

オフライン手書き文字数式化手法の提案と 大規模平均文字の比較

新納真次郎^{†1} 斉藤絢基^{†1} 久保田夏美^{†1} 中村聡史^{†1} 鈴木正明^{†1}

概要: これまでに手書きのストロークをフーリエ級数展開によって数式化することで、複数の手書き文字の平均化や、手書き文字のアニメーション表現などを実現してきた。しかし、従来の手書きストローク数式化手法は、手書き入力システムを用いてタブレット上に書かれたものに対してのみ用いられるものであり、鉛筆やボールペンで書いた紙の上の手書きストロークに用いることができなかった。そこで本研究では、そのような手書き入力システムを必要とせず、画像に含まれるオフライン手書き文字を数式化する手法を提案する。また、提案手法によって、どれだけの精度でオフライン手書き文字の数式化が可能なのかを評価実験により明らかにする。さらに、600人規模の手書き文字データセットを構築し、平均文字を生成することで、平均手書き文字の特性を分析する。

キーワード: 手書き, オフライン手書き文字, 平均文字

1. はじめに

手書き行為は我々の日常生活に欠かせないものである。例えば、メモやノートを取ったり、事務に提出する書類を書いたり、手紙や年賀状を書いたり様々な場面で手書きが用いられている。文化庁の世論調査[1]によると、日常生活において今現在も手書きを利用する機会があると答えた人の割合は72%という統計がでていいる。また、同調査によると手書き文化をこれからも大切にすべきだと考えている人の割合は91%という統計も出ており、人々の手書きに対する関心は高いと考えられる。このような調査結果からも、手書き行為はこれからも我々にとって欠かせないものであると言えるだろう。

一方、タブレット PC やスマートフォンなどのペン入力が行えるタッチデバイスが登場したことで、コンピュータ上で手書き入力を行う機会が増えてきている。しかし、従来の紙とペンによる手書きを行う機会が減っているわけではない。これは、紙とペンによる手書きの方が手軽かつ低コストで、我々にとって馴染みがあるからだと考えられる。一般的にコンピュータを介する手書き行為をオンライン手書きといい、コンピュータを介さない手書き行為をオフライン手書きという。

我々はこれまでに、オンライン手書き文字や図形のストロークをフーリエ級数展開によって数式として表現することで、図1のように文字や図形の平均化を行うことを可能にし、さらにその平均文字や平均図形は綺麗になるということを明らかにしてきた[2][3][4][5]。

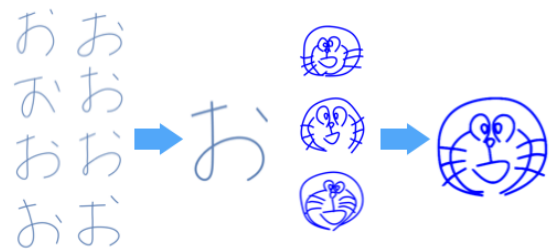


図1 平均文字・平均図形の例

しかし、これらの数式化手法はオンライン手書き入力システムを用いて書かれたストロークにしか適応できず、普段鉛筆やボールペンで書いているオフライン手書きストロークに対して適応することは不可能である。

そこで本研究では、画像内に含まれるオフライン手書き文字に対して、それらのストロークの数式化を可能とする手法を提案する。なお、本手法では数式化する文字がどのような文字であるかが、他の文字認識技術によって判断できることを前提として、数式化を試みる。また、本稿では多くの文字の種類からひらがなのみを数式化の対象とする。さらに、提案手法によって数式化した文字に対して評価実験を行うことにより、どの程度の精度でオフライン手書き文字を数式化し、その数式によって元々の手書き文字を再現可能であるかを明らかにする。また、本手法によって数式化されたオフライン手書き文字に対して、学年ごとの平均文字や、100人単位の大規模な平均文字を生成し、それらを比較することで平均手書き文字の特性について明らかにするとともに、そこから観察されることについて報告する。

本稿では、2章で関連研究について述べる。次に3章でオフライン手書き文字の数式化手法について説明し、提案

^{†1} 明治大学

手法でどれほどの精度で数式化が可能かを4章で述べる。さらに5章では実際に提案手法によって平均文字を生成することで平均手書き文字の特性を明らかにし、最後に6章ではまとめと今後の展望について述べる。

2. 関連研究

本研究のように手書き文字を対象とした研究は様々存在する。中でもオフライン手書き文字認識に関する研究は我々の研究と類似したものといえる。

篠沢は[6]、ユーザの手書き文字における自己相似性特徴という性質を用いることで、オフラインにおける手書き文字認識を行う手法を提案している。自己相似特性は、手書き文字画像から複数のパーツを取り出し、それらを組み合わせることによって、他の文字のパーツを構成できるような特性である。この手法により、約93%の精度でオフライン手書き文字の認識を可能としている。安倍らは[7]、毛筆文字というストロークのつぶれや接触が多く、ストローク抽出が困難とされる文字に対して、計算幾何の分野をもとに抽出を試みている。ここでは、毛筆文字画像において交点や接点が存在する領域と、これらを含まない領域にわけ、交差や接点を含む領域においてストロークがどのように通過するかを求めることでストロークを抽出している。オフライン手書き文字の認識をする際に、書かれた文字の筆順情報が必要になる場合がある。Huangら[8]は、オフライン手書き文字画像から曲線の滑らかさを求めることによって、一筆書きのオフライン手書き文字の筆順を復元可能としている。オフライン手書き文字の筆順を復元する他の研究としては加藤ら[9]の研究が挙げられる。この研究では、グラフ理論に基づき交点の次数を考慮して点列(ノード)を追跡することで、一筆書きのオフライン手書き文字の筆順を復元する手法を提案している。この手法により、「h」などの2度書きする部分が一部ある文字でも、問題なく筆順を復元することを可能としている。

我々の研究は、これらの研究のように文字そのものを認識することはせず、手書き文字を数式化することによって、オフライン手書き文字の平均化などに応用することが目的である。

3. 提案手法

ここではオフライン手書き文字に対して、平均化などを行うために、手書き文字のストロークを認識し、ストロークの数式化を行う。なお、今回は手法を単純化するため、事前に何の文字を書いているかをわかるようにしておき、さらにその各文字についてオンライン手書き文字のデータをサンプルとして用意する。このサンプルデータをもとに、端点の情報や、ストロークの情報をある程度把握しておくことで、オフライン手書き文字の数式化を容易にする。

本章では、オフライン手書き文字数式化手法を5つの手

順にわけて説明する。

3.1 サンプルデータセットの構築

まず、端点の情報やストロークの情報、また画数などがある程度把握するために、筆者がひらがな46語に対するオンライン手書き文字のデータセットを構築した。これらのサンプルデータセットを構築するにあたって、図2で示すような手書き文字入力システムを用いた。

このシステムは、マウスやスタイラスでキャンパス内をドラッグすると、毎フレームごとの点列情報を取得するものになっている。このシステムを用い、全てのひらがなに対して、ある書き順に従って手書き入力を行い、それぞれのストロークに関する点列情報をテキストファイルとして保存する。なお、ひらがなの書き順については、日本において一般的に通用している書き順が示されているサイト[10]を参考にした。

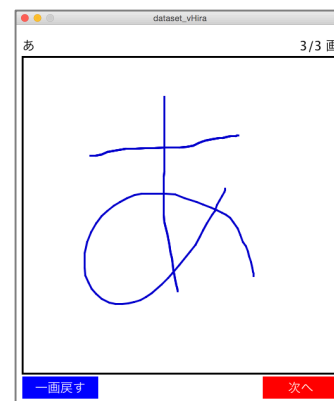


図2 サンプルデータ構築用のオンライン手書き文字入力システムの実行画面

3.2 画像処理による中心線の抽出

オフライン手書き文字におけるストロークを数式化する際には、そのストロークの太さは不要であり、中心線を正確にとることが必要となる。そこで各ストロークの中心線を取得するため、まず画像のグレースケール化を行う。次に、この画像に対して細線化を行うために、2値化処理をする。2値化処理には、大津[11]の手法を用いた。最後に、この2値化画像に対してZhangら[12]の手法を用いて細線化を行い、細線化された1ピクセルの太さのものを中心線として抽出する。

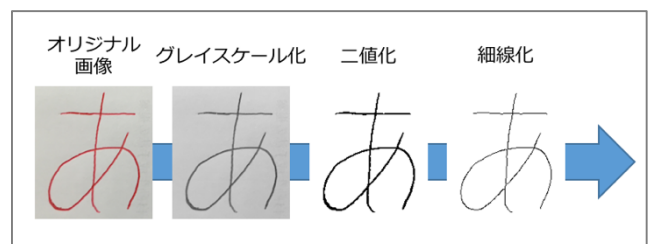


図3 画像処理の流れ

3.3 始点・終点・交差点の探索

細線化によって得られた画像に対して、端点（ストロークの始点，または終点）の探索を行っていく．まず，画像の左上からラスタスキャンを行うことで点の有無を調べ，点があった場合，その点の八近傍に点があるかを調べる．線は1ピクセルの太さで表現されているため，八近傍に1点しか点が存在しなかった場合は，その点は端点とみなすことができる．この方法で，それぞれの文字に対して端点集合を作っていく．ここで作成した端点リストは，

$$P_{edge}(x, y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

のように表す．また八近傍に3個以上の点があり，この点同士が隣接しない場合は，その点は交差点とみなすことができる．これらの交差点集合も同様に，

$$P_{cross}(x, y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

のように表す．

次に得られた端点は何画目のストロークの始点，終点，またはノイズなのかを判定する．ここでは，取得した端点とサンプルデータのストロークの始点，終点の距離の和が最小になるようなマッチングを行うことで判定する．具体的には，まずサンプルデータの文字とオフライン文字の位置合わせとサイズ合わせを行うため，それぞれの文字を矩形とみなしたときの重心と横幅を求める．ここで求めた情報をもとに2文字間の重心を重ね合わせ，横幅の大きさを合わせるように処理を行う．この結果をもとに，点同士の距離が最小値になるパターンを求めていき，

$$(x_n, y_n) = (x_{n'}start, y_{n'}start)$$

あるいは

$$(x_n, y_n) = (x_{n'}end, y_{n'}end)$$

のようにそれぞれの端点に情報を与える．なお，「の」のような交差点が本来の始点・終点になる場合もあるため，サンプルデータにおける端点の近辺にある交差点も，端点とみなすことで解決する．図4は実際に始点・終点のマッチングを行った結果の一部である．

なお，細線化によって図5のような交差点がずれてしまうといった問題があるため，求められた交点リストから近くに交点があるとき，それらを繋ぐものを交差ストロークとして定義する．

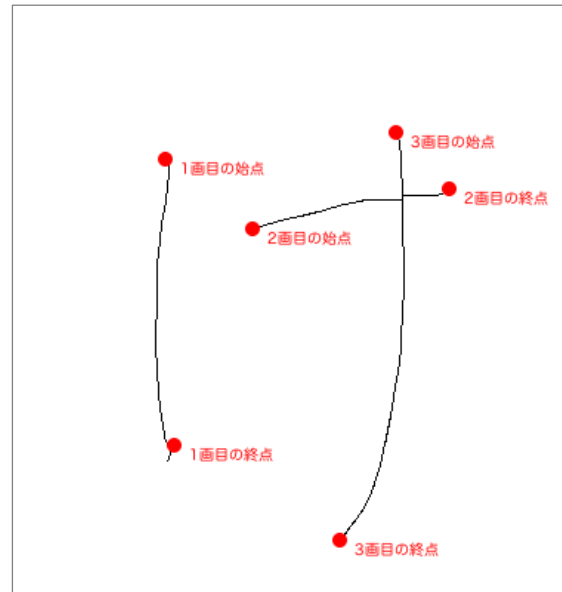


図4 サンプルデータとの始点・終点のマッチング

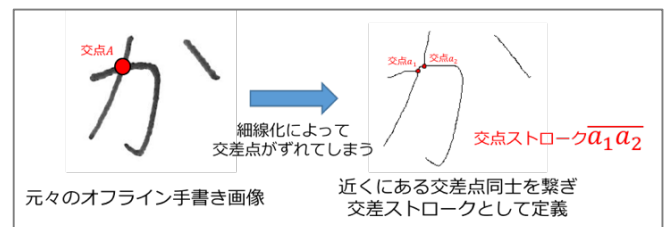


図5 細線化による交差点のずれ

3.4 ストローク認識処理

次に，細線化画像と前節で作成した始点・終点・交点の集合をもとに，どこが1つのストロークかを認識していく．まず，1画目の始点から順に，周辺の点列を辿っていき，1回通った点列は辿らないように，終点に辿り着く経路を求める．なお交差点や交差ストロークは2回辿る可能性があるため，これらに関してはもう一度辿れるような条件を与える．また分岐が生じた場合はどちらの経路も通るパターンを作成し，終点に辿り着いた時に，

$$P_{stroke}(x, y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

という点列情報を格納したストロークとして定義し，候補ストロークリストに追加する．これらの複数パターン候補の候補ストロークに対して，サンプルデータのストローク情報をもとに，実際のストロークと候補ストロークの距離の総和を求め，この値が最も低くなるようなストロークパターンを実際のストロークとして認識する．これを画数分繰り返し行い，各ストロークを認識していく．

3.5 ストローク数式化処理

点列によって構成されたストロークを数式化する手順に

ついて説明する。まず図6のように、取得した点列データに対して、その点を出来るだけ接続するように3次スプライン補間を行い、間を埋める点を生成する。次に、その補完された点を順に通る平面曲線の数式をフーリエ級数によって求める。これにより、手書きストロークを媒介変数表示で数式化し、数式の平均化によって平均文字を生成することが可能となる。なお、ここでスプライン曲線をそのまま利用しない理由は、スプライン曲線は制御点間ごとに数式を取り換える必要があり、平均化などへの応用を考慮すると、計算が複雑になると考えたためである。

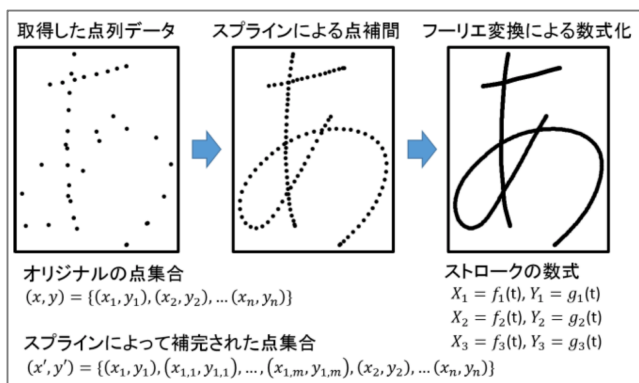


図6 ストローク数式化手法

次に数式化の手順について詳しく説明する。各文字のストロークの手書き入力に、スプライン補間を適用した点列の座標データを終点で折り返し、そのまま同じ点を通る形で始点まで点を増加させることで閉曲線の点列を作る。ここで閉曲線にする理由は、フーリエ級数によって数式化をする際に始点と終点が離れている場合、両端を繋ごうとして両端近辺で曲線が波打ってしまうためである。この点列を通る平面曲線の媒介変数表示を、

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases} \quad -\pi \leq t \leq \pi.$$

としたとき、 $f(t)$ 、 $g(t)$ は周期関数ではないが、

$$f(t) = f(t + 2n\pi) \quad n \text{ は整数}$$

と定義することにより周期関数とみなすことができる。さらに、文字の「角」も近似的に急な曲がり方をした滑らかな曲線とみなすことで $f(t)$ 、 $g(t)$ はフーリエ級数で表示可能である。すなわち、

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \sin nt).$$

と表すことができる。ここで、 $a_{i,n}$ と $b_{i,n}$ は

$$\begin{cases} a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cdot \cos nt \, dt \\ b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cdot \sin nt \, dt \end{cases}$$

で求めることができる。また、座標のデータは離散的であるが、上記の式は座標データが等間隔に並んでいるとすると、 $a_{i,n}$ と $b_{i,n}$ を求める積分を和で近似することができる。

この手法によって、媒介変数表示された平面曲線としての各ストロークの数式を得ることができる。

4. 評価実験

4.1 実験目的・内容

本章では、提案手法で実際にオフライン手書き文字のストロークを認識し、それらのストロークを数式化した際に、どの程度の精度で本来の手書き文字を数式で再現できているかについて評価実験を行う。評価の方法としては、元々のオフライン手書き文字画像と比べて、数式から再描画した文字がそのままの形状を保っているか、また元々のひらがなの書き順通りにストロークが込れているかを筆者が主観で評価していく。ここで正しく数式化できている場合は正解、誤った送り方や形状が全く異なる場合は不正解とすることで、各文字における数式化の精度を算出する。

4.2 オフライン手書き文字データの収集

評価用のオフライン手書き文字データセットを構築するため、「ひらがな 46 個」「漢字 98 個」「著者の苗字」の計 145 個からなるデータセット構築シートを作成した。

今回は特に「ひらがな 46 個」の数式化を行うことにより、その精度を検証する。本実験では、実験協力者は 5 人にはデータセット構築シートにボールペンやシャープペンを用いてマス目内に文字を書いてもらい、これらを 3 回繰り返し行ってもらった。なお、文字を書く際にはできるだけ一画一画をはっきりと、お手本を真似ないように書いてもらうことにした。また、文字の誤記に対応するためにプリントの右下に予備の空欄を用意した。ここで実際に集められた手書きデータをスキャンしたものの一部を図7に示す。

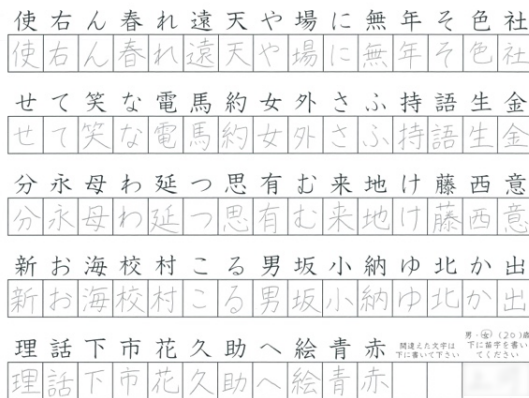
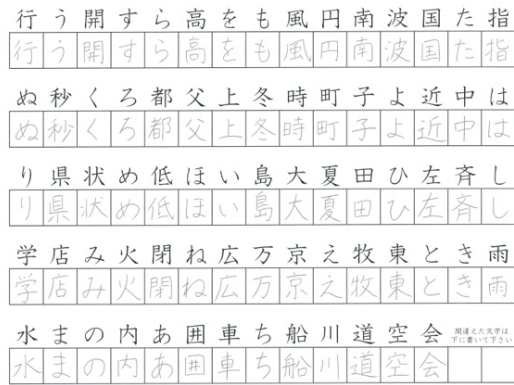


図7 数式化のために用意したプリント

また、1文字1文字を抽出するために、Processingに用意されている画像処理ライブラリOpenCVを用いて、輪郭を取得し、輪郭内部の領域の面積が一定の閾値以上のもののみを抽出することで、マス目の内部の文字抽出を行った。ここで得られたデータは、図8のようにPNGファイルとして出力される。

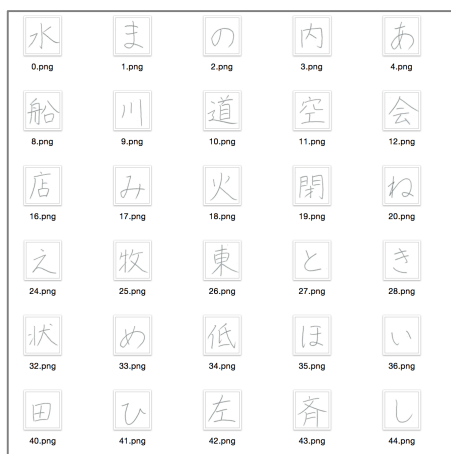


図8 文字抽出結果の一部

これらのオフライン手書き文字全てに対して、3章で述

べた手法をもとに数式化を行う。なお、この処理についてもProcessingとOpenCVを用いて行った。本処理により、5人それぞれが3回ずつ書いた46語(計690パターン)のオフライン手書き文字に対して数式化を行った。

4.3 結果

提案手法により、5人が3回ずつ書いたひらがなのストロークを数式化した際の、各ひらがなの精度は表1のような結果になった。なお、20%以下の精度だったものを青いセルで、80%以上の精度だったものを赤いセルで示している。この表より、平均70%の精度でストロークを認識し、数式化によって再描画できていることがわかる。しかし、その中でも「あ」や「ぬ」、「む」、「め」といった文字については、20%以下の精度となっており、ほとんど数式化できていないことがわかる。

表1 各ひらがなにおける数式化の精度

文字	認識率	文字	認識率
あ	0%	は	60%
い	93%	ひ	73%
う	100%	ふ	33%
え	33%	へ	100%
お	27%	ほ	67%
か	87%	ま	53%
き	87%	み	47%
く	100%	む	7%
け	100%	め	0%
こ	87%	も	87%
さ	100%	や	87%
し	100%	ゆ	53%
す	67%	よ	47%
せ	100%	ら	87%
そ	67%	り	87%
た	80%	る	80%
ち	100%	れ	73%
つ	100%	ろ	87%
て	93%	わ	87%
と	87%	を	73%
な	73%	ん	93%
に	87%	平均	70%
ぬ	0%		
ね	47%		
の	33%		

また、筆者が正解と見なした文字の例で、元々のオフライン手書き文字と、数式化した後面数ごとに色分けして再

描画した文字を左右に並べると図9のようになる。一方で、誤った経路を辿って数式化された文字の例は図10のようになる。

元々のオフライン手書き文字 数式化し再描画した文字



図9 正しく数式化されている例

元々のオフライン手書き文字 数式化し再描画した文字

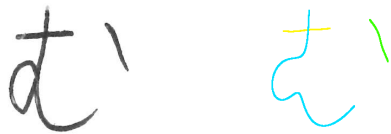


図10 誤った経路を辿って数式化されている例

また、ユーザごとの数式化の精度の結果を見るために、「ふ」を例としてそれぞれの精度を表したのが表2である。この表についても表1と同様に、20%以下の精度だったものを青いセルで、80%以上の精度だったものを赤いセルで示している。この表より、A、B、Cさんは「ふ」の精度が0%と全くできていないのに対し、Dさんは100%の精度で、3 試行全てのデータに対し、正しく数式化できていることがわかる。

表2 ユーザごとの精度の結果の違い

	A	B	C	D	E
ふ	0%	0%	0%	100%	67%

実際に Aさんが書いた「ふ」と、Dさんが書いた「ふ」を並べたものが図11である。図より、数式化ができていないAさんの「ふ」は、1画目と2画目がつづけて書かれており、対してDさんの「ふ」はサンプルデータの書き方と同様に1画目と2画目を分けて書かれていることがわかる。



図11 Aさんの(左図)とDさん(右図)の書き方の違い

4.4 考察

数式化が特に上手くいっていない「あ」や「ぬ」、「む」

「め」といった文字が共通して持つ特徴として、斜めに交差しているストロークを含んでいるという点が挙げられる。これは手書き画像に対して行った細線化のアルゴリズムの特性上、交差点が大きくずれることが原因であると考えられる。これに関しては、交差ストロークを定義するときの手法を、単なる交差点同士の距離が近いものを交差ストロークとするのではなく、サンプルデータの交差点の位置関係を用いることにより、解決することが可能だと考えられる。

また図10で示した「む」のように、間違っただけで経路を辿ってしまう原因として考えられるのは、サンプルデータをもとに、候補ストロークの中から実際のストロークを選出するとき、単にそれぞれの点列同士の距離の総和の比較のみで行っているのが原因だと考えられる。これに関しては、サンプルデータのストロークと候補ストロークの比較に使う特徴量を他にも与えることで解決できると考えられる。具体的には、点列同士の距離の総和に加え、ストロークの曲率が似通っているものは、実際のストロークである可能性が高いという条件を加えることで精度が上がると考えられる。

また表2で示したように、文字によってはユーザごとに数式化の精度に大きな違いがあることがわかった。これは図11に示したように、文字によってつづけ字で書く人とそうではない人の違いがあり、手書き文字の個性が原因であることがわかる。提案手法では、1つの文字に対して、1パターンの文字サンプルデータしか与えていないため、書き方の違いによって、正しく数式化できていないと考えられる。したがって今後は、「き」「さ」「そ」「ふ」などの人によって書き方が大きく分かれる文字に対しては、サンプルデータを何パターンか用意することで、数式化の精度を高くできるものと考えられる。

5. オフライン平均文字

香蘭女学校(東京都品川区)の協力を得て、先述のオフライン手書き文字のデータセット構築用シートを用い、中学1年生から高校2年生までの各学年4クラスの生徒約600人分のデータを収集した。

ここでは提案手法の有用性を示すために、実際に4章で数式化の精度が高かったオフライン手書き文字「う」、「け」、「せ」、「さ」について平均文字を生成し、ある程度大きなグループ(学年単位)で特性に差が出るのか、そして何人程度のデータを混ぜると平均文字は同一化するのかについて調査した。

5.1 学年ごとの平均文字の推移

学年が上がるにつれて平均文字に変化が見られるのかを検証するために、1学年約40人(各クラス10人をランダムに抽出)による平均文字を生成し、各学年の平均文字を比較した。なお、数式化が上手くいかなかったデータは除

外して、平均化を行った。ここで、実際に生成したそれぞれの平均文字を学年ごとに色分けし、重ねて表示したのが図 12 である。

図 12 より、学年ごとの平均文字は大きな字形の変化はないが、中学 1 年生の平均文字は大きく、高校 2 年生の平均文字は小さい傾向があることがわかる。このことから、中学 1 年生は字形がまだ定着しておらず、大きく文字を書いてしまっているのではないかと考えられる。その一方で、中学 2 年生からは字形が安定し、それ以降はさほど形状に変化がなかったのだと考えられる。また、高校 2 年生は受験期が近いということもあり、試験の回答スペースの影響などによって、文字を小さく書いてしまっているのではないかと考えられる。この点については、今後も大規模なオフライン手書き文字データを用いることによって、より詳細に実験および分析を行う予定である。

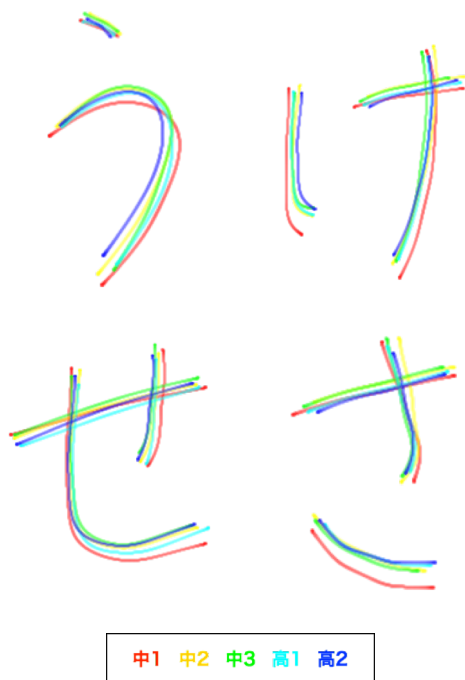


図 12 学年ごとの平均文字の比較

5.2 平均文字の収束

平均化する人数が多くなるにつれて、平均文字の形状は収束するのかを検証するために、100 人による平均文字を 2 グループにおいて生成し、さらにそれらを平均化した 200 人による平均文字を生成した。なお、5.1 節で述べた通り、学年による違いは存在する。そこで各グループの学年に偏りが出ないように、それぞれに均等になるようにデータを振り分けた。こちらも前節と同様に、数式化が上手くいかなかったデータに関しては、平均化をする際に除外した。ここで、2 グループの各平均文字を赤と青に色分けし、重

ねて示したのが図 13 である。また約 200 人の平均文字が図 14 である。

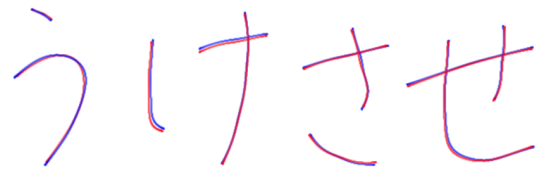


図 13 2 パターンの約 100 人の平均文字

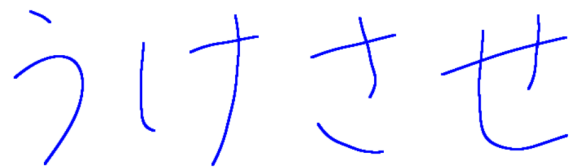


図 14 約 200 人によるオフライン手書き平均文字

図 13 が示す通り、それぞれのグループの字形にほとんど違いがない。また図 14 の約 200 人による平均文字についても、これらの 2 グループの平均文字と変わらないことがわかる。このことから、ある程度の手書き文字データを集めれば、その平均文字は収束するということが考えられる。そこで、今後は 100 人の平均文字のパターンを何通りか作り、それらを比較することで明らかにしていく。

また、このような分析が行えることから、オフライン手書き文字を数式化することは、大規模なデータセット構築をする上でも極めて有用であると考えられる。

6. まとめと今後の展望

本研究では、手書き入力システムに依存にしない、オフライン手書きストロークの数式化手法を提案した。また、本手法でどの程度の精度でひらがな 50 音が数式化可能かを検証するため、評価実験を行った。この結果、平均 70% の精度で数式によって文字を再描画できることがわかった。また、本研究の有用性を示すために、オフライン手書き文字の平均化を行い、それらをもとに学年ごとの平均文字や、大人数における平均文字の分析を行った。

今後は、4.4 節で述べた改善手法をもとに再実装することで、数式化の精度を上げることを検討している。また今回はひらがなのみを扱ったが、今後はひらがなだけでなくカタカナ・漢字・イラストに対しても手書きの数式化を行なっていく。さらに、今回我々が実施したデータセット構築のように、手軽な手書きデータの収集が可能になることから、大規模な手書き文字データセット構築を行い、様々な分析を行っていきたいと考えている。分析の一例としては、ペンタブレットで書いたオンライン手書き文字と、紙に書いたオフライン手書き文字の字形に違いがあるのかと

いったような比較実験などを予定している。なお現在、香蘭女学校の600人ほどのオフライン手書き文字データを収集しているので、今後このデータをもとに分析することを検討している。

謝辞 本研究の一部は、JST CREST, JST ACCEL, 明治大学重点研究 A, 香蘭女学校の支援を受けたものである。

参考文献

- [1] 内閣府: 平成 26 年度「国語に関する世論調査」の結果について, 2014,
http://www.bunka.go.jp/tokei_hakusho_shuppan/tokeichosa/kokugo_yoronchosa/pdf/h26_chosa_kekka.pdf.
- [2] 中村聡史, 鈴木正明, 小松孝徳. 平均文字は美しい, エンタテインメントコンピューティングシンポジウム 2014, (EC2014).
- [3] 中村聡史, 鈴木正明, 小松孝徳. ひらがなの平均手書き文字は綺麗, 情報処理学会論文誌 エンタテインメントコンピューティング特集号, Vol. 57, No. 12, pp. 2599-2609, 2016.
- [4] 新納真次郎, 中村聡史, 鈴木正明, 小松孝徳. 平均図形も美しい, エンタテインメントコンピューティングシンポジウム 2015, (EC2015).
- [5] 新納真次郎, 中村聡史, 鈴木正明, 小松孝徳. コミック作成における平均描画手法の可能性, 第 30 回人工知能学会全国大会, (JSAI2016).
- [6] 篠沢佳久. 自己相似性特徴を用いたオフライン手書き文字認識の提案, 情報処理学会論文誌, Vol. 46, No. 3, pp. 859-869, 2005.
- [7] 安倍広多, 柴山守. 計算幾何学を用いた毛筆文字画像からのストローク抽出, 電子情報通信学会技術研究報告, 2006.
- [8] Tong Huang, Makoto Yasuhara. Recovery of Information on the Drawing Order of Single - Stroke Cursive Handwritten Characters from Their 2D Images, 情報処理学会論文誌, Vol. 36, No. 9, pp. 2132-2143, 1995.
- [9] 加藤 美治, 保原 信. 手書き文字の筆順復元, 全国大会講演論文集 第 56 回(人工知能と認知科学), 1998.
- [10] 漢字の正しい書き順(筆順), <http://kakijun.jp/>.
- [11] 大津展之. 判別および最小 2 乗基準に基づく自動しきい値訴定法, 電子情報通信学会論文誌, Vol. J63-D, No.4, pp. 349-356, 1980.
- [12] T. Y. ZHANG and C. Y. SUENA. Fast Parallel Algorithm for Thinning Digital Patterns, Communications of the ACM CACM Homepage, 1984.