

オノマトペ発話による毛筆フォントのデザインと編集

Design and Editing Japanese Calligraphy Font using Onomatopoeia Utterance

村田健一^{*1} 中村剛士^{*1} 遠藤和也^{*2} 加納政芳^{*2} 山田晃嗣^{*3}
 Kenichi Murata Tsuyoshi Nakamura Kazuya Endo Masayoshi Kanoh Koji Yamada

^{*1}名古屋工業大学 Nagoya Institute of Technology
^{*2}中京大学 Chukyo University
^{*3}情報科学芸術大学院大学 Institute of Advanced Media Arts and Sciences

This paper focused on design support for Japanese Kanji-calligraphic “Shodo” fonts, especially design of scratched- and blurred- look. In order to support the design, we proposed a system using onomatopoeia utterance. The system can generate the scratched- or blurred- look suitable to the onomatopoeia the user speaks. Thus the user can design intuitively and conveniently a variety of scratched- and blurred- look of “Shodo” fonts. In order to realize the design support, the system utilized the correspondence between onomatopoeias and the scratched- and blurred- look. This paper reported the concept and the architecture of the proposed system. We demonstrated some output examples generated by the system.

1. はじめに

毛筆フォントとは、毛筆で書かれた文字を模したフォントであり、毛筆フォントのデザインに関連した研究はこれまでに多々存在する。中でも中村ら [1] は、掠れや滲みの表現を既存の毛筆フォントに付与するシステムを提案した。しかし、中村の手法を含め、こういった研究例の多くにおいては、毛筆フォントの外観デザインは、システムが提供する定量的な数値パラメータ設定に基づいて変化する。そのため、ユーザは、パラメータを熟知し、設定する必要がある。しかしながら、そのようなパラメータ設定は複雑であり、システム設計者以外が直感的に使用することが困難な場合が多い。

一方、日常的に用いる直感的なコミュニケーションツールとして、オノマトペがある。オノマトペは擬音語と擬態語の総称であり、物事の動作や状態を直観的かつ簡潔に表すことができるとされる。一般的に日本語のオノマトペの性質は、そのオノマトペを構成する文字の音印象で決定されるため、オノマトペは直観的な表現に適していると考えられる [2]。図 1 に、漫画で使用されるオノマトペの例を示す。オノマトペをグラフィックデザインに利用する研究は多々存在する。神原ら [3] は、描く線の特徴をオノマトペで指定できるオノマトペを提案した。飯場ら [4] は、オノマトペから連想される色彩を提示するシステムを提案した。このように、オノマトペをグラフィックデザインに利用することは効果的であると考えられる。また、フォントデザインにオノマトペを利用した研究報告はこれまで見受けられない。

そこで本研究は、フォントデザインにオノマトペを利用することを提案し、今回、漢字毛筆フォントのデザインを試みる。ユーザは、自身がデザインしたい掠れ・滲みを表現するのに相応しいと思われるオノマトペを入力し、フォントの掠れ・滲みのデザインをすることができる。書道の指導の際には、指導者が筆記時の腕の速度や力の加減をオノマトペで表現する例がある。これは、オノマトペから運筆動作が連想されることを期待したものである。このような指導と同様に、オノマトペを毛筆フォントデザインのシステムに利用することで、直観的な操作ができるシステムの実現を目指す。



佐藤秀峰, ブラックジャックによるしく, 漫画 onWeb, <http://mangaonweb.com>

図 1: 漫画で使用されるオノマトペの例

2. 提案システム

本研究で提案するシステムは、先行研究 [1] のシステムをベースに構築した。先行研究 [1] のシステムは、既存の毛筆フォントに掠れ・滲み表現を付加できる。図 2 に示すように、先行研究 [1] のシステムは、入力フォントの骨格上に複数の筆触カーソルと呼ばれる画像パッチを配置し、掠れ滲みを持った毛筆フォントを形成する。配置される 2 値カーソルは多値カーソルから生成され、2 値カーソルの黒点画素と白点画素の割合はシステム内の掠れパラメータによって決定される。また図 3 に示すように、先行研究 [1] のシステムは、2 値カーソルの黒点画素を中心とする円を描画し滲みを表現しており、この円の半径はシステム内の滲みパラメータによって決定される。そのため、ユーザは掠れパラメータと滲みパラメータを変更することにより、出力フォントの掠れと滲み度合いを調節できる。しかし、先行研究 [1] のシステムは、フォント全体に 1 組の掠れ・滲みパラメータしか割り当てられず、フォントを構成する部分毎に異なる掠れ・滲み度合いを設定できないため、フォントデザインの表現が制限される。また、ユーザは、掠れ・滲みパラメータと形成される掠れ・滲み表現との関係を事前に理解しておかなければならない。

そこで本研究では、先行研究 [1] のシステムに以下に示す 2 つの機能を追加する。まず、文字を構成するストロークの細かい部分毎に掠れ・滲みパラメータを設定できる機能を実装する。図 4(a) は、ある毛筆フォントの漢字の“一”である。本

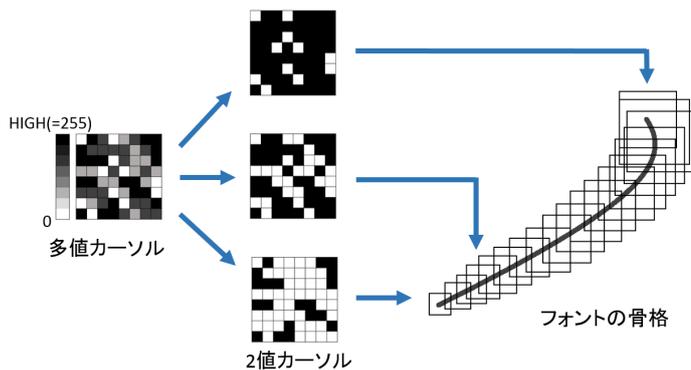


図 2: 先行研究 [1] の処理概要

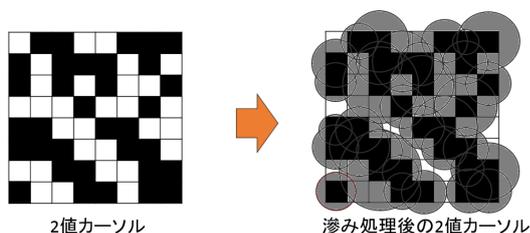


図 3: しみ表現

システムでは、フォントのグラフ構造化により、図 4(b) に示すように“一”を 3 つの部分に分け、それぞれの部分に異なる掠れ・しみパラメータを設定できる。もう 1 つの機能として、オノマトベ発話により、掠れ・しみパラメータの設定を行う機能を実装する。図 4(c) は、図 4(b) のように分けられた部分に左からそれぞれ、オノマトベ“ぐ”、“しゅ”、“ぐ”を割当てた例である。

図 5 に、本システムの処理の流れを示す。まず、ユーザはシステムに処理対象フォントを入力する。システムは入力されたフォントのグラフ構造を生成する。一方で、ユーザは処理対象フォントの指定された部分（グラフ構造のノードまたはエッジ）に与えたいオノマトベを入力する。オノマトベの入力方法は音声入力を採用し、音声認識ソフトウェア Julius[6] を用いて実装した。次に、システムは、オノマトベを掠れ・しみパラメータに変換し、得られた掠れ・しみパラメータをノードまたはエッジに割当てて、これを繰り返すことでフォント全体の掠れ・しみ処理を行う。

3. フォントのグラフ構造化

本システムでは、漢字を部位毎に分解し、掠れ・しみパラメータを割り当てる。漢字は一つ以上のストロークの組み合わせで成り立つ。また、それぞれのストロークは“止め”や“払い”といった部位に分類できる。ここでは、図 6 に示すように漢字の部位を定義し、分解する。

図 7 に、フォントのグラフ構造化の流れを示す。(1) 処理対象ストロークが入力される。(2) 細線化処理により処理対象ストロークの骨格を取得する。(3) 骨格を構成する画素群において、半径 R_{node} の円内に存在する骨格の端点・分岐点で点集合を作成し、集合内の端点・分岐点およびそれらをつなぐ画素群で 1 つのノードを形成する。また、あるノードが別のノ

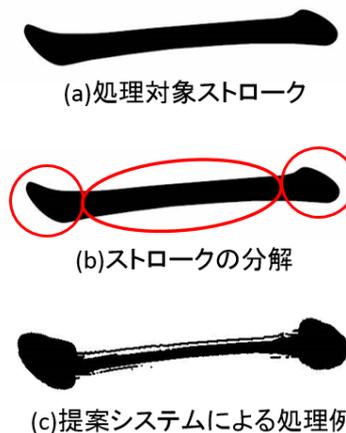


図 4: フォントの部分毎のデザイン

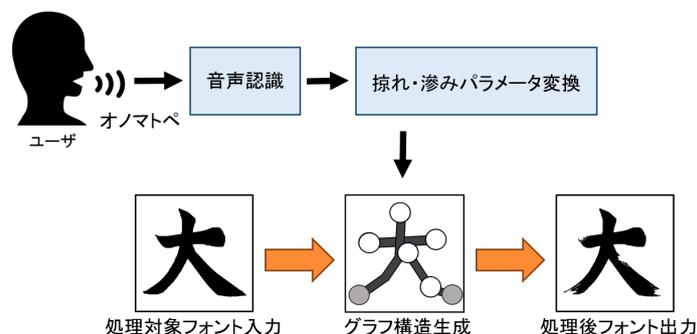


図 5: 提案システムの概要

ドとつながっている場合には、それら 2 ノード間の骨格構成画素群で 1 つのエッジを形成する。(4) ノードを N^+ 、 N^- の 2 つに分類する。ノード内の端点・分岐点が半径 R_0 以下の円内に存在する場合、このノードを N^- 、半径 R_0 より大きくかつ R_{node} 以下の円内に存在する場合、そのノードを N^+ に分類する。システムは N^+ のノード、エッジのそれぞれにユーザが入力したオノマトベを与え、掠れ・しみ処理を行う。

N^- のノードについては、オノマトベの割当てを行わないこととする。一般的に書道の指導において、図 6 における“跳ね”、“払い”のストロークの終端にはオノマトベを割当てない。さらに、“入り”のストロークの始端が細い場合にもオノマトベを割当てない。また事前実験の結果から、ノード N^- が“跳ね”、“払い”のストロークの終端、または“入り”のストロークの始端に相当することが分かった。そのため、 N^- のノードに対しては、オノマトベを割当てをしないものとした。

4. オノマトベと掠れ・しみパラメータの関係

図 8 に示すように、本システムではオノマトベから掠れ・しみパラメータへの変換を、オノマトベと筆圧・筆速の関係と、筆圧・筆速と掠れ・しみパラメータの関係を用いて行う。北條ら [5] はオノマトベと筆圧・筆速の関係についての調査を行った。一方、毛筆書道において、筆記時の筆圧・筆速は形成される毛筆文字の掠れ・しみに大きな影響を与える。また、先行研究 [1] のシステムにおける掠れ・しみパラメータはそれぞれ掠

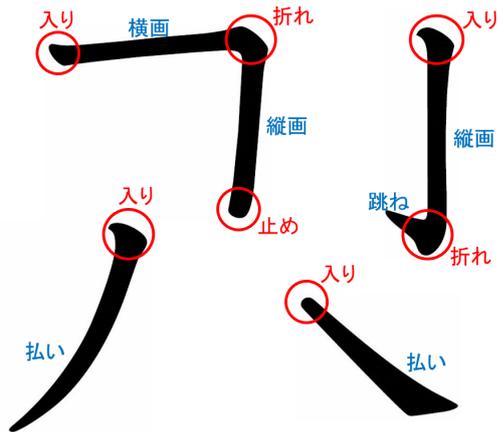


図 6: 漢字の部位の名称

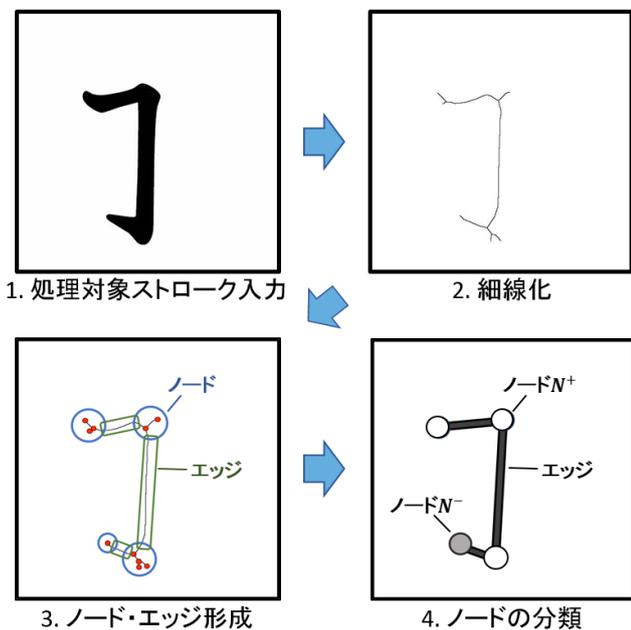


図 7: グラフ構造化の流れ

れ・滲みを表現するパラメータである。そこで、これら 2 つの関係を用いてオノマトペと掠れ・滲みパラメータの関係を導出し、本システム内で用いることとした。

オノマトペと筆圧・筆速の関係を調査するにあたって、本システムで用いるオノマトペの選定、およびオノマトペ毎の筆圧・筆速の計測を行った。オノマトペの選定に関しては、成人男女 13 名に対して毛筆書道で用いるオノマトペについてのアンケートを行った。表 1 に、アンケートの結果採用されたオノマトペを示す。その後、それぞれのオノマトペをイメージしながら筆記を行った際の筆圧・筆速を計測した。計測にはワコム製のペンタブレット Intuos4 を使い、10 ミリ秒ごとにサンプリングして計測を行った。本システムにおいては、サンプリングされた筆圧・筆速の平均を、そのオノマトペの筆圧・筆速として用いることとした。表 1 に、オノマトペ毎の筆圧・筆速を示す。筆圧の値はペンタブレットにより計測された値を [0,1] の範囲で正規化したものであり、1 に近いほど筆圧が高い。

表 1: オノマトペと筆圧・筆速の関係

ノード/ エッジ	部位	オノマトペ	筆圧	筆速 (pt/msec)	
ノード	入り	ぐ	0.983	0.0055	
		す	0.746	0.0023	
		とん	0.886	0.0541	
	折れ	かく	0.882	0.0003	
		く	0.894	0.0004	
		くい	0.890	0.0002	
		ぐい	0.916	0.0003	
		びた	0.707	0.0006	
	止め	ぐ	0.816	0.0089	
		びた	0.751	0.0076	
	エッジ	縦画	ぐー	0.989	0.3051
			さ	0.927	0.7450
さー			0.871	0.3912	
すー			0.885	0.3511	
ずー			0.982	0.3280	
ずず			0.995	0.4353	
横画		さー	0.694	0.6290	
		しゅー	0.828	0.6970	
		すー	0.807	0.5496	
		すいー	0.834	0.5660	
跳ね		さ	0.732	0.9203	
		しゃ	0.788	0.7782	
		しゅ	0.902	0.5731	
		す	0.755	0.8027	
		ぴ	0.872	0.6210	
		ぼん	0.884	0.7373	
払い		さ	0.716	1.3636	
		しゃ	0.704	1.6226	
		しゅ	0.692	1.3640	
		す	0.668	1.2086	
		ば	0.824	1.7656	
		ふわー	0.612	0.7537	

次に、筆圧・筆速と掠れ・滲みパラメータの関係を調査した。調査には、毛筆を装着でき、一定の筆圧・筆速で筆記が可能なロボットを用いた。ある筆圧・筆速の組と、ロボットがその筆圧・筆速で筆記を行った際のストロークと同程度の掠れ・滲みを持つフォントを出力する掠れ・滲みパラメータの組を結びつけた。

5. 実験・考察

図 9 にモリサワフォント G-OTF 筆順 ICA Std M56 の“永”とそのグラフ構造を示す。図 9 の左図の数字は筆順を表し、右図の白円、灰円、黒線はそれぞれノード N^+ 、 N^- 、エッジを表す。図 10 に、本システムによるフォントのデザイン例を示す。また、それぞれのデザイン例の下にノード N^+ ・エッジに与えられたオノマトペを示す。

図 10 に示すように、同じ入力フォントに対して異なるオノマトペを与えることで、異なる掠れ・滲みをもった毛筆フォントを得られることが分かる。また、1 つのストロークの中に異なる掠れや滲みの表現が見られ、先行研究 [1] に比べ細かい部

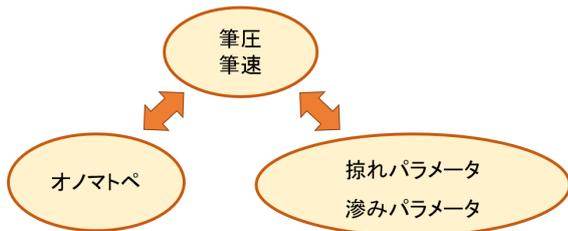


図 8: オノマトペと掠れ・滲みパラメータの関係

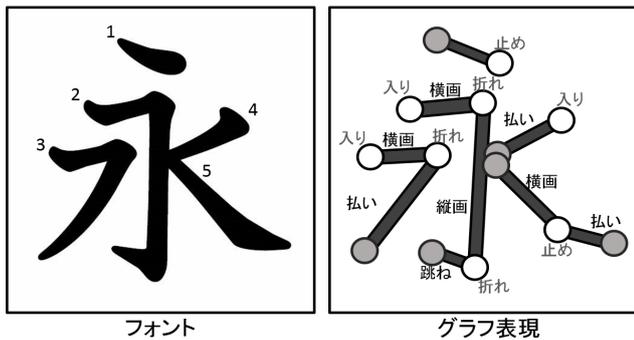


図 9: “永” のグラフ構造

分毎のデザインが可能となった。しかし本システムでは、処理対象フォントのグラフ構造化を細線化アルゴリズムを利用して自動的に行っている。そのため、フォントによっては、ユーザのイメージしたようなグラフ構造が得られず、望むようにオノマトペの割り当てができない、という問題がある。

6. おわりに

本研究では、ユーザがより直観的かつ柔軟な毛筆フォントの掠れ・滲みデザインできるシステムの開発を目的に、先行研究 [1] のシステムにオノマトペ音声入力機能、及び、グラフ構造化によるストローク毎の掠れ・滲みデザイン機能の実装を行った。今回のシステム実装では、筆記動作の筆圧・筆速と掠れ・滲みパラメータの関連付け、グラフ構造におけるノード・エッジへのオノマトペの割り当て方等の課題が残ったため、これらの点について、今後検討し解決する必要があると考えている。

参考文献

[1] Tsuyoshi Nakamura, Lifeng He and Hidenori Itoh: A proposal of generating artistic Japanese calligraphic fonts, *Advanced Topics in Artificial Intelligence*. Springer Berlin Heidelberg, pp.490-491, 1999.

[2] Teruhisa Komatsu and Hiroki Terashima: Moya-moya drawing: Proposing a drawing tool system which can utilize users' expressed onomatopoeias as a drawing effect, *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS)*, 2012 Joint 6th International Conference on, IEEE, pp.494-497, 2012.

[3] Keisuke Kambara and Koji Tsukada: Onomatopen: Painting using onomatopoeia, *Entertainment*



1画目: さー,びた	1画目: すー,ぐ
2画目: ぐ,さー,ぐい,すー,ぐい,しゅ	2画目: す,しゅー,かく,ずず,くい,ぴ
3画目: す,すー,ぐい,ば	3画目: とん,すいー,びた,ふわー
4画目: ぐ,しや	4画目: す,しゅ
5画目: しゅー,びた,ば	5画目: すー,びた,ふわー



1画目: しゅー,ぐ	1画目: すいー,びた
2画目: とん,すー,く,ずー,びた,ぼん	2画目: ぐ,すいー,びた,さく,さ
3画目: ぐ,さー,くい,しや	3画目: とん,しゅー,かく,す
4画目: とん,す	4画目: す,ふわー
5画目: すいー,ぐ,さ	5画目: さー,ぐ,しや

図 10: 本システムのデザイン例

Computing-ICEC 2010. Springer Berlin Heidelberg, pp.43-54, 2010.

[4] Saki Iiba, Ayumi Shiga and Makoto Sakamoto: A method to propose color associated with onomatopoeia based on sound symbolism, *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS)*, 2012 Joint 6th International Conference on, IEEE, pp.518-522, 2012.

[5] Hiroki Hojo et al, *Classification of writing-skill features using embodied expertise onomatopoeias*, *Fuzzy Systems (FUZZ-IEEE)*, 2014 IEEE International Conference on, IEEE, pp.994-999, 2014.

[6] Akinobu Lee and Tatsuya Kawahara: Recent development of open-source speech recognition engine Julius, *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp.131-137, 2009.