の誤認識が一度に修正されるならば、修正の回数は減る。同じ作業を繰り返さなくてすむだけでも負担が大きく軽減される。よって、できるだけ少ない回数で修正がすむことを本研究の目標とした。そのために、認識実行時に、同じ誤りの文字が同じクラスタになるように認識対象文書中の文字・記号画像のクラスタリングを行い、クラスタ単位で文字認識結果を多数決により確定する。そして、ある文字の修正が行なわれると同じクラスタのものは自動的に修正されることにより修正回数の減少を図ることにした。また、修正が行なわれた数式領域に関して再び数式構文解析を行なうことにより文字認識の修正回数だけでなく、数式構文の修正回数の減少も図ることとした。

また、文字認識結果が正しいとき、数式構文解析の認識率は高い ([2])。そこで、クラスタリング、文字認識結果の確定後、クラスタごとに文字認識結果の確認、修正を行ない、数式構文解析を行なうという新しいインタフェースを実現することでも修正回数の減少を図った。

前半では 2 種類のクラスタリングの具体的な方法、認識結果の確定方法、実験結果について述べる。後半ではクラスタリングの情報を用いた修正作業の効率化の実現方法と実験結果について述べる。

1.2 実現方法

クラスタリングの代表的な方法の 1 つとして、k-means 法が挙げられる。この方法はクラスタ数やデータ数が多いと非常に時間がかかる。しかし、今回は認識時にクラスタリングを行なうため、現実的な時間でなくてはならない。

そこで、クラスタ数を固定せずにデータを逐次クラスタリングしていき、距離が一定以上のものは別クラスタとし、新しいクラスタを作成する。つまり、1 つの文字種が複数のクラスタに分かれることを許す。この条件により実時間クラスタリングを実現する。

クラスタごとに文字認識結果の確定、修正を行なうことを実現するため、2 種類以上の文字種が 1 つのクラスタに混ざることはないことを条件とする。1、l、|(vert) については形が良く似ているため混在を許し、今回の自動修正の対象外文字とする。他の類似文字（S と s,O と o と 0 など）も混在を許す。

クラスタリングは基本的に連結成分単位で行なう。ただし、文字認識結果で複数連結成分で 1 つの文字とみなしたもの (例　i, ≦, 分離文字など) に関してはそれらを 1 つの単位としてクラスタリングを行なう。また、点類、アクセント類についてはクラスタリングを行なわない。

2. 輪郭線方向線素特徴量を用いたクラスタリング

2.1 輪郭線方向線素特徴量

2.1.1 輪郭線方向線素特徴量の定義

文字画像から境界点の情報を取得し、輪郭点と輪郭点のつながりを上下左右方向を同一視した 4 方向を数値化したもので表す。各方向の数を数え上げたものを輪郭線方向線素特徴量とする ([1])。今回は文字・記号画像の外接矩形を 3×3 の領域に分割し、各領域内で求めた計 36 次元の特徴量を使用した。

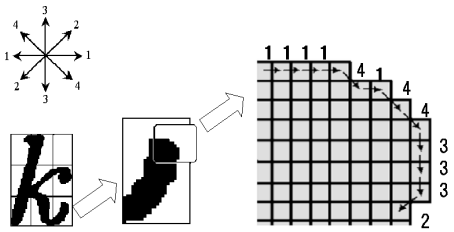


図 1　輪郭線方向線素特徴量

2.1.2 輪郭線方向線素特徴量の距離の定義

2 つの特徴量の距離に各成分の差の和を用いた。

2.2 処 理 手 順

2.2.1 流　　れ

各成分に対して、輪郭線特徴量を求め、特徴量の距離によりクラスタリングを行なう。成分とクラスタの特徴量が一定距離以内のものをそのクラスタに属するとする。全成分に対してクラスタリングを行なった後、距離が近いクラスタ同士は統合する。

各クラスタはサイズ (幅、高さ)、クラスタに属する成分の平均特徴量、属する成分の中で平均特徴量に近い輪郭特徴量を持っている。

新しい成分が入る度に、平均特徴量を更新し、平均特徴量に近い特徴量と成分の特徴量とで平均特徴量に距離が近い方を平均特徴量に近い特徴量とする。

2.2.2 手　　順

（1） 成分のサイズと輪郭線特徴量を求める。

（2） 成分が属するクラスタを判定する。

クラスタに属する条件は以下の 3 つである。

- クラスタと成分のサイズがほぼ同じ。

成分と平均特徴量に近い特徴量について、

- 各成分の差が一定値以内。
- 距離が一定値以内。

（3） 2 の条件を満たすものが存在した場合、該当クラスタの平均特徴量を更新する。平均特徴量と成分との距離が平均特徴量と平均特徴量に近い特徴量との距離より小さいとき、平均特徴量に近い特徴量を成分の特徴量とする。条件を満たすものが存在しない場合、新しいクラスタを作成し、そのクラスタのサイズを着目成分のサイズ、平均特徴量及び平均特徴量に近い特徴量を成分の特徴量にする。これを全成分に対して行なう。

（4） クラスタの再クラスタリングをおこなう。クラスタの平均特徴量に近い特徴量を特徴量として、2 の条件を満たすクラスタが存在した場合は統合する。その場合、統合する 2 つのクラスタの所属個数が多い方に統合し、平均特徴量、平均特徴量に近い特徴量の更新は行なわない。

	クラスタリング	再クラスタリング
サイズ	3	3
各成分の差	15	15
各成分の差の和	110	80

表 1　各パラメータ値

3. 画像マッチングによるクラスタリング

この方法は、クラスタと成分の画像において一方の 2 ピクセル拡大が他方の画像を含むかを互いに満たすかで判定する。

3.1 RunList

3.1.1 RunList の定義

文字・記号画像を以下の形式に変換する。

連続する黒画素の並びを Run と呼ぶ。外接矩形の高さを h とする。$i(i = 1, \cdots, h)$ について、横方向にみたときの Run のリストを RunList と呼ぶ。

RunList L の i 番目の Run を L_i で表し、L_i の開始、終了位置を $L_{i,s}, L_{i,e}$ で表す。

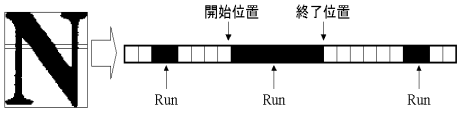


図 2　Run

3.1.2 RunList の拡大

RunList L を a 拡大を以下のように定義する。

$$L_{i,s} := L_{i,s} - a, \quad L_{i,e} := L_{i,e} + a$$

ただし、

$L_{i,s} - a \leqq L_{i-1,e} \Rightarrow L_{i,s} := L_{i-1,s}$, L_{i-1} は削除。
$L_{i,e} + a \geqq L_{i+1,s} \Rightarrow L_{i,e} := L_{i+1,e}$, L_{i+1} は削除。

3.1.3 RunList の和

RunList X, Y の和 RunList Z を以下に定義する。

$X_{i,s} \leqq Y_{j,s} \leqq X_{i,e}$ または $X_{i,s} \leqq Y_{j,e} \leqq X_{i,e}$ を満たす j が存在したときは

$$Z_{k,s} = \min(X_{i,s}, Y_{j,s}), \quad Z_{k,e} = \max(X_{i,e}, Y_{j,e})$$

とする。上の条件を満たさない X_i または Y_i が存在したときは Z_k をそれぞれ X_i, Y_i とする。

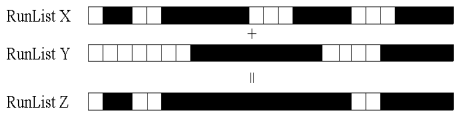


図 3 RunList の和

3.1.4 RunList の包含

RunList X が RunList Y に含まれるとは

$$\forall\, i \;\; \exists\, j \;\; s.t.\; Y_{j,s} \leqq X_{i,s} \text{ and } X_{i,e} \leqq Y_{j,e}$$

が成り立つときであるとする。

X が Y に含まれないときは含まれなかった部分の長さの和を含まれない個数とする。

3.2 処 理 手 順

3.2.1 流　　れ

クラスタの各 RunList が着目成分の RunList に含まれるか、またその逆も成り立つかどうかでそのクラスタに属するかどうかを判定する。

各クラスタはサイズ、重心と最初に属した成分の各行の RunList を持つ。

3.2.2 手　　順

（1） 成分のサイズ、RunList、重心を求める。

（2） 成分がどのクラスタに属するかを判定する。

x, y 座標に関するクラスタと成分の重心のずれを求める。重心のずれは最大 1 まで許す。成分がクラスタに属する条件は、以下の 3 つである。

- クラスタと成分のサイズがほぼ同じ。
- クラスタの i 行目の RunList を 2 拡大し、前後 2 行の和を求める。ただし、前後 2 行のいずれかが存在しないときは残りの RunList との和を求める。x 座標の重心のずれを考慮して、成分の対応する行の RunList がこれに含まれるかどうかを調べる。クラスタの i 行目に対応する行が存在しないときは成分の RunList は空と想定する。成分の i 行目に対応する行が存在しないときはクラスタの最後 (最初) の行を 2 拡大し、前 (後)2 行の RunList の和を求める。このとき、x 座標の重心のずれ分だけずらして 2 つの RunList が包含関係にあるかを調べる。含まれない場合は含まれない数を数え、一定値以内のときは次の行について調べる。含まれない数の和が一定値を超えたらそのクラスタには属さないとする。
- 同じやり方で逆を行なう。

サイズ (幅、高さ)	3
高さ $>$ 45 かつ縦横比 $>$ 1.5	10
高さ $>$ 45 かつ縦横比 $\leqq$ 1.5	5
高さ $\leqq$ 45	0

表 2 パラメータ値

（3） 条件を満たすクラスタが存在しない場合、新しいクラスタを作成し、成分のサイズ、RunList を新しいクラスタのそれぞれとする。これを全成分に対して行なう。

4. 文字認識結果の確定

4.1 クラスタの候補文字

各クラスタは所属する成分の第 1 候補をそのクラスタの候補文字とする。成分に対して文字認識結果が 2 つ以上の文字と認識した場合は複数文字で 1 つの候補文字とする。候補文字が 2 つ以上の文字からなる場合は、その候補文字は候補文字の外接矩形を基準としたときの各文字の開始位置とサイズを持つ。

4.2 候補文字のコスト

候補文字のコストはその候補文字を第 1 候補とする成分のコストの和を個数で割ったものとする。2 文字以上で 1 つの候補の場合、各文字のコストの和を文字数分で割ったものをその候補のコストとする。

4.3 文字認識結果確定

クラスタに所属する成分の第 1 候補で多数決をとる。最も数が多かった候補文字をクラスタの第 1 候補とする。数が等しかった場合は、コストが良い方を第 1 候補とする。クラスタの第 1 候補を各成分の第 1 候補とする。第 1 候補が 2 文字以上からなる場合、クラスタに記録した情報をもとに成分を分ける。ただし、$1, l, |$ に関しては自身の文字認識結果にする。

5. 実 験 結 果

8 種類の英文数学論文 70 ページを 600dpi でスキャンした接触文字を含まない画像について輪郭線方向線素特徴量と画像マッチングを用いたクラスタリングを行なった実験結果について述べる。

5.1 文字認識率

文字・記号を英数字、ギリシャ文字、括弧、矢印、数学記号に分類し、文字サイズ別 (本文、添え字サイズ) に誤認識率を求めた。データ数を表 3 で示す。

	英数字	ギリシャ	括弧	矢印	記号	合計
本文	79476	1793	4450	118	1932	86869
添え字	2424	567	189	28	342	3550
合計	81900	2360	4639	146	2274	91319

表 3　文字種ごとの実験データ数

笹井氏らが開発した文字認識エンジン ([1]) を使用した場合の実験結果を表 4 に示した。

		英数字	ギリシャ	括弧	矢印	数学記号	合計
クラスタリングなし	本文	0.90	1.90	1.24	0	1.09	0.94
	添え字	3.18	3.18	1.06	0	9.06	3.61
	合計	0.97	2.20	1.23	0	2.29	1.05
輪郭線特徴量	本文	0.61	1.73	0.79	0	1.04	0.52
	添え字	2.52	3.53	1.59	0	9.65	3.30
	合計	0.67	2.16	0.82	0	2.33	0.76
画像マッチング	本文	0.58	2.01	0.67	0	1.04	0.62
	添え字	2.23	3.18	1.06	0	9.65	3.01
	合計	0.63	2.29	0.69	0	2.33	0.71

表 4　誤 認 識 率

また、OCR ソフト Express Reader Pro(注1)の文字認識エンジンを用いてテキスト領域文字を対象としたときの実験も試みた。

	誤認識率
クラスタリングを行なわない	0.48
輪郭線特徴量によるクラスタリング	0.22
画像マッチングによるクラスタリング	0.22

表 5　Express Reader Pro を用いた場合

部分的に認識率が下がるところもあるが、全体として誤認識数が約 3 分の 2 から半分に減少した。

5.2　クラスタ候補文字数

候補文字を複数持つクラスタは全体の 2, 3% 程度 (表 6) で多くは全一致で正解か、不正解である。

候補文字数	クラスタ数	
	輪郭線特徴量	画像マッチング
1	5436	5286
2	116	146
3	22	23
それ以上	2	3
計	5576	5444

表 6　各クラスタの候補文字数

5.3　認識率に影響を与えるクラスタ、文字

影響を及ぼすクラスタ数、文字数を表 7 に表した。

良い影響を及ぼすクラスタとは、候補文字が複数あり、多数決の結果、第 1 候補が正解候補となるクラスタで、悪い影響を及ぼすクラスタは、その逆である。良い影響を受ける文字とは、良い影響を及ぼすクラスタに属し、自身の文字認識結果は不正解の文字で、悪い影響を受ける文字は、その逆である。

	輪郭線特徴量	画像マッチング
良い影響クラスタ数	100	117
悪い影響クラスタ数	33	30
良い影響文字数	285	316
悪い影響文字数	71	58

表 7　良い影響と悪い影響

文字画像はノイズを含むため特徴量に差が現れ、誤認識が生じる。正解に認識される場合とそうでない場合の特徴量の境界付近を含むクラスタが作成されると、良い影響か悪い影響を及ぼすクラスタになるが、文字認識率が高いため、誤認識文字は少なく、良い影響か悪い影響を及ぼすクラスタになる可能性が高いと考えられる。そうでない場合でも誤認識文字数が少ないため悪い影響を受ける文字は少なく、結果として文字認識率が向上しているのではないかと考えられる。

表 8 より、実際、悪い影響を及ぼすクラスタの半数以上のクラスタが悪い影響を受ける文字が 1、2 であることがわかる。1 つの良い影響を及ぼすクラスタの中で良い影響を受ける文字数も少ないが、クラスタ数が多いため、認識率が向上している。

影響を受ける文字数	クラスタ数			
	悪い影響クラスタ		良い影響クラスタ	
	輪郭線特徴量	画像マッチング	輪郭線特徴量	画像マッチング
1	20	19	61	58
2	6	6	16	27
3	2	1	7	11
それ以上	5	4	19	21
計	33	30	100	117

表 8　影響を及ぼすクラスタに属する影響を受ける文字数

5.4　クラスタ数

図 4,5,6 でページ数増加に伴うサイズ別文字種数とクラスタ数の増加を表した。輪郭線特徴量の方は再クラスタリングを行なう前のクラスタ数になる。

増加傾向としては輪郭線特徴量も画像マッチングの場合もほぼ同じといえる。またサイズ別文字種とクラスタの増加傾向が似ていることから文字種が一定になれば、クラスタ数も落ち着くと考えられる。

表 9 は同じ文字種がどれくらいのクラスタに分かれるかを調べたものである。

(注1)：(株) 東芝。

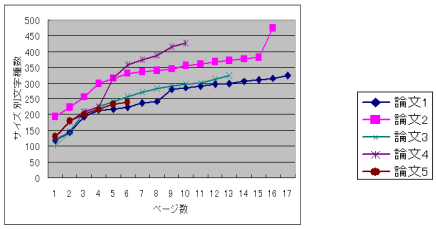


図 4　サイズ別文字種数増加

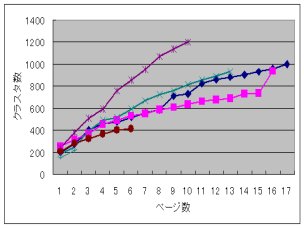


図 5　クラスタ数増加
(画像マッチング)

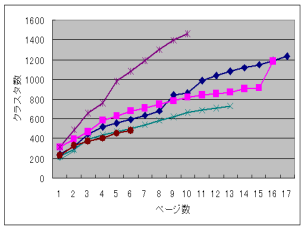


図 6　クラスタ数増加
(輪郭線特徴量)

	輪郭線特徴量	画像マッチング
平均	2.38	2.32
最大	57	37

表 9　文字種あたりのクラスタ数

5.5　混在クラスタ

今回は、閾値を厳しくしたため、2 種類以上の文字が 1 つのクラスタに混在することはなかった。

5.6　クラスタリングにかかる時間

クラスタリングにかかった時間は 1 ページ平均 3.81 秒（輪郭線特徴量）、5.30 秒（画像マッチング）で、現実的な時間である (Pentium　, 1GHz dual, Memory 2Gbyte)。

5.7　閾値変化による文字認識率

今回は、文字種の混在がないように閾値を厳しくした。したがって、所属文字数が少ないクラスタが多くなった（所属文字 1 のクラスタが全クラスタ数の約 37%）。所属文字数が 1, 2 のクラスタでは多数決が有効に働かない。文字認識率の向上という点だけから考えると、文字種が混在するクラスタが存在してもいい影響を受ける文字数が多ければよい。

そこで、輪郭線特徴量によるクラスタリングにおいて、再クラスタリングのときに所属文字数が 1, 2 のクラスタはできるだけ近いクラスタに統合されるように閾値を段階的に緩めた。所属文字数が 3 以上のものは最初のクラスタリングと同じ閾値で実験を行なった。実験結果の文字認識率、クラスタ数及び文字種が混在しているクラスタ数を図 7,8,9 に表した。

文字認識率の向上という点から考えると、文字種の混在を許してもある程度、閾値を緩めた方が良い

サイズ	3
各成分の差	20
各成分の差の和 (距離)	110

表 10　最初のクラスタリングでの閾値

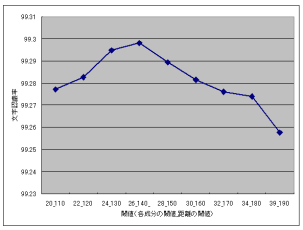


図 7　認識率の推移

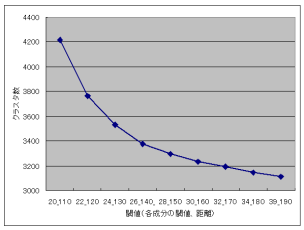


図 8　混在クラスタ数の推移

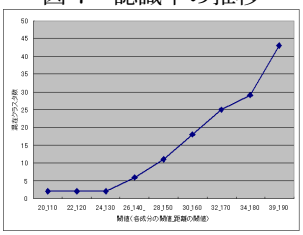


図 9　混在クラスタ数の推移

I　l(エル)

図 10　混在しやすい例

ことがわかる。混在しやすい文字の例を図 10 に挙げた。$1, l, |$ に関しては混在を許したが、I に関しても混在を許し、個別処理を加えた方が良いと考える。

6.　認識結果の自動修正

6.1　修正インタフェースの説明

InftyOcrUI と呼ばれる認識結果の表示・修正インタフェースが存在する。

InftyOcrUI はブロックごとに認識結果を表示する (図 11 参照)。主な機能は以下のとおりである。

- 文字認識結果候補の切り替え、入力。
- 文字の統合、文字の分離 (最大 3 つまで)。
- Math,Text モードの切り替え
- 選択された Math 領域の再認識
- 数式パレットを用いた数式構文の修正
- 切り取り、コピー、貼り付けといった編集機能

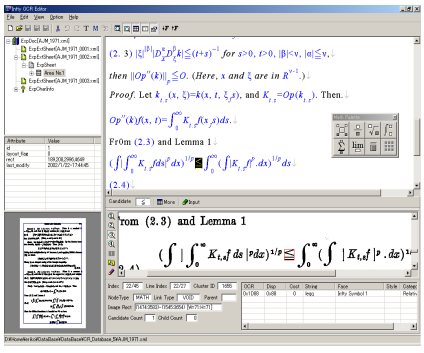
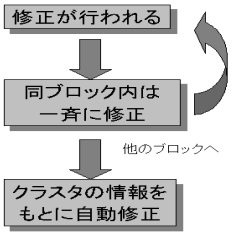


図 11　InftyOcrUI 画面

6.2 処理の流れ

InftyOcrUI を使い、自動修正の実装を行なった。流れはある修正が行なわれるとその位置以降の同一ブロック内に存在する修正されたものと同じと思われるものは同時に修正し、修正内容をクラスタに持たせておく。他のブロック、他のページに関してはそのブロックへ移るときにクラスタ情報を使用して修正を行なう。



画像 (図 12) を認識し、結果を表示したものを図 13 で示した。○で囲まれたところは'(' と p が接触し、認識結果がリジェクトとなった。この訂正を行なうと、図 14 のようになる。リジェクトのままのところは修正を行なった箇所以前にあるので、変更されない。修正箇所以降のリジェクトはすべて変更された。

Further, for all primes p, we have

$$C(p) \cong \prod_{p\in P} Z(p)/p\prod_{p\in P} Z(p) \cong \left\{X, p\prod_{p\in P} Z(p)\right\}\Big/p\prod_{p\in P} Z(p$$
$$\cong X/X\cap p\prod_{p\in P} Z(p) = X/pX.$$

If we consider a finite cyclic group $C(p^k)$, then ([2], p. 243)

$$\mathrm{Ext}(C(p^k), X) \cong X/p^kX \cong C(p^k),$$

図 12　原　画　像

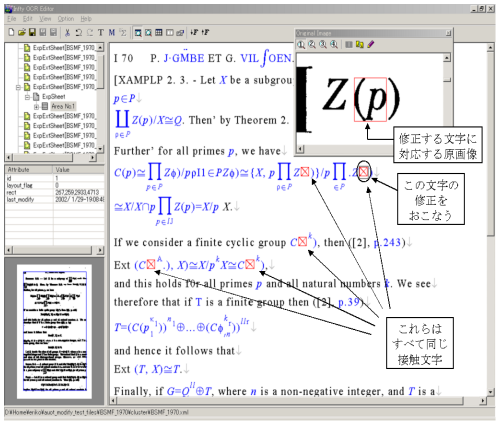


図 13　修　正　前

6.2.1 文字認識結果の自動修正

ある文字認識結果が変更されると同一ブッロク内の同じクラスタに属する文字で修正箇所以降の文字はすべて同じ候補文字に変更する。また、文字が属するクラスタの第 1 候補文字も変更する。

文字の分離を行なった場合も同様に同一ブロックで同じクラスタの文字は分離する。クラスタの情報としては分離された文字数分だけ新しいクラスタを

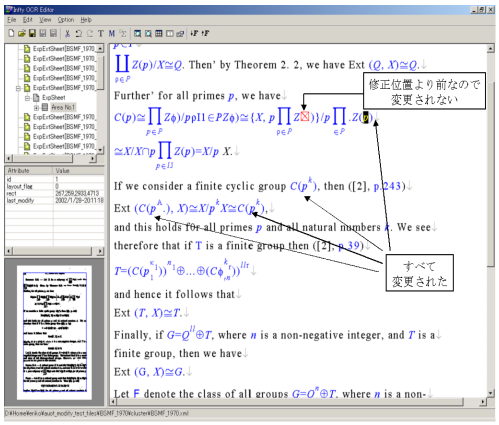


図 14　修　正　後

作成、それぞれに文字候補と分離する以前の外接矩形を基準とした各候補の開始位置とサイズ、もとのクラスタがどれであるかを記録する (図 15 参照)。

文字の統合を行なった場合も同様に同一ブロック内の同じクラスタの並びで位置関係がほぼ同じものは統合する。また、新しいクラスタを作成し、統合された文字を第 1 候補文字とし、統合されたクラスタと統合された文字の外接矩形を基準としたときの各文字の開始位置とサイズを記録する (図 16 参照)。

修正された文字が属するクラスタには修正が行なわれたことを記録する。文字の分離、統合の場合は参照すべきクラスタがどれかも記録する。

他のブロックへ移った際には各文字が所属するクラスタが、修正が行なわれているときはクラスタ情報をもとに文字認識結果の変更、統合、分離を行なう。

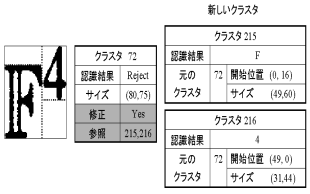


図 15　文字の分離

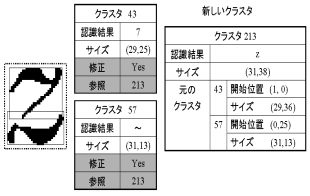


図 16　文字の統合

6.2.2 数式構文解析の自動修正

文字認識結果の修正が行なわれた文字を含む数式領域は再び数式構文解析を行なう。ここでは江藤氏らが開発した数式構文解析を利用している ([2])。文字認識結果が正しい場合の数式構文解析の完全正解率は 96.66% であることより、文字認識結果が修正されれば数式構文解析の結果もほぼ正しくなる。

6.3 実 験 結 果

自動修正に関する実験を行なった。実験方法はクラスタリングを行なった場合とそうでない場合とで認識を行なう。その認識結果をクラスタリングを行

なった場合とそうでない場合、更にクラスタリングを行なった場合は自動修正を行なわないときと行なうときとでの修正回数を数える。

修正回数の数え方は文字認識結果については候補文字の変更、分離、統合をそれぞれ修正1とする。数式構文解析については接続誤りが1つでも存在したものを修正が必要な数式として1と数えた。ただし、領域の切り分け、行分割の誤り箇所は対象外とした。

2種類の論文から合計31ページを認識し、自動修正の実験を行なった結果は表11のとおりである。

				修正回数	
				文字認識	数式認識
クラスタリング	なし			438	41
	あり	自動修正	なし	355	41
			あり	144	34

表 11　自動修正に関する実験結果

クラスタリングを行なうことにより文字認識の結果が向上する (5.) よって、クラスタリングは行なったが自動修正はしない場合でも修正回数が少なくなった。

クラスタリングを行なわなかった場合と自動修正を行なった場合とでは文字認識の修正回数が約3分の1に減少した。数式構文解析の修正回数については文字認識ほどの減少はなかった。接続誤りが1つでもあれば1つの誤りと数えているが、複雑な数式などでは1つの数式でも複数の接続誤りが存在する場合があり、1回の修正回数では直らない場合もある。それゆえ、実際の修正回数はより少ない。

7. 新しいインタフェース

クラスタリングを用いた新しいインタフェースの提案を行なう。従来の数学文書の認識は文字認識、数式構文解析を行い結果を出力する。一般的に数式構文の誤りは文字認識結果の誤りによるものが多い。今回のクラスタリングでは文字種の混在はないので、例えば論文単位で全ページの文字認識とクラスタリングを行なった後、各クラスタの結果を確認、修正後に数式構文解析を行なえば誤りのほぼない認識結果が得られると考え、このインタフェースの実装を試みた。このインタフェースでクラスタの認識結果の修正が行なわれると、自動修正のときと同様のクラスタ情報をもたせる。そして全クラスタについて確認、修正が行なわれた後、クラスタ情報をもとに各文字認識結果を変更する。

600dpi でスキャンした高品質の印刷文書30ページを対象に実験を行なった結果を表12で示した。

実験結果より文字認識の修正回数が約半分となった

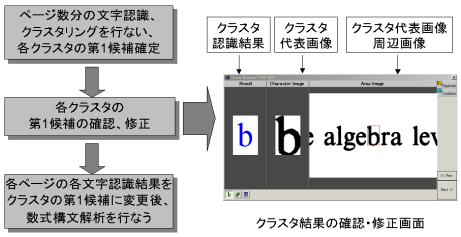


図 17　流れと新しいインタフェース画面

	修正回数	
	文字認識	数式認識
従来の認識	385	41
新しいインタフェース	171(78+93)	29

表 12　新しいインタフェースによる実験結果

たが、これは開発途中であるため、問題点を含む。そのため、新しいインタフェースでの修正を行なっても93の誤認識文字を含んだ。このうち、59はクラスタリング対象外文字や混在を許した文字になる。

通常の修正作業では誤認識箇所を1文字ずつ目で追って探し出す。これは修正作業での大変な労力でなる。しかしながら、この提案手法ではクラスタリング後に画像と認識結果を自動的に出力するため、完全に実現されると、クラスタ数分の確認・修正を行なうだけで正解が得られることになる。

今回の実験で新しいインタフェースによる確認・修正作業にかかった時間は約43分であり、インタフェースを利用しない場合は1ページの修正にかかる時間は平均10－20分ほどであるから全体で5－10時間ほどかかることになる。時間面からみても修正作業の効率化が図れると期待される。

8. ま と め

本論文では、文字・記号画像のクラスタリングによる英文数学書の認識手法と修正作業の効率化に関する手法について提案をした。そして、実験結果により修正作業の効率化と文字認識率の向上を示した。今後の課題として、以下のことが挙げられる。

- Text/Math の領域誤りと $1, l, |$ の自動修正
- 画像マッチングによるクラスタリングを用いた接触文字の推定、太字の識別
- クラスタリングを用いた日本語文書の認識

文　献

[1] 笹井真樹:「輪郭線方向線素特徴量による数学記号認識と科学技術印刷文書のレイアウト解析」平成12年度修士論文
[2] 江藤裕子:「仮想リンクネットワークを用いた数式構文認識」平成12年度修士論文
[3] 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」信学技法、PRMU2000-202、pp7-14(2001-03)