

特徴点を用いた手書き平仮名文字上達支援システム

森 悠人, 戴 瑩

岩手県立大学ソフトウェア情報学部

1. はじめに

自分の書く文字に自信のある人は約 2 割しかない¹⁾。そのために、手書き文字上達の支援を行うアプリケーションが存在する。しかし、いずれのアプリケーションもタブレットに書いた文字を評価するもので、実際に紙に鉛筆やペンで書いた文字を評価するものではない。

そこで本研究では、実際に紙に鉛筆やペンで書いた文字に AKAZE 特徴検出器を用いて、特徴検出、特徴マッチング、類似度判定、不一致部分の推定を行い、手書き文字の上達を支援するシステムを構築する。

2. 提案手法

2.1. システム概要

本研究では、比較するための見本平仮名文字をサンプルデータ、任意の被験者から得られた手書き平仮名文字をユーザーデータとして用いる。それぞれ一文字ずつに対して AKAZE を使用し、特徴点と一致度により手書き文字の美的評価を行いユーザーの手書き文字の特徴不一致部分を推定し、手書き文字の上達支援をする。システム構成図を図 1 に示す。

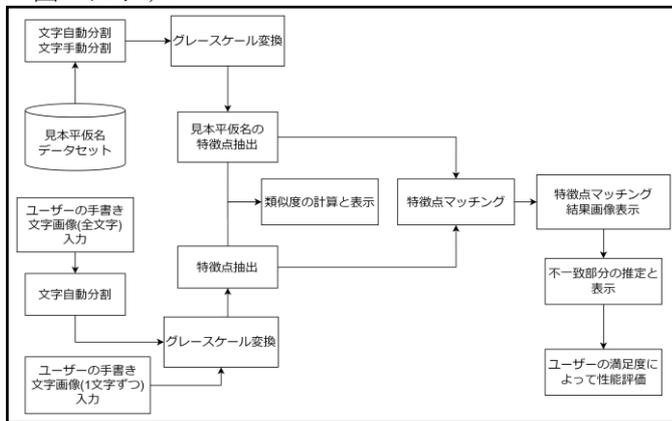


図 1 システム構成図

2.2. データセット

比較対象の見本となる平仮名は、Web サイトで提供されているものから 5 種類を収集し、一文字ずつ分割し見本平仮名データセットとし、これをサンプルデータとして利用する。任意の被験者 11 名から得られた手書き平仮名文字をユーザーデータとして利用する。ユーザーデータは 1 枚の

紙にすべての文字を書いたものと、1 枚の紙に 1 文字のみを書いたものの 2 種類があり、それぞれをスマートフォンのカメラで撮影したものと、プリンターでスキャンしたものを利用する。

2.3. AKAZE 検出器

OpenCV の AKAZE²⁾を用いて特徴検出・マッチング・類似度判定・不一致部分の推定の流れを図 2 に示す。



図 2 流れ図

2.3.1. 特徴点検出

AKAZE を生成し、あらかじめグレースケール変換、サイズ変換したサンプル画像とユーザー画像に AKAZE を適用し、それぞれの特徴点を検出する。

2.3.2. 特徴マッチング

検出した両方の特徴点群データを双方向で総当たりマッチングする。マッチング結果をすべて表示する。

2.3.3. 類似度の計算

総当たりマッチングの結果から、特徴量の距離を出し、平均を取ることで求める。平均値の数字が小さい方が類似度は高い。

2.3.4. 不一致部分の推定

総当たりマッチングの結果を特徴量の似ている順にソートし、その結果の下位 10 個の座標を取り出し、不一致部分としてサンプル画像に円を描画する。マッチングの結果が 10 個以下のときはその数すべてに円を描画する。

3. 実験の準備と結果

3.1. 実験の準備

5 つのサンプルデータの「あ」と、11 人から得たユーザーデータの「あ」について類似度判定を行い、その結果からユーザー 1 人 1 人について、最も類似度が高いサンプルデータを選択し、そのサ

Support for improving handwritten hiragana characters using feature points

Yuto Mori and Ying Dai

Faculty of Software and Information Science, Iwate Prefectural University

サンプルデータをユーザーそれぞれに割り当てた。その結果を表 1 に示す。

表 1 サンプルデータの割り当て

ユーザー1	ユーザー2	ユーザー3	ユーザー4	ユーザー5	ユーザー6
サンプルデータA	サンプルデータD	サンプルデータA	サンプルデータA	サンプルデータA	サンプルデータD
ユーザー7	ユーザー8	ユーザー9	ユーザー10	ユーザー11	
サンプルデータD	サンプルデータA	サンプルデータA	サンプルデータA	サンプルデータA	

3.2. 実験結果

3.2.1. 特徴検出結果

サンプルデータ A とユーザー1 の「あ」の検出結果を図 3 に示す。それぞれ、サンプルデータ A(左), ユーザー1 スマートフォンで撮影したもの(中), ユーザー1 プリンターでスキャンしたもの(右)の配置である。

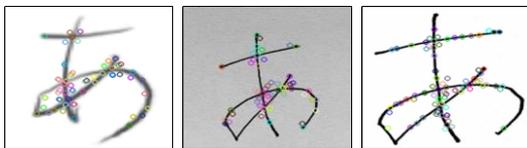


図 3 サンプルデータ A とユーザー1 の特徴検出

3.2.2. 特徴マッチング結果

サンプルデータ A とユーザー1 の「あ」の特徴マッチング結果を図 4 に示す。

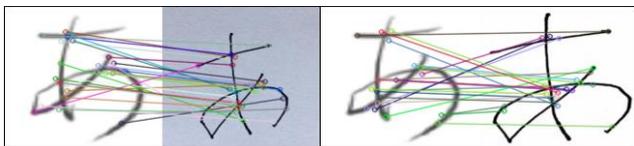


図 4 サンプルデータ A とユーザー1 のマッチング

3.2.3. 類似度の計算結果

サンプルデータ A とユーザー1 のスマートフォンで撮影したもの「あ」について類似度の計算結果を図 5 に示す。

```
In [2]: runfile('F:/ruijido/ruijido.py', wdir='F:/ruijido')
TARGET_FILE: 1cam (1).png
tr1.png 98.875
```

図 5 サンプルデータ A とユーザー1 のスマートフォンで撮影したものの類似度

3.2.4. 不一致部分の推定と表示の結果

サンプルデータ A とユーザー1 の「あ」の不一致部分の推定と表示の結果を図 6 に示す。

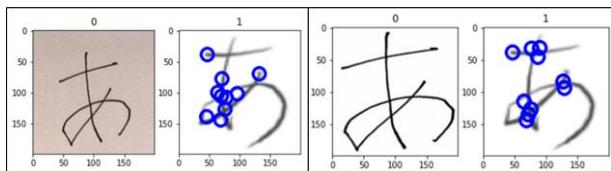


図 6 サンプルデータ A とユーザー1 の不一致部分の推定と表示

3.3. 性能評価

ユーザーが独自で選んだ 3 文字に対し撮影したものとスキャンしたものをそれぞれ以下の 5 項目に ◎・○・△・× の 4 段階で点数付けをすることで評価する。

- ①見本の文字とユーザーの文字を比べて不一致部分が正しく表示できていると思うか
- ②不一致部分に注意すれば手書き文字が上達できると感じるか
- ③実際に不一致部分に注意して文字を書いたとき手書き文字が上達できたと実感できたか
- ④類似度の点数は正しく表示されていると思うか
- ⑤総合的にこのシステムの評価

その結果を表 2 に示す。

表 2 性能評価の結果

撮影	◎	○	△	×
①	6	14	6	7
②	8	10	12	3
③	12	8	12	1
④	3	11	14	5
⑤	0	6	5	0
スキャン	◎	○	△	×
①	8	13	9	3
②	10	16	6	1
③	11	13	6	3
④	7	13	11	2
⑤	1	10	0	0

4. 考察

図 6 の結果より、スキャンしたものと撮影したもので同じ文字でも不一致部分の表示で違う箇所に円が描画された。これは、特徴検出の結果とマッチングの結果に違いがあったため、不一致部分についても違う箇所に円が描画されたと考えられる。表 2 の結果によると、撮影したものよりもスキャンしたものの方が精度は高いと言える。

5. まとめ

本稿では、AKAZE を使用して特徴検出・マッチング・類似度判定を行い不一致部分の推定をすることで手書き平仮名文字の上達支援システムを提案した。

今後は、カタカナや漢字についても上達支援できるようにデータを収集し、さらに自動で文字分割できる手法を試し、ユーザーの手間を削減する手法を検証していく。

参考文献

- 1) Sirabee リサーチ 自分は「字がうまい」と自信を持って言える?若い人に自称達筆率が目立つ傾向も (2019)
<https://sirabee.com/2019/04/30/20162064960/>
- 2) OpenCV3 で AKAZE 特徴量を検出する (2018)
<https://aicam.jp/tech/opencv3/akaze>