

自他との平均化により 手書きをきれいにするシステムの提案

又吉 康綱 久保田 夏美 齊藤 絢基

大島 遼 中村 聡史 鈴木 正明

明治大学 総合数理学部 先端メディアサイエンス学科

yasutsuna.matayoshi@gmail.com

概要 ペン入力可能なコンピュータが普及しているが、その手書き入力の良さが活かされていないことが多く、電子ノートや電子黒板における手書きは従来より大きな変化はない。そこで本研究では、ひとは平均化された自身の手書き文字や他者と融合した手書き文字を高く評価するという特長を生かし、ユーザの手書き文字を自身の過去の手書き文字や他者の手書き文字とリアルタイムに平均化することによって、電子ノートや電子黒板といったペン入力可能なデバイスにおける手書きをより良いものとする手法を提案する。また、提案手法をウェブシステムとして実装し、その可能性について検討を行う。

キーワード 手書き文字, 平均文字, ノートテイク, 電子黒板

1 はじめに

学習において、講義のノートやメモをとるという行為は欠かせない。ここで、ノートやメモをとる方法としては、ノートなどに手書きするということと、キーボードを使ったタイピングによって情報端末機器に記録するという方法が一般的である。手書きの利点は、複雑な図形や数式も思い通りに書けるという柔軟性にあり、キーボードを使ったタイピングの利点は、文字を書くのが得意でなくてもフォントを用いることで、きれいにノートやメモをとることが可能になることにある。ここで、ズルキフリーらの情報系大学院生を対象とした実験[1]によると、単純な文字入力だけであっても、キーボードを使ったタイピングに比べ、手書きの記入の方が早く入力できるばかりか、記憶して多く入力できることを明らかにしている。このようなことから、手書きでのノートやメモはいまだ重要であるといえる。

一方で、手書きで文字を書くことに苦手意識を感じ、ノートを書くのをためらう人もいる。ゼブラ株式会社が行った調査[2]によると、自身の手書き文字を恥ずかしく思う人の割合は64%存在することがわかっている。このように恥ずかしく思う人が多い理由としては、「自身が書いた文字をきれいだと思わずに、他者へ自身の文字を見せることに抵抗を感じた経験があるから」と考えられる。

ここで中村ら[3]は、ひらがなを対象とした手書き文字に関する実験によって、ひとは書写技能によらず自身の

文字を他者より高く評価することを明らかにしており、必ずしも自身の手書き文字に対して悪い評価をしていないことがわかる。また、自身のオリジナルの手書き文字よりも、複数回書いたものを平均化した文字を高く評価することを明らかにしている。さらに齊藤ら[4]は、自身と他者の手書き文字を平均化したものは、好感度が高くなるということを明らかにしている。そこで我々は、ユーザの手書き文字を、自身の過去の手書き文字と融合したり、他者の手書き文字と融合したりすることによって、きれいで好感度の高い文字へと変換および提示することにより、文字を書くことに自信のない人でも手書きでノートやメモを積極的にとる手法を提案する。

ここで Zitnick[5]は、入力した手書きストロークの曲率をリアルタイムに計算し、これまでに書いたストロークとの一致度を求め、一致度が高いものを集めて平均化することで、ユーザ個人の手書き文字をきれいにする手法を提案している。しかし、Zitnickの手法はストローク単位(一画単位)で処理を行うものであり、英数字などの単純な曲線集合の文字ではきれいにするのが可能であるが、漢字などの画数が多く、位置関係のバランスを取ることが重要な複雑な文字には適用できない。また、同一人物のストロークは曲率が類似するという前提としているため、曲率が異なる他者との融合には適していない。

そこで我々は、ユーザの手書き文字認識を行い、その手書き文字を数式化として蓄積するとともに、これまでに蓄積されている数式化された手書き文字と平均化することによって、複雑な漢字でもきれいにでき、他者と

融合することで好感度が高い手書き文字を提示可能とする手法を提案し、プロトタイプシステムを実装する。本システムにより、自身が書いた手書き文字を違和感なくきれいにすると同時に、好感度が高い文字にすることで、文字に自信がないユーザでも抵抗感を感じずにノートやメモをとることができる。と期待される。

2 関連研究

手書きを活かした情報端末機器でのノートテイクに関する研究については様々なものがある。平山ら[6]は、手書き入力のコミュニケーションをより良いものにするため、リンク機構を用いながら複数人がリアルタイムに手書きを共有可能なコラボレーションツールを実現している。また辰川ら[7]は、テキスト入力にはキーボードを使い、図形の描画にはペンを使うノートエディタを開発している。これによってキーボード入力によるテキストの読みやすさや編集のしやすさ、ペン入力による多様な図形の描画など、双方の入力の利点を活かしたものとなっている。しかし、手書きを苦手としている人はこのようなシステムを使用する意識が低いといった問題がある。我々は手書きを支援することで、積極的にペン入力可能な情報端末でのノートやメモを取るようになることを目指す。

手書きを支援する方法としては 2 種類あり、一方は手書きを認識し、フォントや整形された図形に変換するもの、もう一方は手書きを残しつつブレを軽減するものである。

手書きをフォントや整形された図形に変換するシステムとしては Nebo[8]などがあり、ユーザの手書き文字や数式をリアルタイムに認識し、テキストにすることでノートをきれいにすることを可能にしている。また AutoDraw[9]は、AI を用いてユーザが入力した手書きイラストを推定し、きれいに整形された図形を複数提案することで、その中からユーザが選択したものを入れ替えつつスケッチ可能とするシステムである。Speech pen[10]では、音声認識と手書き文字認識を組み合わせることで、講義やプレゼンテーションにおける手書き文字入力を効率化することを支援している。

書き手の筆跡を残しつつ手書きをきれいにする支援を行う研究としては、はじめにでも説明したように、Zitnick[5]が、ストロークの曲率の類似を利用して、過去に書いたストロークと現在のストロークとの平均化処理を行うことによって、より良い手書きを可能とする手法を実現している。また Niino ら[11]は、手描きイラストの特性を明らかにするとともに、既存のペイントシステムにあるような基本的な機能に合わせ、手書き文字を平均化する手法を応用することによって、ユーザが描いた絵のストロークを自動で平均化し、絵を描くのが苦手な人でもきれいな図形を描くことを可能としたプロトタイプシステムを

実装している。さらに DrawFromDrawings[12]は、大規模スケッチデータベースをもとにユーザが描いている絵を推定し、データベース上の絵とユーザが描いた絵を融合することで、きれいな絵の描画を可能としている。一方、dePENd[13]は、ボールペンの強磁性を用いて、机内部に設置された磁石をコンピュータで制御し、ボールペンを誘導することで、手書き文字をきれいにすることや、手書きのコピーアンドペーストなどを可能とするシステムを提案している。

以上のように、手書きを支援する研究は多くなされているが、漢字のような複雑な手書きを手軽に平均化するには適していない。そこで我々は書いた文字の文字認識を行い、書き手の筆跡の特徴を残しつつ、手書きをきれいにする手法の実現を目的とする。

3 提案手法

本稿では、ユーザの手書き文字を違和感なく自他の文字と融合することによってきれいで好感度が高い文字の提示を行う。そのため、図 1 のように、ユーザが書いた手書き文字を認識して手書き文字データベースに数式として格納するとともに、融合対象とする手書き文字の数式をデータベースから取得し、本章で説明する平均化手法を用いることによってアニメーションさせながら融合し、提示する。

なお、ユーザがどのような手書き文字を入力しようとしているのかは、ユーザがその文字を入力し終わるまでは判断がつかない。そこで、手書き文字の入力中に融合及び提示するのではなく、手書き文字を入力した後に認識し、あとから徐々に融合および提示することで手書きをきれいにしていく。

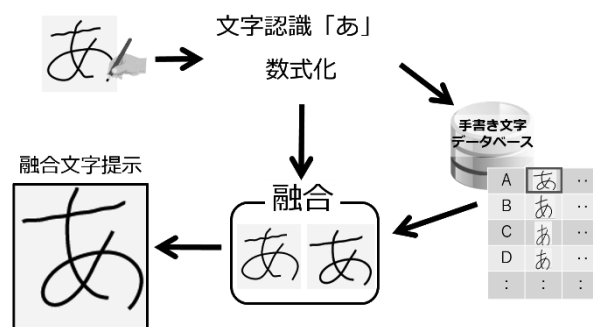


図 1 提案手法の概要

3.1 手書き文字の数式化

自身の手書き文字をきれいにするために、中村ら[3]の手書き文字平均化手法を応用する。ここでは、ペンによって入力されたストロークを点列データとして取得する。次に、その点をできるだけ接続するように3次元スプライン補間を行うことで、点間の距離を狭くする。これは、入力された点列データをそのまま用いると点の数が少なく、なめらかな曲線にならないためである。その後、スプライン

ン補間を適用した点列の座標データを辿っていき、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のは、フーリエ級数展開によって数式化する際に始点と終点が離れていた場合、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。次に、この点列をフーリエ級数展開により t を媒介変数とした下記の式により表現する。

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

ただし $f(t)$ は

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

と表すことができる ($g(t)$ も同様に考えることができるため省略する)。ここで、 a_n と b_n は

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

の式で求めることができる。なお、座標のデータは離散であるが、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_n と b_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画の数式を得ることができる。

なお、数式化された手書き文字はそれぞれ書く場所が異なるため、平均化を行うためには書く場所をそろえる必要がある。そこで、まず対象とする手書き文字の x 座標、 y 座標の最小値、最大値を求め、最小値と最大値の平均をとることによって、文字の中心座標を求める。その後、数式から文字の中心座標を減算することにより文字の中心を $(0, 0)$ とする。

3.2 手書き文字の融合

ユーザの手書き文字を自他の文字と融合提示するには、そのユーザの書いた文字がどのような文字であるかを認識する必要がある。そこでユーザが手書き文字を入力するたびに 3.1 節の手法で数式化するとともに、点列をもとにして文字認識を行なう。次に、認識された文字とその画数をもとに自身または他者の同一文字の数式の集合を取得し、ユーザが直前に書いた手書き文字と平均化することによって融合文字の数式を生成する。

ここで自身または他者が M 回書いたある文字の i 画目のストロークを

$$(x, y) = \left\{ (f_{i,1}(t), g_{i,1}(t)), \dots, (f_{i,M}(t), g_{i,M}(t)) \right\}$$

とし、自身が入力したその文字の i 画目を

$$(x, y) = (f_{i,last}(t), g_{i,last}(t))$$

としたとき、一方に α を、他方の式に $1 - \alpha$ を掛け合わせることによって、任意の $\alpha (0 \leq \alpha \leq 1)$ による加重平均化処理を以下の式で行う。

$$\begin{cases} x = \alpha \sum_{k=1}^M \frac{f_{i,k}(t)}{M} + (1 - \alpha) f_{i,last}(t) \\ y = \alpha \sum_{k=1}^M \frac{g_{i,k}(t)}{M} + (1 - \alpha) g_{i,last}(t) \end{cases} \quad 0 \leq t \leq \pi$$

なお、ここで求めた融合文字の数式をそのまま使用すると、ユーザの書いた文字と異なった大きさや位置で描画されてしまう。そのため、求めた数式を入力された点列データの大きさや座標に合わせて補正を行う。具体的には、ユーザの書いた場所に融合された文字を提示するため、ユーザが書いた文字の縦横の大きさをもとに、数式の係数を修正する。

3.3 融合手書き文字の提示

ユーザが書いた手書き文字と自他のこれまでの手書き文字との融合において、手書き文字を融合した文字へと入れ替えるだけでは違和感が生じる。そこで、文字認識後に手書き文字から融合後の手書き文字へとスムーズにアニメーションするよう、下記のように表示する数式を切り替えていく。

ここで、ユーザが最後に書いた手書き文字の i 画目の数式を $(f_{i,last}(t), g_{i,last}(t))$ 、ユーザが書いた手書き文字に対応する融合後の手書き文字の i 画目の数式を $(f_{i,after}(t), g_{i,after}(t))$ としたとき、下記の式の β の値を 0 から 1 に変化させることによってアニメーション表現する。

$$\begin{cases} x = (1 - \beta) f_{i,last}(t) + \beta f_{i,after}(t) \\ y = (1 - \beta) g_{i,last}(t) + \beta g_{i,after}(t) \end{cases} \quad -\pi \leq t \leq \pi$$

この β の値の変化速度および変化方法により、アニメーションのスタイルを変更することが可能となる。

4 プロトタイプシステム

4.1 実装

手書き文字認識においては、どこからどこまでのストロークが 1 文字なのかを認識する必要がある。ここで、入力されたストローク群の近接度合いなどで判断することも考えられるが、字同士が近接している場合の認識が難しい。そこで本プロトタイプシステムでは簡単のため、入力されるストロークとストロークの間の時間を利用し、その時間がある一定以上である場合に別の文字を書いていると判断するようにした。また、手書き文字は 100% の精度で認識できるわけではない。そこで、その文字の認識確信度がある一定以上である場合に文字として認識し、融合するものとした。つまり、ある一定以下の認識確信度である場合は、そのままの文字として提示するこ

とした。

プロトタイプシステムを Web アプリケーションとして実装した (<http://mojirage.com> から利用可能)。実装には JavaScript, PHP, MySQL を用いた。また、手書き文字の描画には SVG (Scalable Vector Graphics) を用い、文字認識については MyScript[14] の文字認識 API を利用した。

クライアントサイドは JavaScript で実装しており、ユーザが入力する点列の取得、フーリエ級数展開、次の文字を書くまでの時間間隔を調整するスライダーの制御、MyScript[14] の文字認識 API や融合化を行うサーバへの非同期的な送受信を行う。

サーバサイドは PHP および MySQL を用いて構築しており、入力された点列データ、フーリエ級数展開によって得られた数式、文字認識結果やユーザ情報などを元に提案手法を用いて融合化の処理を行うようにした。システム上、融合する際の平均を導く処理を高速化するため、これまでの平均の数式もキャッシュとしてデータベースに保存しておき、その数式と直接融合を行えるようにした。また、ユーザの書いた点列データやページはデータベースに格納するようにした。

手書き文字のアニメーションについては、SVG と CSS を用い、前後の数式と対応するように変形させる。なお、ユーザが入力した手書き文字から融合文字へのアニメーションは 1 秒間とした。なお、アニメーション例を図 2 に示す。



図 2 アニメーション例(一番左がオリジナルで、一番右がアニメーション後のもの)

4.2 システムの利用方法

ユーザが Google アカウントを利用してシステムにログインをすると、これまでに書いたページが一覧で表示される(図 3)。右上の「+」をクリックすることでページの新規作成を行い、ノート編集画面へと遷移することができる。また、ページをクリックすると編集モードになり、入力や削除などの編集が可能になる。

編集モードでは、背景に薄い罫線が入ったページが表示される。ここでユーザがストロークを入力している際には、ユーザ自身のストロークの点列を提示し、提案手法による融合処理を行ない、しばらくするとアニメーションによって変形させながら融合された文字の提示を行う。また、ページの保存ボタン、Undo ボタン、文字の太さ指定ドロップダウン、認識までの時間指定スライダーを利用することにより、ノートの編集を自在に行えるようにし

ている。

また、上部のメニューからユーザが書いた文字を自身の手書き文字と融合するのか、他者の手書き文字と融合するかを設定することができる。ここで他者との融合を行う場合、「他者の書体を探す」ボタンを選択すると、各ユーザが最も多く書いている文字の上位 5 件が提示される。ユーザはその中から他者の書体を選択し、ページ編集画面へ進むことで、指定したユーザとの融合文字を書くことができる。なお、他者との融合を行う場合には、加重平均の割合を指定できるようにスライダーを設けた。図 4 はページ編集画面である。



図 3 ページ一覧画面

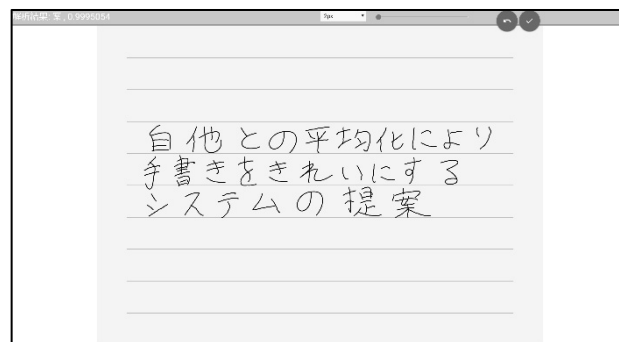


図 4 ページ編集画面

5 利用テストと考察

ユーザの書いた文字が、過去に書いた自身や他者の文字と融合され提示された際に、その手書き文字に対してどのような印象を受けるのかをシステムの利用テストをもとに検証し、考察する。

利用テストでは、21 歳から 23 歳の学生 5 人にシステムを用いて指定の文章を書いてもらい、フィードバックをもらった。利用テストでは、自身との融合と他者との融合の両者について試してもらった。なお、入力端末には iPad Pro (12.9inch) および Apple Pencil を使用した。また、文章はフォントの書体見本でよく用いられる宮沢賢治作の「ポラーノの広場」[15] の 1 節を用いた。

自身との融合による調査では、協力者にシステムを用いて 3 回ずつ提示される文章を書いてもらった。その

融合前 あ の イーハトーヴォ の すきとおった風、

融合後 あ の イーハトーヴォ の すきとおった風、

図5 ある実験協力者の自身との融合前後の字形

融合前 あ の イーハトーヴォ の すきとおった風、

融合後 あ の イーハトーヴォ の すきとおった風、

図6 ある実験協力者の他者との融合前後の字形

融合前 あ の イーハトーヴォ の すきとおった風、

融合後 あ の イーハトーヴォ の すきとおった風、

図7 ある実験協力者の他者との融合前後の字形

際、初めの1回目は数式化による影響のみだが、2回目は1回目に書いたものとの融合、3回目は1,2回目との融合を行う。また、各回の融合前と融合後の文字を見比べてもらい、システムや文字に対するコメントを貰った。

また他者との融合による調査では、協力者に他者との融合を融合比率50%でそれぞれ1回ずつ書いてもらった。ここでは、他の協力者が自身との融合の調査にて3回ずつ書いたものから作成した平均文字を用いた。なお、融合する対象は、協力者個人に選んでもらった。最後に他者との融合前と融合後を見比べてもらい、システムや文字に対するコメントを貰った。

図5は、ある協力者の3回目の自身との融合前と融合後の字形を比較したものである。全体的に線の細かなブレが少なくなると同時に、ストローク間の隙間が整うことによってきれいな字に変化していることが確認できる。また、協力者からは「平均化された漢字が特にバランスがよく、好きだった」、「やたら大きく書いた文字でも、平均化することでいい大きさになった」など、肯定的な意見が多かった。中村ら[3]の研究結果では、ユーザが実際に書いた文字よりも、自身の平均文字の方を高く評価すると主張しているが、同じような傾向が明らかになった。一方、「字のバランスなどの違いを感じなかった」との意見も得られた。これは、「筆跡の恒常性」に起因するものであり、ブレがない文字を書く協力者には、平均化が限定的なものになっているものだと考えられる。

図6,7は、それぞれある協力者の他者との融合前と融合後の字形を比較したものである。他者との融合は、自身との融合よりも字形が大きく変化している。例えば、「す」を比べてみるとバランスが大きく変化していることがわかる。協力者からは、「自身の字がすごい勢いで変換されて、自身の字の汚さを実感し、融合後の字が好きであると感じた」や「他者と融合することで、いつも上手く書けなかった文字も納得いく文字になった」などの肯定的な意見が得られた。一方、「融合相手が気に入らなかった」や「色んな人の書体がほしい」、「縦横の比率が維持されないのが違和感」といった意見も得られた。

利用テストにおける観察から、斉藤ら[4]の研究により明らかになっていた、自身と自身よりも文字がきれいな他者の手書き文字を融合することで、ユーザにとって好感度が高い文字が生成される現象を確認することができた。なお、元からきれいな字を書くユーザが、自身よりもきれいではないと判断するユーザと融合した場合には、あまり良い結果となっていなかった。また、融合するユーザ間の書き順が異なる文字の融合を行うと大幅に文字が崩れてしまっていた。これは、それぞれの画数同士に対して融合の処理を行っているためである。これらの問題を防ぐためには、システム側で書き順判定を行い、画数の入れ替えを行うことが考えられる。

また、著者らによる講義のノートをとるなどといったプロタイプシステムの利用から、本システムが講義を記録するうえで問題がなく、手書きを楽しむことができていた。

一方、同じ文字でありながら画数が異なる場合に、融合後の文字が崩れるといった問題があった。この問題については、融合することによって形が大きく崩れてしまうようなものは違うものとして保存するなどの工夫が必要になると考えられる。また、現在は文字認識を実行するために、ストロークとストロークとの間の時間を利用しているが、急いで書いているときなどには文字が分割されず、まとめて認識されてしまうという問題もあった。この点については、認識後に分割するなどの方法で対応する予定である。

融合対象によってその手書き文字の満足度が大きく異なり、一般的にはきれいな字の人と融合されると満足度が高くなるが、字が綺麗でなくても異性のユーザの字と自動融合された場合に、字がやわらかく・かたくなり、他者とノートと一緒にとっている感覚が得られることがあった。つまり、アイドルなどの手書き文字データベースを構築することができれば、ユーザの手書き文字が自動的にアイドルの手書き文字と融合されていくため、アイドルとともにノートを手書きしているような新たな体験を創出することができるのではと期待される。

6 まとめと今後の課題

本稿では、ユーザが書いた手書き文字を自他の文字と融合して提示する手法を提案し、プロトタイプシステムを実装した。また、利用テストで得られた意見から、システムの可能性について検討を行った。

提案手法やシステム全体について肯定的な意見が多く寄せられた一方で、以下のような問題点も明らかになった。

- 同じ文字を書いても文字認識の結果が異なることで違う文字と判定され、融合されないことがある。
- 1文字を確定させるために毎回、数秒間待たなければいけないのが負担になる。
- 書き順が異なるユーザ間で融合を行うと融合後の文字が崩れてしまう。
- 自身よりも文字がきれいではない他者と融合してしまうと、好感度が上がらず文字のバランスが崩れてしまう。

今後は、これらの問題点を中心に改善を進め、ユーザが自他との平均化によりきれいにノートやメモをとることができるのかの実験を行う。また、スマートフォンなどの画面が小さなモバイル端末や、電子黒板などの大きなディスプレイ上でも快適に動作するように改良していく。さらに、文字認識だけではなく、絵や図形などにも活用し、検証を行う予定である。さらに、色々なユーザの手書き文字を蓄積していくことにより、ノートの手書きの新たな体験を創出する予定である。

謝辞

本研究の一部は、JST ACCEL (Grant 番号 JPMJAC1602), 明治大学重点研究 A の支援を受けたものである。

参考文献

- [1] ムハマドズルキフリー, 田野俊一, 岩田満, 橋山智訓: 日本語のメモ書き作業における手書き入力の有効性, 電子情報通信学会論文誌, D, 情報・システム J91-D(3), 771-783, 2008.
- [2] ゼブラ株式会社, 「キレイ文字調査」実施。小・中・高校で実施した「キレイ文字教室」の動画を公開, <http://www.zebra.co.jp/press/news/2015/0403.html>, (2017.6.3 確認)。
- [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, 57, 12, pp. 2599-2609, 2016.
- [4] 斉藤絢基, 新納真次郎, 中村聡史, 鈴木正明, 小松孝徳: 手書き文字に対する書き手識別と好感度に関する調査, 研究報告ヒューマンコンピュータインタラクション (HCI), 2016-HCI-169, 6, pp. 1-8, 2016.
- [5] Zitnick, C. L.: Handwriting Beautification Using Token Means, ACM Transactions on Graphics (TOG) - SIGGRAPH 2013 Conference, Volume 32, Issue 4, No. 53, 2013.
- [6] 平山拓朗, 丸山一貴, 寺田実: Web ブラウザを用いた手書きメモ共有システムの提案, 情報科学技術フォーラム講演論文集, 11, 3, pp. 401-404, 2012.
- [7] 辰川肇, Nigel WARD: キーボードとペンを併用するノートエディタの研究, 情報処理学会研究報告コンピュータと教育 (CE), 2002, 17(2001-CE-063), pp. 67-74, 2002.
- [8] MyScript Inc., Nebo, メモをとるのに最も便利な手段 | MyScript, <http://www.myscript.com/nebo/>, (2017.6.3 確認)。
- [9] Google Inc., AutoDraw, <https://www.autodraw.com/>, (2017.6.3 確認)。
- [10] Kurihara, K., Goto, M., Ogata, J. and Igarashi, T.: Speech pen: predictive handwriting based on ambient multimodal recognition, CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 851-860, 2006.
- [11] Niino, S., Hagiwara, N., Nakamura, N., Suzuki, M.M. and Komatsu, T.: Analysis of Average Hand-drawing and Its Application, Intertain, (to appear), 2017.
- [12] Matsui, Y., Shiratori, T. and Aizawa, K.: DrawFromDrawings: 2D Drawing Assistance via Stroke Interpolation with a Sketch Database., IEEE Transactions on Visualization and Computer Graphics, Volume 23, Issue 7, pp.1852-1861, 2016.
- [13] Yamaoka, J., Kakei, Y.: dePENd: augmented handwriting system using ferromagnetism of a ballpoint pen, UIST, pp. 203-210, 2013.
- [14] MyScript Inc., MyScript Cloud | Documentation, <http://doc.myscript.com/MyScriptCloud/3.2.0/myscript-cloud.html>, (2017.6.3 確認)。
- [15] 青空文庫, 宮沢賢治 ポラーノの広場, http://www.aozora.gr.jp/cards/000081/files/1935_19925.html, (2017.6.6 確認)。