修士論文　2003年度（平成15年度）

骨格情報を用いて生成される印刷媒体のためのかな書体の開発

慶應義塾大学大学院　政策・メディア研究科

山本晃士ロバート

修士論文要旨　2003年度（平成15年度）

骨格情報を用いて生成される印刷媒体のためのかな書体の開発

近年、コンピュータのハードウェアとソフトウェア、両面の品質向上に伴い、コンピュータを用いた組版環境であるDTP（デスクトップ・パブリッシング）は日本の文字組版の現場に急速に浸透した。

印刷史をふりかえると、印刷技術の変化が、常に新しい文字表現を生んできたことがわかる。活版印刷、写真植字の時代を経てDTPが普及した現在、コンピュータならではの新しい文字表現が生まれる土壌は整ったといえる。

かつて、日本の筆文字は、なだらかな大小の繰り返しや、左右の斜線による連綿など、さまざまな表現能力を持っていた。しかし、活版印刷が導入されて以来、主に経済的な理由によって、印刷媒体の文字においては、そうした文字表現は失われた。

本研究では、日本の手書き文字が本来持っている表現の要素を、コンピュータを利用することで印刷媒体の文字にも導入できないかと考えた。具体的には、文字の骨格情報と、仮想的な筆の上下動の情報を利用することで、文字を組む（並べて文章にする）と同時に、動的にタイプフェイス（印字面）が生成される新しいかな書体を制作する。この方法により、前後関係に依存して形が変化し、文章中の位置によって固有のかたちを持つかな文字の表現が可能となる。

キーワード
書体　フォント　かな　日本語組版　DTP

慶應義塾大学大学院　政策・メディア研究科
山本晃士ロバート

Abstract of Master's Thesis Academic

Development of KANA typeface generated using stroke data for print media

Today, Improvement in quality of both computer software and hardware made DTP (desktop publishing) as computer-aided printing technology popular in printing industry in Japan.

Printing history shows that, changes in the printing technology give rise to a new typography. After the time of letterpress or phototypesetting, in these years DTP spreads throughout printing industry, it have the ability to create new typographic expression.

The handwriting character written by the Japanese brush has various expressions. For example, a repetition of gently-sloping size, the natural-connection by the slash on either side, etc. However, when letterpress printing was introduced in Japan, such character expression was lost in the print media for mainly the economical reason.

In this study, I aim to introduce the expression element which handwriting character originally has into the typeface of the print media by using computer. In particular, the new KANA style of handwriting by which a typeface generated dinamically at the same time it puts a character in order with the stroke data on a character and the up-and-down motion data of an virtual brush. By this method, a form changes depending on an order relation and expression of the kana character which has a peculiar form with the position in a text is attained.

Key Word

Typeface, Font, KANA handwriting, Japanese typography, DTP

Keio University Graduate School of Media and Governance

Kohji R. Yamamoto

目　次

0 用語の定義

文字や文字のかたちを扱う分野は定義が曖昧な言葉が多い。文中で用いられるいくつかの用語を、誤解の無いよう、ここで簡単な説明をしておく。

書体：

実際に文字を書いたり、印刷用の文字を作る際に用いられる、一貫した様式のこと。また、「フォント」とは、そうした一貫した様式で設計された、ひとそろえの文字セットを指す。現実的には、コンピュータ上で扱われるフォント名（「平成明朝」や「ヒラギノ明朝」など）は、そのまま書体の名前としても使われている。

字形と字体：

字形とは文字形態の特徴であり、文字を構成する字体の骨格のバランスとその様式。たとえば、上の十字部分と最下部の斜め線が分離している「さ」と、繋がっている「さ」は字体は同じだが字形が異なる。

また、字体とは、どこにどういう線が繋がって、どういう形をとっているか、という概念を指し、字形より抽象度が高い語である。たとえば「国」と「國」は字体が異なる。また、書体が違っていても、たとえば同じ「山」の字なら、字体は同じである。

厳密に定義しようとすると非常に紛らわしくなるので、本論文中ではこれ以上深く説明はしない。また「字形」ということばに関してはやや広義に解釈し、手書き文字の字形は毎回異なる、という使い方もしている。

タイプフェイス（typeface）：

原義的には印刷用の文字書体の、インクの付着面の具体的なかたちを指す言葉。近年はその意味も拡張されてきているが、本論文中では、原義に即した意味に使う。

文字を「組む」：

印刷や組版の現場において、文字を並べ、ひとつながりに読んだり見たりできる状態にレイアウトすることを「組む」と呼ぶ。日本語の場合、縦書きの状態にレイアウトすることを「縦組みする」、横書きの状態にレイアウトすることを「横組みする」と呼ぶ。

活字：

活字という言葉は一般名詞としては文字そのものや、印刷された文字、の意味で使われる機会も多いが、本論文中では活版印刷における活字の実体、手で持つことのできる、いわゆる原義的な活字のことのみを指している。

文字表現：

論文中では、「文字を並べてレイアウトする際、種々の工夫を用いて、体裁のコントロールをした結果として生まれる、文字による表現」程度の意味で使っている。一般的な言葉ではないし、慣用的な印刷用語というわけでもないが、他に適切な言葉が見あたらなかったので使用した。

1 研究の目的と意義

1.1 はじめに

本研究では、コンピュータを用いてはじめて可能となる表現を生む、印刷媒体のための日本語かな書体を制作する。

近年、コンピュータを用いた文字組版環境であるDTP（デスクトップ・パブリッシング）は、日本の文字組版の現場に急速に浸透した。

しかし、現状のDTPによる組版は、DTP以前の印刷技術である活版印刷や電算写植の組版を、高品質に、また正確に模すことを目標としたものにすぎない。DTPが出現した当初は、その品質への不満の声も高かったが、今日ではその不満も徐々に解消され、品質にも一定の評価が得られはじめている。コンピュータを用いたからこそ生まれる新しい文字表現への追求が、行われてもよい段階に来ているといえる。

印刷技術の歴史を振り返ると、新しい技術が、新しい文字表現を生んできたことがわかる。続いてそれを示す。

1.2 日本語の印刷技術の現状

1.2.1 日本における印刷技術の変遷

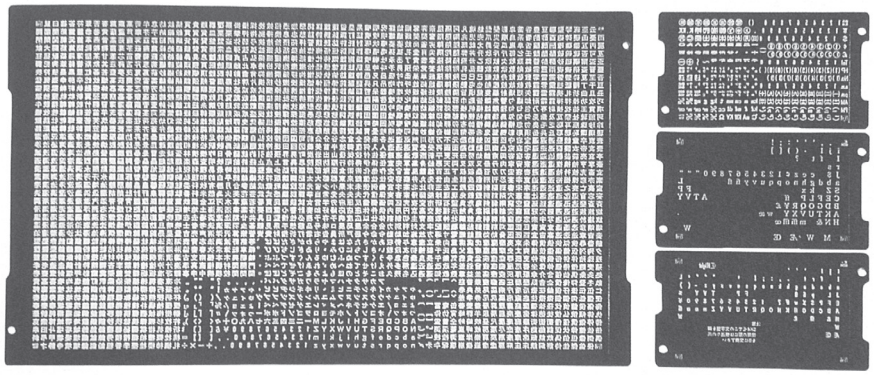
ここでは、今回の研究と直接関係してくる、活版印刷の技術が導入されて以降を取り扱うことにする。

西洋では、グーテンベルクが活版印刷の基礎を確立したといわれる1440年代以来、活版印刷はさまざまな分野で活用されることになったが、漢字を扱う文化圏では、実用上数千の活字を揃えなければならないという障害があり、木版活字の時代を経たのち、錫や鉛の活字が生まれるまで広く普及することはなかった。日本でも、江戸時代には既に一般市民の間にも印刷物が多く出回ることになるが、基本的にはいわゆる整版印刷（1ページ全体をひとつの版として彫るもの）であり、活字が使用されたことは稀であった。

日本における近代活版印刷の技術は、明治2年（1869年）、本木昌造が確立したとされる。ここで本木が行ったことは大きく2つある。ひとつは、活字の具体的な鋳造法や印刷方法を日本の産業事情に合わせて考案し、量産に耐えうる地盤を作ったこと。もうひとつは、鉛活字を鋳造する段階において、大小様々な活字が必要とされることから、検討の末、数

種類の活字のサイズを制定し、「号数活字」として体系化したことである。ここでは初号から8号までのサイズが規定され、さまざまな印刷物はその号数を基準として制作されることになる。

その後数十年を経て、いわゆる写真植字（以下写植）機が誕生することになる。写植機の発明は、1924年、後に写植メーカーの創業者となる石木茂吉と森沢信夫によるといわれている。写真術の応用によって、文字の原盤となる文字盤から所定の文字を選び出し、ネガフィルムとなっているガラス面[図1－1]を通して、暗箱内の印画紙に文字のかたちを焼き付けるものである。活字のように、刷るごとの摩耗・消耗が起こらないこと、印画紙に焼き付ける際にレンズを挟み、そのレンズを取り替えることで大きさを自由にコントロールできることは、画期的な出来事であった。



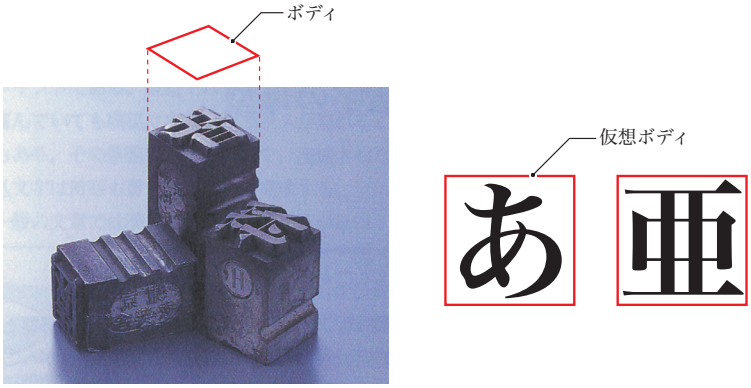
［1－1］写植のメイン文字盤とサブ文字盤。
硬質ガラスにネガフィルム状に記録された文字が並んでいる。

そしてさらに数十年後、DTPの技術が出現することになる。実質的なDTP環境のはじまりは、1985年、Apple社のコンピュータMacintoshと、Adobe Systems社が開発したページ記述言語PostScriptを扱えるプリンタLaserWriterの出現とされている。DTPとは、すなわち出版物のデザイン・レイアウトをパーソナルコンピュータ上で行ない、電子的なデータを印刷所に持ち込み、出版（publishing）することである。文字のかたちの実体はデジタルフォントとなり、物理的な存在が無くなった。また、かつて文字の扱いに関しては、活版工や写植のオペレータに指示を送るだけの立場だったデザイナや組版者の手元に常に文字の実体（つまり、デジタルフォント）があり、最終的な出力結果を見ながら同時に作業が進められるようになった。アルファベットに比べて遙かに数が多い日本語の文字の扱いや、日本語独自の組み方をどう再現するかという問題もあり、日本では欧米よりも普及はゆっくりとしていたが、現在は、長文を扱う文芸書等の分野をのぞいて、DTPは日本の組版の分野にも大きく普及している。

非常に簡略化した説明ではあるが、日本では明治以降、活版から写植へ、DTPへと、印刷媒体の文字を扱う環境が次々と変化していったのである。

1.2.2 技術が文字表現を生む

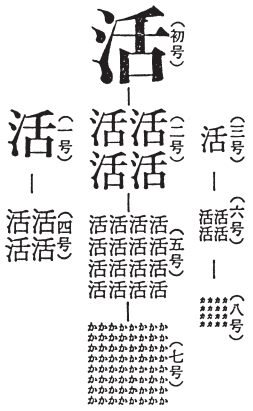
文字による表現、いわゆる組版やレイアウトの表現は、文字のかたちを記録する構造により決定されている部分が大きい。例を挙げると、活版印刷においては、活字の物体としての形が文字表現に制約を加えている［図1－2］。



［1－2］活字の持つボディの大きさと、デジタルフォントの仮想ボディ。写真は築地活版の活字と秀英社の活字。
本質的に、活字はボディの物理的制約から逃れられない。
また、デジタルフォントであっても、仮想的に設定した正方形のボディをもとにしてデザインされている。

活字による組版では、文字ひとつに対して、金属製の活字ひとつ分、という、物理的な大きさが存在する。文字のレイアウトをする（＝活字を並べる）うえで、この大きさを無視することはできない。活字を扱うときには、こんにちのワープロソフトを使えば簡単にできるような「同じ行に、複数の違った大きさの文字を並べる」ことすら、かなり困難な作業になる。これにより、活字を用いた組版では、自然と「活字をすべて同じ大きさにつくり、ぴったりと並べて組む」というルールができあがってくる。これは、印刷結果の読みやすさや美的な問題とは無関係な、文字の構造によるレイアウト表現の制約である。

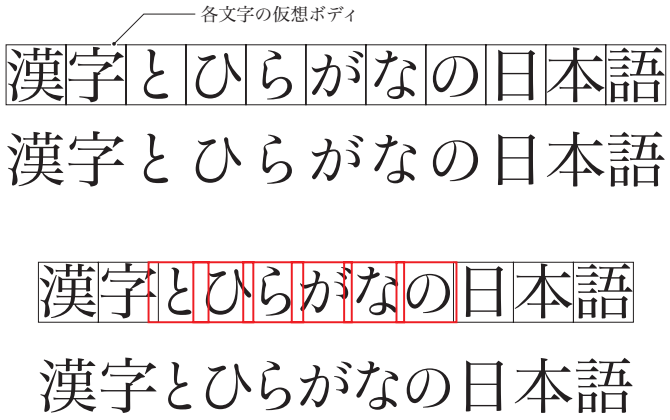
また、活字は物理的な大きさが存在する以上、文字の大きさも制限されている。たとえば1.2.1で触れた号数活字の場合は、当然ながら初号から8号までの9種類の中から選ぶことになり、仮に5号活字より大きい文字を印刷したいと思えば、次は4号を選ぶしかない［図1－3］。号数活字は非常によく計算された体系であり、その中で選んでいけば版面全



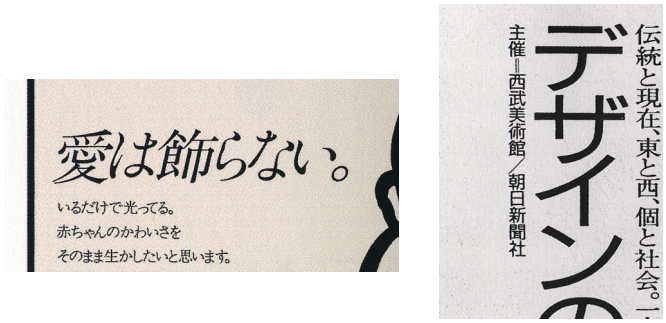
［1－3］号数活字のシステム図（縮小）。

体のレイアウトの破綻をきたしにくいという利点はあるものの、すくなくとも、全く自由に文字の大きさを選ぶことはできない。のちに号数活字とは別の文字サイズの体系も生まれてはいるが、文字のサイズを連続的に、自由に選べるようになったのは、写植が生まれて以降になる。

そして、写植の時代に生まれた文字表現として、自由な文字サイズとは別に、詰め組み[図1－4]と光学的な文字の変形[図1－5]がある。



［1－4］写植の詰め組み。
上段が詰めない組み方、いわゆる「ベタ組み」。下段が詰め組みの例である。
タイプフェイスが仮想ボディよりも食い込んでいるのがわかる。



［1－5］写植の光学変形の例。
左が斜体、右が平体。

印刷技術とその歴史を詳細にまとめた本、『印刷博物誌』の中に、次のような記述がある。

写真植字は光学式である点から、かまぼこ型の特殊レンズを光軸に挟み、そのレンズの角度を変えることにより、斜体や長体・平体など、活字では考えられないような変形文字をいとも簡単に印字できる。このことから文字によるデザイン分野に大きな影響をもたらした。

（中略）

写真植字機の利点は、文字盤を自由に交換することができるので、レンズの選択とあいまって多数の書体を印字することができることである。

さらに写研の石井裕子が古文の筆によるひらがなの送りにヒントを得て考案されたものに詰め文字方式がある。

『印刷博物誌』P.880より

活字にある物理的な大きさという問題が、原字をネガフィルムで持つ写植の技術により解決されたために、仮想ボディ [図1－2] より「詰めて」レイアウトする手法が一般化したのである。また、印画紙に焼き付ける際に用いられるレンズを、一定の歪みのある特殊なレンズに取り替えることで、原字からさまざまな変形が施された文字も利用できるようになった。詰め組みや斜体・平体・長体は、DTP 環境でもシミュレーションされ、一般的に使われている。

このように、文字による表現は、必然的に印刷技術や文字（タイプフェイス）を保持する構造やシステムの影響を必ず受ける。印刷技術そのものが、文字表現を制約し、規定しているのである。

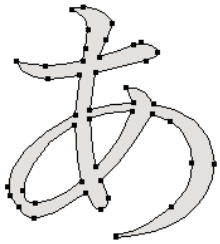
1.3 研究の動機

活版印刷から写植へと移行したときに、文字サイズの無段階化、詰め組みや文字の光学的変形が実現されたように、写植から DTP への移行とともに新しい文字表現、新しい組版の処理が生まれてもよいはずだ。DTP が誕生した当初、日本語のデジタルフォントも、組版ソフトウェアも、その品質を疑問視する声が少なくなかった。しかし、現在ではその問題も徐々に解決され、「写植の方が高品質であった」という声は小さくなりつつある。写植の組版を正確な再現した組版が、DTP 環境でも実現されつつあるいま、コンピュータを用いたからこそ生まれる文字表現を模索する段階だと考えた。

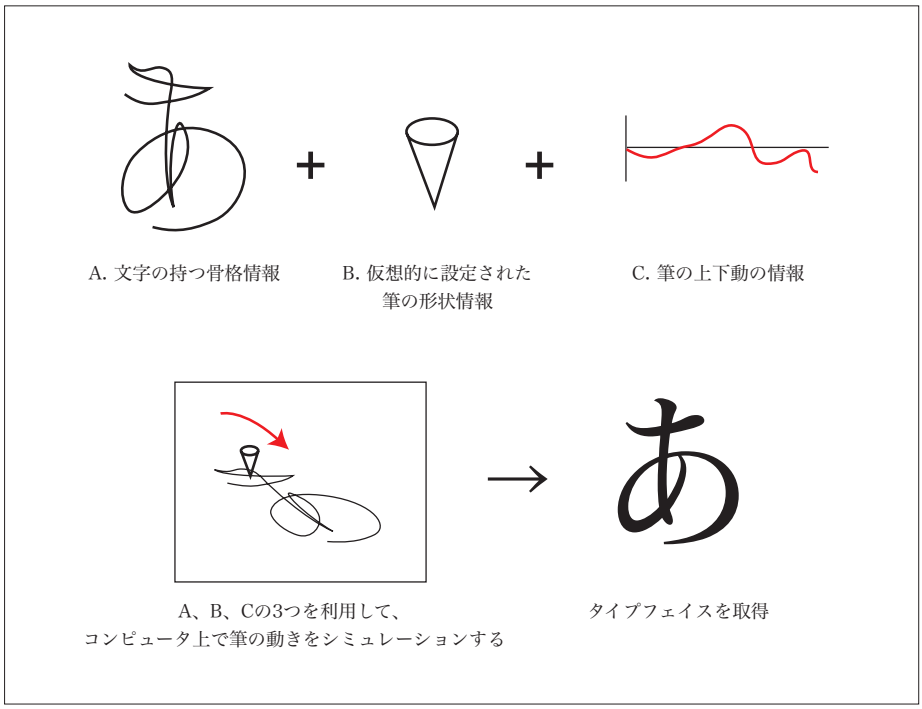
2 研究の概要

2.1 研究方法

本研究では、使用される文中の位置によって、字形が変化するかな書体を制作する。現在コンピュータ上で扱われている印刷用フォントのように、タイプフェイスのアウトラインを保持する方法［図2－1］ではなく、それぞれの文字に対し、文字の骨格データ、仮想的な筆の上下動のデータを制作し、文字を組む際にそれらを組み合わせることでタイプフェイスを生成する［図2－2］。



［2－1］アウトラインフォントは、タイプフェイスの外側の曲線を保持し、
中を「塗りつぶす」ことで文字のかたちを得るフォント形式である。



［2－2］骨格情報を利用してタイプフェイスを得る方法

1. 各文字の骨格を配置する　　2. 前の文字の終点と、後ろの文字の始点を接続する　　3. 独自のアルゴリズムにより、平滑化の処理を行う

［2－3］骨格の平滑化

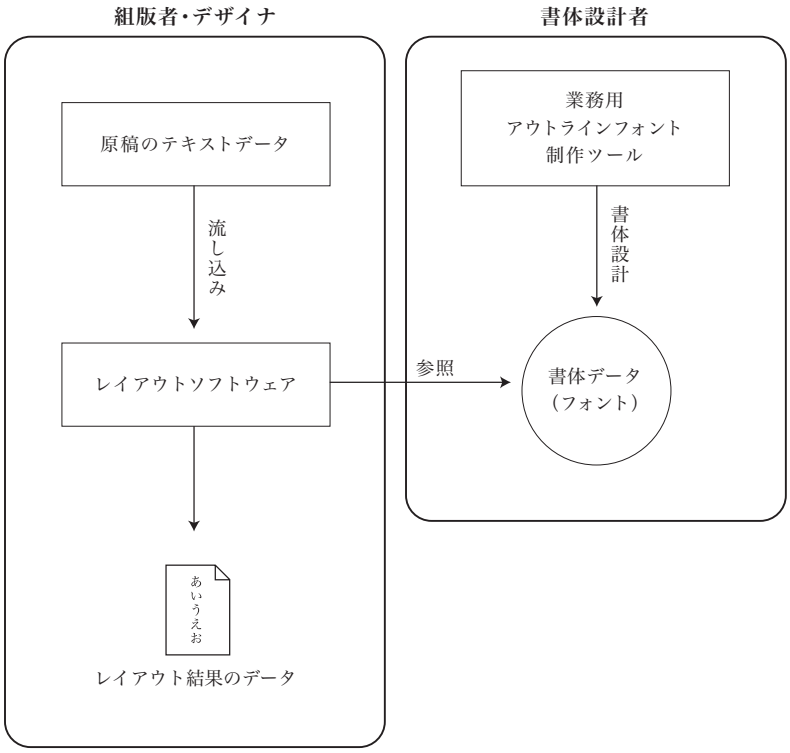
文章を組む際に、文字ごとの骨格データを接続し、平滑化等の処理を行う［図2－3］ことで、一行に使われている文字全体を通る一本の骨格データを作る。同様に筆の上下動データも接続し、平滑化したうえで一続きのデータを作る。こうして作られた、行全体を通る骨格データと筆の上下動データを組み合わせ、仮想的な紙面の上の筆を動きをシミュレーションすることで、前後の文字の影響により字形が変化したタイプフェイスを生成することができる。前後の文字の種類により、文字が持つ字形が変わるのである。今回は、こうして生成されたかな文字を、既存の明朝体の漢字と組み合わせて、さまざまな印刷媒体で使用することを想定している。

また、現存するDTPのシステムでは、本研究で作ろうとしている動的に変化する字形を持つ書体を扱うことができない。コンピュータ上で扱える（OSレベルでサポートされている）フォントのフォーマットは限られており、基本的にはアウトライン型のフォントの利用が前提となっている［図2－4］。そこで、実際にDTPの作業手順の中へ組み込むため、「書体データを制作するソフトウェア」「それを利用して作った書体データ」「書体データを参照し、字形に変化を加えるソフトウェア」の3つを制作する。生成されたタイプフェイスを、現在のDTPシステムの事実上の標準フォーマットであるPDF形式で出力することで、実際に印刷の現場で使われているレイアウトソフトウェア上で扱えるようにする［図2－5］。

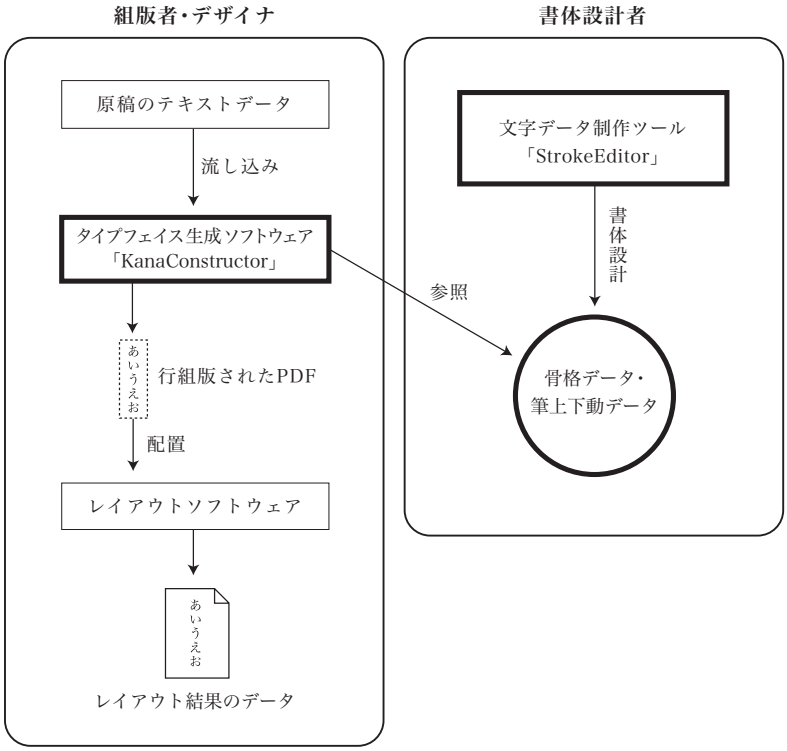
［付記：かな文字のみを制作対象とすることについて］

今回の研究では、筆文字や連綿文字を作りたいわけではなく、あくまで印刷媒体の、一般的な文章を組むための文字を制作したい。筆文字や連綿文字、あるいは楷書体は、未だに印刷の現場においては本文用の書体としては定着していない。日本語の印刷媒体の本文組みを想定するなら、まず明朝体の発展型を探るのが適切と思われた。明朝体のかな文字のみ別の書風のものに差し替えるという手法は、すでにひろく行われている。かな文字のみの変化でも、紙面が与える印象を様々に変化させることができる［図2－6］。

また、漢字部分を読み慣れた明朝体とすることで、かな文字を変形させた際に、文章が可読性が下がる危険が減らせるのではないかと考えた。



［2－4］現在の DTP における作業手順を単純化して示している。



［2－5］今回の研究が組み込まれる場合。
書体設計者側・組版者側両方の領域に踏み込む必要があった。
太線で囲われている部分が、本研究で制作した部分になる。

プルートォ――というのがその猫の名であった――は私の気に入りであり、遊び仲間であった。食物をやるのはいつも私だけだったし、彼は家じゅう私の行くところへ

プルートォ――というのがその猫の名であった――は私の気に入りであり、遊び仲間であった。食物をやるのはいつも私だけだったし、彼は家じゅう私の行くところへ

プルートォ――というのがその猫の名であった――は私の気に入りであり、遊び仲間であった。食物をやるのはいつも私だけだったし、彼は家じゅう私の行くところへ

［2－6］3種とも、漢字は同じでかな部分だけ書体を変えてある。

2.2 研究のねらい

2.1で示した方法により、文字が本来持っている「固有性」を再現したかな文字が再現できる。DTP 以前の印刷技術では、文字に「固有性」を持たせることは、主に経済的な理由により実現できなかった。コンピュータを用いることでそうした経済的な問題を解決し、印刷媒体に用いられる文字にも持たせることができると考えた。

ここで唐突に「固有性」ということばを使ったが、論点を明確にするため、本論文の中では、この「固有性」ということばを、ある特定の意味に使いたい。続いてそれを示す。また、印刷術の成立を追っていくと、印刷媒体の文字にこの「固有性」を持たせるという発想は、ごく自然なものであるといえる。それについても同時に触れていきたい。

2.3 文字の持つ「固有性」について

2.3.1 文字の固有性とは

人により、地域により、時代により、文字はその形に様々な差異をもつ。ひとつの文章の中でさえ、実際に文章を手で紙に書くところを想像するとわかるように、同じ文字でも文中の位置によって、違った形を持つのが普通だろう。

はじめまして、わたしは山本と申します。

などという文章でも、こうしてコンピュータを用いて印刷すれば、当然同じ文字は完全に同じ字形を持つ。しかし、同じ文章を手で書けば、「はじめまして」の「は」と「わたしは」の「は」は、どうしても違う形になる。また、内容のまったく違う文章であれば、同じ人であっても、字のかたちは違ったものになるだろう。元来、文字とは、それが形になる段階で「ひとつとして同じものはできない」ものだ。これが文字の持つ固有性である。

現在われわれが目にする印刷媒体の文字には、この固有性というものはほとんど失われている。しかし、印刷術が生まれた当初は、こうした「書く人間による」固有性を保持したまま大量複製しようという意図があった。

2.3.2 活版印刷における文字の汎用化

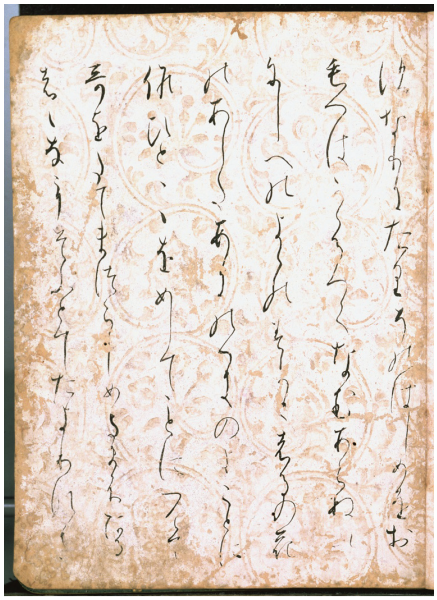
グーテンベルクによりその基礎が作られたとされる活版印刷術は、もともと手書きの写本を機械的に再現し、大量複製することが目的であった。活版印刷で作られた最初の書物といわれる『42行聖書』も、写本の「手書きらしさ」を出すために、同じ文字でも違う字形の活字を複数用意し、ひとつの印刷物の中でも、状況に応じて組み替えて使っていた[図2–7]。今から考えれば特殊なことをしているように捉えがちだが、写本の再現という目的を考慮すれば自然なことだろう。

［2–7］42行聖書の活字。
同じ文字でも、かたちが一定していない。

その後、印刷術の発展とともに、文字は汎用化の道を辿る。大きな理由は単純に経済性の問題である。1セットの書体の中で、同じ字に対して異なる字形を複数用意することは作業的・コスト的負担が大きい。そのため、文字のかたちを統一し、汎用性を持たせることで、文中のどこに来ても違和感がないように調整されることになる。「a」の字は文中のどこでも同じかたちになり、「e」の字は文中のどこでも同じかたちになる。現在では当たり前のことのように感じるが、写本に見慣れていた当時の人々にとっては、同じ字が、文中のどこでもまったく同じかたちをしている印刷物は、かなり奇異なものに映ったに違いない。

2.3.2 日本の印刷の場合

日本の場合はどうだろうか。かつて、日本の筆で書いた文字では、なだらかな大小の繰り返しや、左右の斜線による連綿など、文章に作者の感情が織り込まれた文字表現が行われていた[図2–8]。しかし、活版印刷の技術が導入された際、やはり印刷媒体においてはそうした文字表現は失われてしまうことになる。



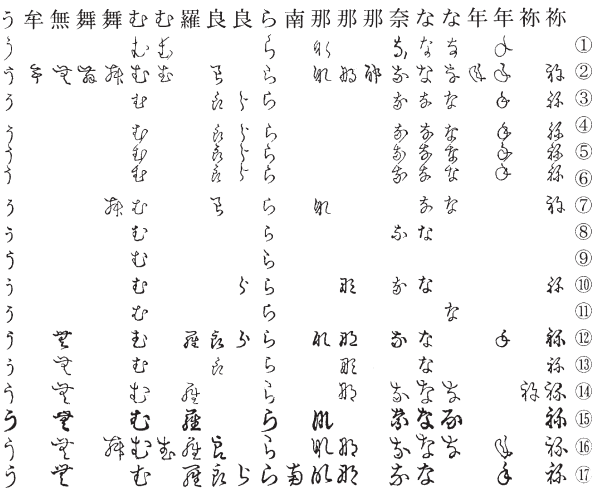
［2－8］古今和歌集・元永本上巻より。
筆の動きによりさまざまな表情がある。

日本語の印刷術においても、活版印刷が導入される以前の木版の時代は、まず人が紙に書きつけた絵や文章を原稿として、刷られる面（ページ）全体を木に彫り付けて版を作っていた［図2－9］。いわゆる整版印刷である。この時点では、印刷物でありながら、人が書いた文章をそのまま印刷に載せているので、文字の固有性は保持されていたといえる。

［2－9］絵入り「伊勢物語」と豊国画読本の版木。

続いて、活版印刷術が導入されることになるが、大小さまざまな形、大きさの字を持つ日本語は、そのままでは活版のシステムにそぐわない。そこで、すべての文字をまったく同じ大きさの正方形におさめることになった。ここで初めて、「すべての文字がすべて同じ大きさ」という現存の印刷物の考え方が生まれたのである。この時点ですでに、印刷媒体においては、筆文字の持つ表現力は大幅に失われてしまったといえる。

その後の初期の日本の活版印刷では、同じ字でも印刷所によって、あるいは活字の種類によって違う字形を持っていた事例もある。分かりやすい例として、変体かな活字［図2－10］の存在を挙げたい。



［2－10］実際に使用されていた活字を、号数別・年代別にまとめたもの。
同じ字でも、形が一定していないのがわかる。

当時、かな活字は、さまざまな人が書いた、書風の癖のある字をそのまま活字へと転用したものが主流であった。変体かな活字もそうした中で生まれたものである。大きさも字形も、現在と比べればかなりばらつきがある（そもそも、印刷術が普及するまでは、書き文字の文字の大きさをそろえる、という意識そのものが希薄だったと思われる）。

しかし、活版印刷が普及するにつれ、こうした変体かな活字は姿を消し、字形も、それぞれの文字の大きさのバランスも、次第にひとつに収斂されていった。アルファベット語圏の場合と同じように、文字の汎用化が行われたのである。

筆者は（そしておそらく戦後の義務教育を受けた大半の日本人も）、小学校で文字の書き方を習うとき、正方形の中におさめられた「お手本」を参考にして練習をさせられた。ここで教えられた文字の形や、正方形の中におさめるように書く、という方法論はあきらかに、活版印刷が導入されて汎用化が行われた活字の影響によるものだ。印刷術の世界で変容した字形が、人々の間に普及し、慣れ、手書きの文字に「逆輸入」されたのである。

現代の印刷物においては、われわれは同じ文字がすべて同じ字形であることに違和感を感じなくなったが、これは単純に多くの印刷物を目にして、大きさを合わせた文字を書くことが普及し、同じ字がまったく同じ大きさ、まったく同じ字形を持つことに「慣れた」からに過ぎない。

2.3.3 文字に固有性を持たせた最近の事例——OpenType フォント

DTP 全盛となっている近年、これまで説明してきた文字の「固有性」に着目したと思われる技術が発表されてきている。ここでは、Adobe Systems 社と Microsoft 社が共同開発したフォントファイル形式の一つである、OpenType フォントの独自機能を例として挙げる。OpenType フォントは、商業印刷で扱われる多くの機能を有しているが、「固有性」に着目したといえる機能を抜き出して、ここで紹介したい。

◎前後関係に依存する文字

3.1.1で触れるが、前後の文字や文章中の位置に依存して、特別に用意された字形に置き換えるという手法は、活版印刷の時代からあった。合字とは、組まれた文中に特定の組み合わせの文字が現れた場合、特別に作られたひとつの文字に置き換えることで、視覚的に美しく見せる効果を狙うものである。図2－11では、「fi」「fj」などを合字に置き換える例が示されている。スワッシュとは、アルファベットに「ひげ」状の装飾をつけたものであり、基本的に、段落頭で用いられるために特別に作られた文字である。活字からデジタルフォントへと変化した現代においても、こうした合字やスワッシュは生き残っている。

without ligatures	This office fjord halfb
with ligatures	This office fjord halfb

without swash	*Aidan Sue Veronica*
with swash	*Aidan Sue Veronica*

［2－11］OpenTypeフォントの合字機能、スワッシュ機能。

◎横組み用仮名・縦組み用仮名

同じフォントの中で、縦組み用、横組み用、それぞれの特性にあわせた仮名文字を有する例。横組み用のかなは、全体としてふところが広く、字のシルエットが正方形に近づいている。

あいうえお
あいうえお

［2－12］OpenTypeフォント「ヒラギノ明朝Pro」の縦組み仮名（上）と横組み仮名（下）。

◎漢字の異体字サポートの機能

異体字を持つ漢字について、さまざまな異体字を呼び出し、置き換えて使用することができる［図2－13］。

こうした異体字は、いままでコンピュータ上で扱うことができずに、組版者や印刷会社が個別に、外字と呼ばれる専用の字を作って対応していた。異体字サポートの機能とは、こうした異体字を統一したルールで扱えるようにしたものである。

現在普及している日本語の文字コード（JISコード等）では、このような異体字に統一したコードが割り振られていない。日本語の文字コードが制定される際、常用的に使われる漢字に優先してコードを割り振っていった結果である。漢字コードの規格は制定機関

［2－13］OpenTypeフォントの機能で呼び出した「剣」の異体字。

や年数によって何種類もあるが、例としてJIS X 0208－1990の規格を挙げると、文字セットとして定義されているのは6,879文字である。それに対して、現在発表されているOpenTypeフォントの一部は、最大17,000文字以上を収録している。

「似ているから」あるいは「もともと同じ意味の漢字だから」という理由で、コストのためにひとつに集約されていた漢字について、再度正確なばらばらな字体を再現しようとしているのである。この状況も、技術的な制約によって汎用化されていた文字に対して、（個人ではなく、地域や時代によって生まれた）固有性を与えなおそうとする動きといえるだろう。

このように、文字の固有性に注目した実験は、徐々に行われはじめている。本研究も、「コンピュータの力を借りて文字に固有性を持たせる」という意味において、このような試みの一環と位置づけられる。

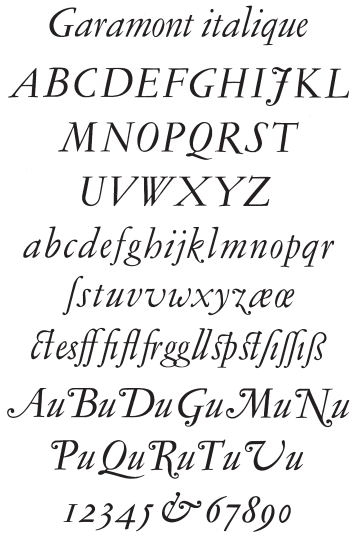
3 先行事例

ここでは特に、「印刷媒体の文字に固有性を持たせる試み」と、さまざまな形で「文字を生成する技術」について、先行事例を取り上げる。

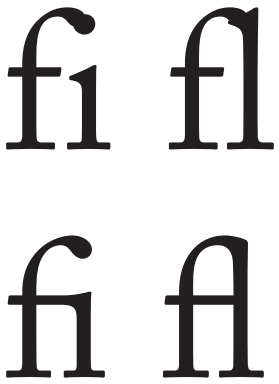
3.1 文字に「固有性」を持たせたもの

3.1.1 合字とスワッシュ・レター

合字もスワッシュ・レターも、活版印刷の出現とほぼ同時期に作られた考え方である。文字を組む際、文中の場所や文字の組み合わせによって、美的な効果を狙い、単体で作った活字とは別の、専用の活字に置き換えるものである［図3－1、3－2］。



［3－1］パリのドベルニ・アンド・メイニョ活字鋳造所「Garamond Italic」書体見本。1600年代に作られた活字の見本だが、ct, es, ff, fi などのリガチャ、さらに大文字のスワッシュ・レターがある。

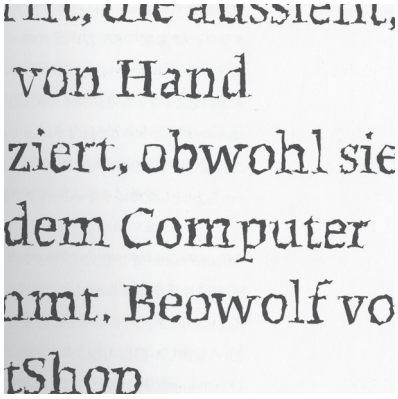


［3－2］現在販売されているフォント「Adobe Caslon」に収録されている合字の例。上段が通常の文字を個別に並べた状態。下段が置き換えられる合字。

また、かつて日本の活版印刷でも、ひらがなの組み合わせなどによる合字はあった（例としては、1591年にヤソ会がインドのゴアから持ち込んだ印刷機で制作された和文字等）が、現在はほとんど使われていない。おそらく、文字数が多い日本語には、組み合わせのパターンが増えすぎてそぐわないこと、そして縦組みと横組み両方に対応しなければならないという、日本語固有の事情のためだと考えられる。合字を作ってしまうと、縦組み用か横組み用、どちらか専用の字形になってしまい、制作や管理の負担が増大してしまうのである。

3.1.2 Beowolf

1990年、Just van Rossum、Erik van Blokland により制作されたコンピュータ上で扱うための書体。文字が再生されるたびにある種のランダム性をもって形が変わり、二度と同じ形が現れないという性質を持つ [図3－3]。ランダマイズという手法によって、文字に固有性を持たせた例といえるだろう。



［3－3］Beowolf の組見本。同じアルファベットでも、
文中に出てくるたびに違うかたちに変化しているのがわかる。

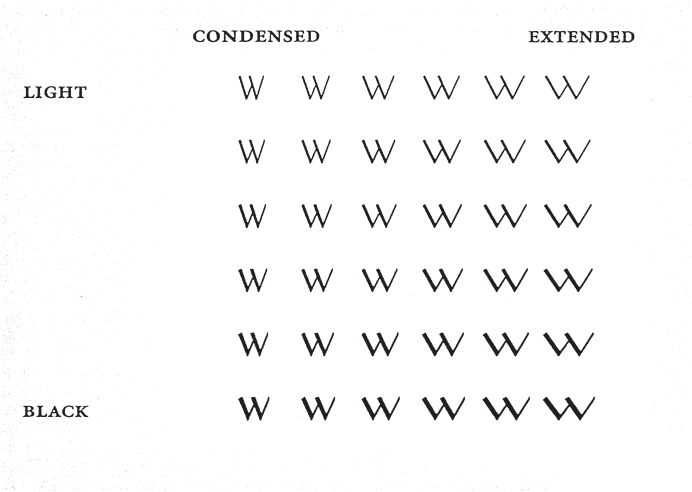
3.1.3 OpenType の独自機能

前章で詳しく触れたのでここでは割愛する。2.3.3を参照のこと。

3.2 文字を生成する技術

3.2.1 マルチプルマスターフォント

Adobe Systems 社開発によるフォントフォーマット。プライマリフォント(元になるフォント)のウェイト(線の太さ)や字幅を変更した新しいフォント(マルチプルマスター・インスタンス)をユーザが自由に生成できる。



［3－4］マルチプルマスター・インスタンスの表。
文字幅や文字の太さ等を自由に変えられる。

文字ひとつひとつに固有のかたちを持たせるわけではなく、使用する人間が、用途に合わせたウェイトや文字幅をもつ「文字セット」を作ることができるものである。革新的な技術と評価されたが、出力が安定しない等の理由で大きく普及するには至っていない。

3.2.2 METAFONT

Donald E. Knuth により開発された、コンピュータ上で書体を設計するための、一種の統合環境。その実体はコマンドラインベースのプログラムである。おのおのの文字がどういったかたちを持つのかが記述されたソースファイルを実行することで実際のタイプフェイスを描画し、gf 形式のフォントファイルを書き出す。

きわめて多くの機能を持っているが、そのひとつとして、ある数値で表現された骨格に対し、仮想的なペンを走らせて形を描画するという機能があり、骨格をもとにしたタイプフェイスを生成することが可能である。ただし、前後関係によって字形が変化するような機能はない。

内部的に文字のアウトラインを保持しておき、必要に応じて内部を塗りつぶしたビットマップデータを書き出す方式をとる。アウトラインフォントであるとも、ビットマップフォントであるともいえないが、一般的なアウトラインフォントにおいて、出力機が担当するラスタライズ機能を「内包した」アウトラインフォントの一種であると考えることができるだろう。

3.2.3 本阿弥光悦「風神」プロジェクト

財団法人デジタルコンテンツ協会が実施している「先導的コンテンツ市場環境整備事業」の「本阿弥光悦流連綿書体の開発とその Internet 上のネットワーク連歌環境の構築」プロジェクトにおいて、和歌・俳句専用メーラとして「JiJiBaBaPet 光悦（仮）」のβ版を完成させた。また、後にクロス・プラットフォーム環境への対応を目指し Web 上で利用できる Java 対応の「風神プロジェクト」へと受け継がれることになったとされているが、現在は開発は休止し、未完成である。

ユーザが紙や書体を選択し、自分の作った和歌や俳句をテキスト入力すると連綿書体にして表示し、それをメールとして送付できるシステム。印刷媒体への適応を目指したものではないが、日本語環境において、前後関係によって字形が変化する文字を作ろうとした研究として注目できる。

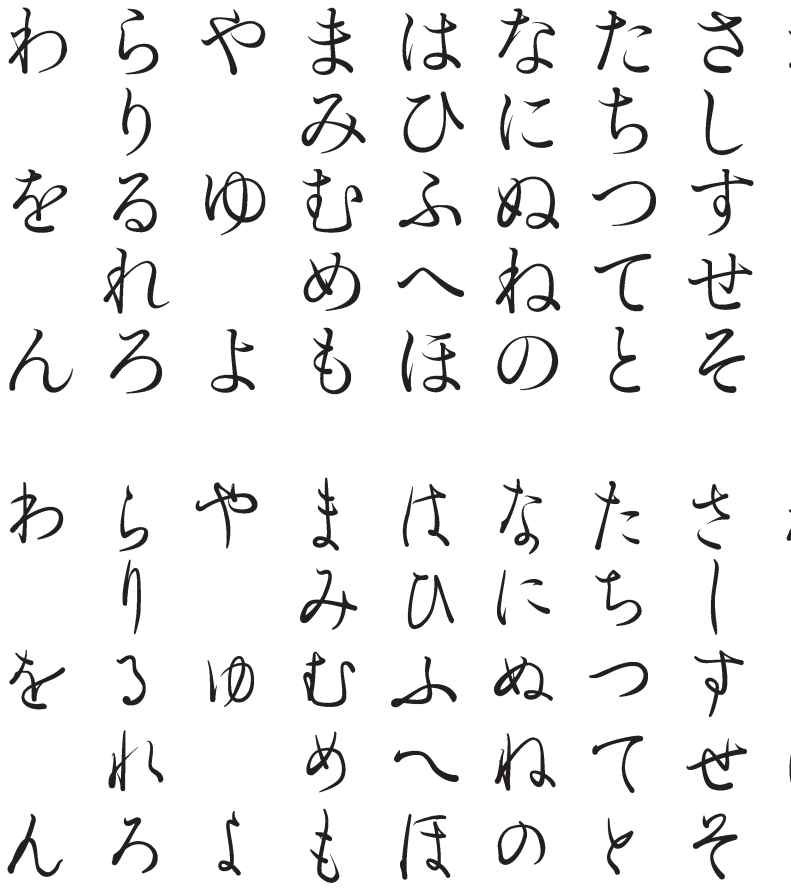
4 結果：制作したかな書体

実際にソフトウェアを試作し、かな文字の生成を行った。まずは、生成されたかな文字を、実際に組んだものを紹介していく。

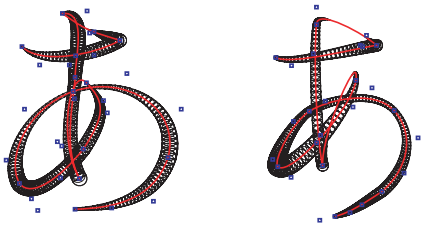
4.1 書体見本

4.1.1 書体データの種類

2章で示したとおり、今回制作したのは「書体データを制作するソフトウェア」「それを利用して作った書体データ」「書体データを参照し、字形に変化を加えるソフトウェア」の3点だが、書体データに関しては、書風の違いによるソフトウェアの親和性や、組んだ結果の違いを見るため、書風の傾向が違うものを2種類制作した[図4−1、4−2]。



［4−1］「築地かな（上）」「さがのかな（下）」の見本。
何の変形も加えていない状態。



［4－2］それぞれの文字の骨格とコントロールポイントを表示させた状態。
赤い線が骨格データ、青い四角が骨格を定義するためのコントロールポイント。双方とも、形は違っても、
骨格データと筆上下動データにより生成されていることには変わりがない。

「築地かな」は、現在一般的に普及している出版向けデジタルフォントの、ひとつの原型になったと思われる、築地活版印刷所の活字の骨格を参考にして作ったかな書体である。現在流通している印刷物にも、この活字の書風を参考にした書体は数多くあるので、多くの人にとって、オーソドックスである、と感じるだろう。現在普及しているデジタルフォントと近い書風を持つ文字を、今回制作したソフトウェアで扱ったとき、どう見えるのかを確認できる。

「さがのかな」は、書体設計家である今田欣一の制作した和字書体「さがの」と、「さがの」を作る際に原型にしたという、嵯峨本『伊勢物語』の中の文字を参考にして制作した。これにより、草書の崩しや連綿を強く残す文字を扱ったときどう見えるのかをたしかめたい。

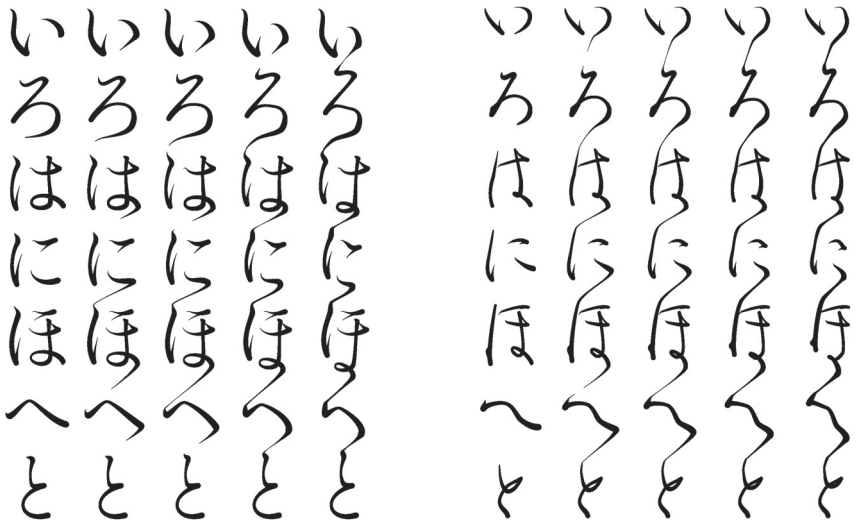
以下この2つの書体に対して、「築地かな」「さがのかな」という名称を使う。

4.1.2 前後の文字による影響

まず、実際に前後の文字により字のかたちが変化しているところを見せる［図4－3］。

［4－3］左2行が「築地かな」、右2行が「さがのかな」の組見本。
赤枠が何も変形を加えない状態の字形で、黒で示したものが、実際に前後の影響により変形させた状態である。
字形が前後の文字に応じて変形しているのがわかる。

また、今回文字を組むために「KanaConstructor」というアプリケーションを制作したが（機能の解説は5章参照）、アプリケーションに入力するパラメータを変えることで、変形の度合いを連続的に変化させることができる。「骨格移動」「骨格回転」「筆運び接続」という3種のパラメータを、それぞれ0から100までの間で設定できる。この3つの組み合わせにより、字形の変化の度合いを調整することができる[図4－4]。



［4－4］前後の文字の影響力を段階的に変化させた例。
影響力を上げれば上げるほど、文字に強い変形がかかることになる。

書体を実際に使用する段階で、意図したい効果に応じてパラメータを調整することができる。なお、字形の変化をわかりやすくするため、「KanaConstructor」の「文字サイズをランダマイズ」機能は切っている。

4.2 長文の組見本

次に、実際に、小説の文章を組んでみる。パラメータ等は任意に設定した。

すべて同じ文章を使用し、比較のため既存の一般的なフォントで組んだもの［図4－5］も掲載した。図4－6、図4－7が、それぞれ今回制作した「築地かな」「さがのかな」による組見本である。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにそこの主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

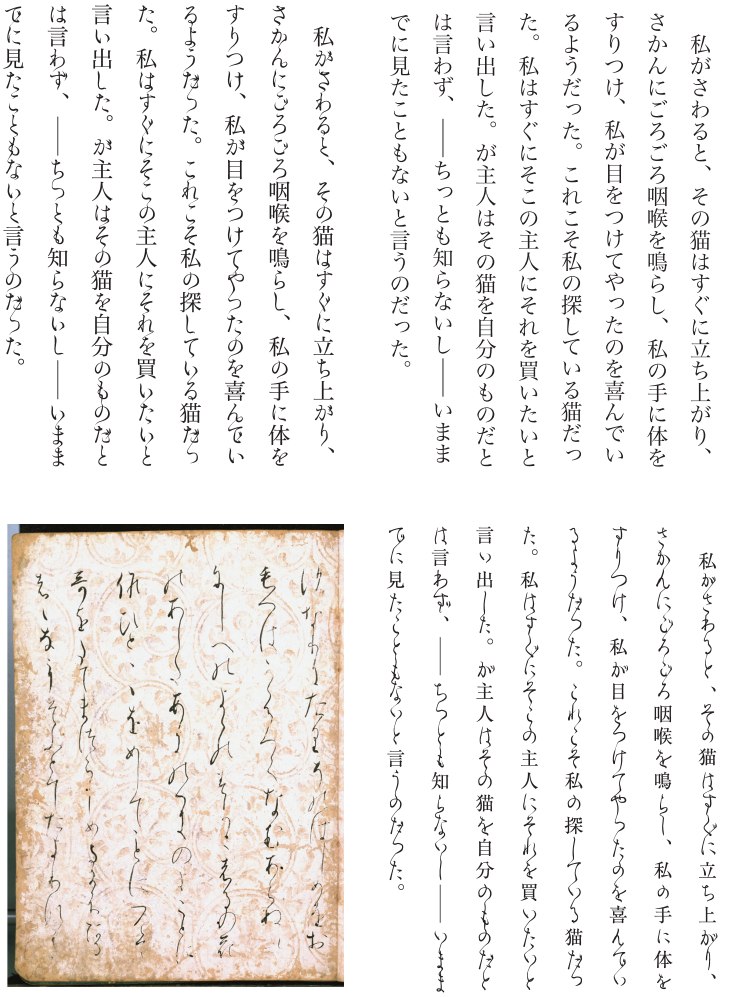
［4－5］一般的なフォントで組んだ例。使用したのは「ヒラギノ明朝 Pro-W3」である。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにそこの主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

［4－6］制作した「築地かな」による組見本。漢字部分は「ヒラギノ明朝 Pro-W3」。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにその主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

［4－7］制作した「さがのかな」による組見本。漢字部分は「リュウミン-R KL」。



［4－8］3つの組見本を縮小して並べた。左下は2章で引用した古今和歌集の図版。

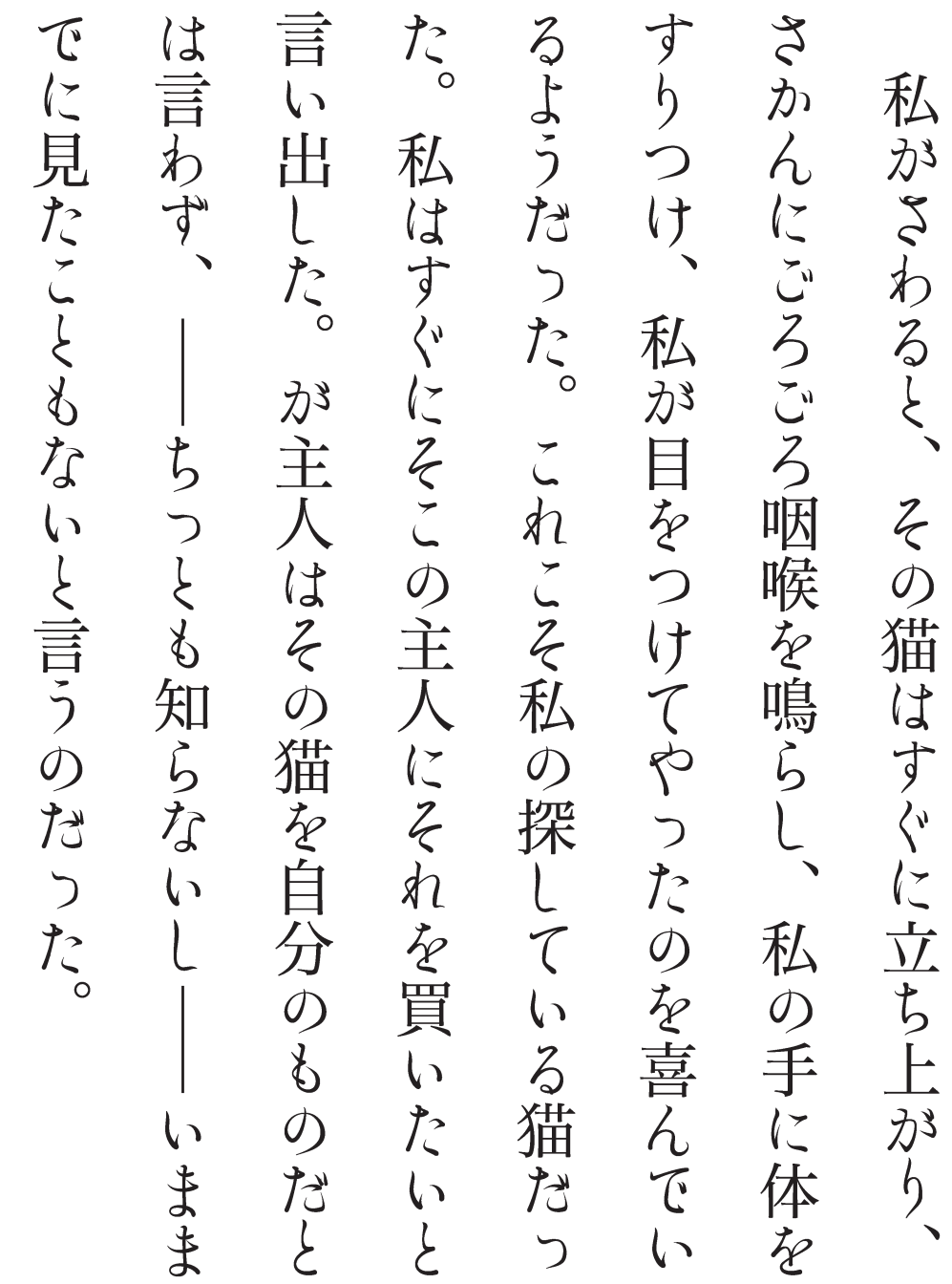
もちろん、筆文字を再現することが目的ではないが、組んだ面［図4－8］の印象や雰囲気は、既存のフォントで組んだものと比べ、より筆文字に近づいたことがわかると思う。筆の持つ表現力を生かしつつ、可読性が保たれていれば、この試みは成功といえるだろう。

4.3 手書きの書風から離れた表現

当初の目的では、手書きの書風を参考にした印刷用の文字の表現をさぐる、という意図があったが、もともと手書きのまったくの再現を目指すものではないならば、手書きの書風の作り方から離れた表現も可能ではないかと考えた。印刷媒体で扱われる文字として、結果に何かしらの新鮮さがあれば、あらたな文字表現として成立する可能性はある。そこで、「手書きの書風を参考にする」という意図からはなれた表現を模索した。

4.3.1 上下動を接続せずに、字形変化のみをさせる例

筆の上下動情報に平滑化をかけずに、字形のみを前後の文字によって変化させた例。



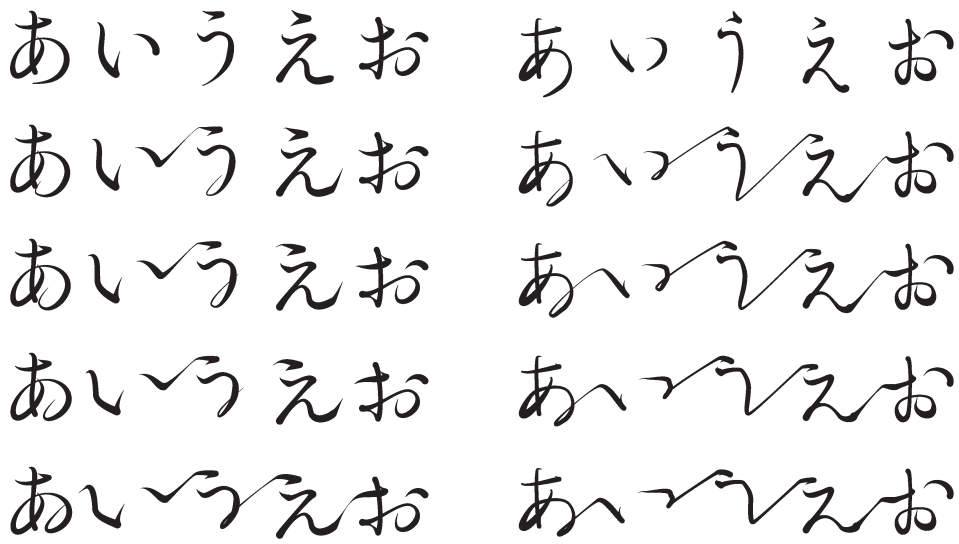
［4－9］筆の上下動を接続せず、骨格のみ変形をさせた場合。

ぱっと見たときの「いかにも変わった」という印象はないものの、もともと印刷物の書体は変化を嫌う分野でもあり、こうしたある程度抑制された変化のほうが受け入れられやすいかもしれない。多くの人は、読んでいる本の書体がどんなものか、などいちいち気にしないだろうが、それでも書体の違いによる印象や雰囲気の違いは、確実に読む人に影響を与えている。こうした気づかない程度の変化でも、長い文章として使った際には、十分な効果が期待できるのではないだろうか。

4.3.2 横組みの例

日本語は同じ文字で縦方向にも、横方向にも並べられるという、世界的に見ればかなり異質な性質を持つ文字だ。今回考案した字形変化のアルゴリズムを、そのまま横組みにも転用できないかと考え、横組みの機能も盛り込んだ。

縦組みと同様、制作したソフトウェア「KanaConstructor」のパラメータを調整することで、前後の文字の影響の度合いを変化させることができる［図4－10］。



［4－10］変形の度合いの調整が可能。左が「築地かな」、右が「さがのかな」。

続いて、長文の組見本も制作した。縦組みと同様、既存のフォントで組んだ例［図4－11］、「築地かな」で組んだ例［図4－12］、「さがのかな」で組んだ例［図4－13］を順に掲載する。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにそこの主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

［4－11］既存のフォントで組んだ例。使用したのは「ヒラギノ明朝 Pro-W3」。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにそこの主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

［4－12］制作した「築地かな」による横組みの見本。漢字部分は「ヒラギノ明朝 Pro-W3」。

私がさわると、その猫はすぐに立ち上がり、さかんにごろごろ咽喉を鳴らし、私の手に体をすりつけ、私が目をつけてやったのを喜んでいるようだった。これこそ私の探している猫だった。私はすぐにそこの主人にそれを買いたいと言い出した。が主人はその猫を自分のものだとは言わず、――ちっとも知らないし――いままでに見たこともないと言うのだった。

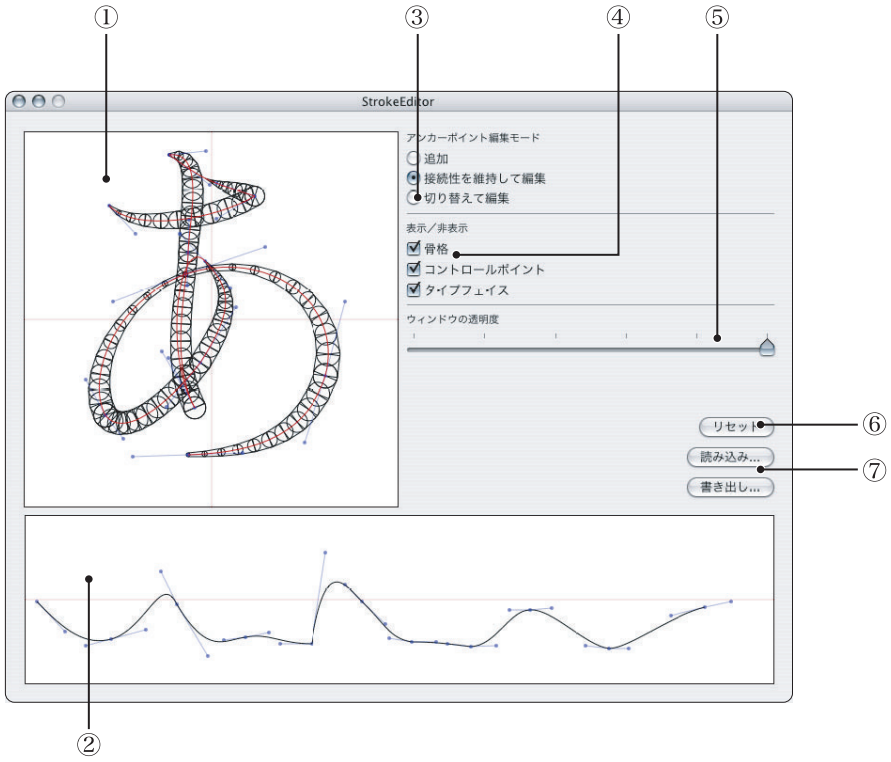
［4－13］制作した「さがのかな」による横組みの見本。漢字部分は「リュウミン-M KL」。

基本的に、伝統的な筆文字は横方向には書かないものだ。見慣れないためか、変形の影響力を高めたものは、最初はやや違和感を感じたが、筆者としては非常に興味深いものがあった。文字を扱う上では、違和感と新鮮さを区別することは難しい（おそらく、区別することはできないのだろう）。横組みについては、6章でもう少し踏み込んで考察を加えたい。

5 結果：ソフトウェアの概略

字形が変化するかな書体を再現するために、2つのソフトウェアを制作した。この章では、それぞれのソフトウェアの動作について解説をしていく。

5.1 文字データ制作アプリケーション “StrokeEditor”



［5－1］「StrokeEditor」のスクリーンショット。

5.1.1 インターフェイス

① 骨格表示領域

十字に薄く表示されているのは文字を描くときの基準となる線である。十字の交差点が文字のセンター・ポイントとなり、文字データを記録する際には、このセンター・ポイントを基準とした相対座標値が記録される。文字の骨格を作る際には、③で使用するツールを選択し、この領域上にマウスやペンタブレット等のデバイスを用いて実際に描いていくことになる。

② 筆上下動表示領域

基本的には①と同じ方法で描画する。天地中央に薄く表示されている赤線が仮想的な紙面となり、曲線で表示されているのが仮想的な筆の先端を表している。筆の先

端部が紙面よりも下にあるときに筆が接地していると判断し、紙面からの深さに応じて線の太さが決定され、タイプフェイスが生成される。

コントロールポイントの数を骨格表示領域と合わせることで、両者の位置の対応付けがなされる。

③ アンカーポイント追加モード

それぞれ Adobe Systems 社のドローアプリケーションである Illustrator のペンツール、ダイレクト選択ツール、切り替えツールに近い挙動をする。「追加」は①および②の任意の場所をドラッグすることで、ベジェ曲線のコントロールポイントを追加する。「接続性を維持して編集」は、ベジェ曲線の始点（あるいは終点）のコントロールポイントと、前後のコントロールポイント（変曲点）の位置関係を維持したまま座標を移動させることができる。「切り替えて編集」は、個々のコントロールポイントを独立して座標移動させるものである。全てマウスポインタのドラッグによって操作する。

④ 表示／非表示選択

それぞれ骨格・コントロールポイント・タイプフェイスを表示するかしないかが選択できる。

⑤ ウィンドウの透明度

ウィンドウ全体の透明度を調整できる。スキャナ等で取り込んだ手書きの原字等をトレースしやすくするためにつけた機能。取り込んだ原字の画像を画像ビューワ等で表示させた状態で、透明度を下げたウィンドウを重ねると、形を確認しながらトレースの作業ができる。

⑥ リセットボタン

骨格・筆上下動ともに初期化する。

⑦ 読み込みと書き出し

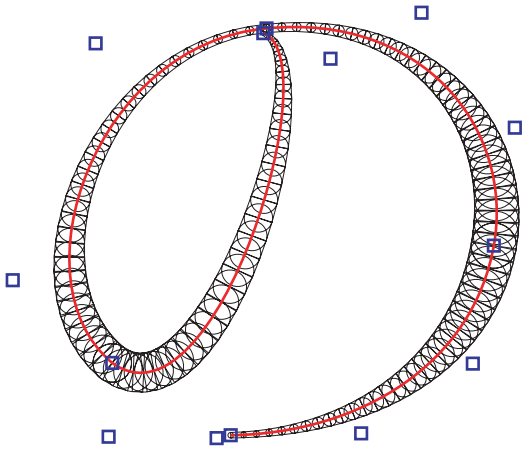
それぞれ Open、Save ダイアログにより、かなデータの読み込みと書き出しができる。かなデータは各文字ひとつずつ、拡張子 “.kana” という独自のバイナリ形式で保存される。また、字の内容と、ファイル名は一致していなければならない。たとえば「あ」のデータならファイル名は「あ.kana」となる。

また、基本的に、あらゆる操作はダイナミックに描画領域に反映される。文字の形をコントロールしやすくするための仕様である。

5.1.2 タイプフェイスの生成

文字ごとの骨格のデータ、および筆の上下動のデータは、“.kana” ファイルに保存されているが、その実体はベジェ曲線を定義するコントロールポイントの、座標値の集合である。

タイプフェイスの生成の流れは次の通り。



［5－2］「築地かな」の「の」の字の骨格とコントロールポイントを表示させた状態。
赤い線が骨格、青い四角がコントロールポイントである。実際のデータには、
コントロールポイントの位置が記録されており、赤い線はそこから曲線補完して導いたもの。
タイプフェイス（肉付け）の部分は、さらに筆の上下動の情報を取り出してから生成されたものである。

1　文字の骨格データを取り出す。各々の座標値をコントロールポイントとし、ベジェ曲線の形式で座標値間を補完する。制作したソフトウェア「KanaConstructor」では、コントロールポイント間の補完は30段階行っている。

2　筆の上下動のデータを取り出す。1と同様、各々の座標値のあいだをベジェ曲線の形式で補完し、仮想的に設定した紙面の高さとの差を基準として、仮想的な筆の幅の数値を決定する。

3　1で得られた座標値各々を中心として、2で得られた筆幅の円を描画する。

4　3で描かれた円と円を基準として、幾何学的な手法で、間をなめらかにつなぐ矩形を描画する。

骨格データと筆の上下動のデータの対応関係については、双方のコントロールポイントの数をまったく同じ数にすることで行う。もともとはプログラムの設計上、曲線補完をしやすくするための措置だったが、骨格の変化と筆の上下の変化は呼応しているので、文字の形を制御しやすいという利点もあった。

5.1.3 曲線補間法の選定について

制作したソフトウェアでは、文字の骨格、および筆の上下動を3次ベジェ曲線により記録し、描画している。パラメトリック曲線を数種類試した中で、最終的にはベジェ曲線を選ぶことになった。では、どうしてベジェ曲線を選択したのか。

例を挙げると、当初はB－スプライン曲線を利用していたが、文字同士の接続感が良好である反面、形の細かいコントロールにたいへん手間取った。コントロールポイントの位置を変えることによる骨格の変化が文字全体に及ぶので、骨格の形を思い通りにしにくいのである。また、急激な骨格の変化（折れ）を表現しづらいこともわかった。

今回の事例においては、（少々乱暴な一般化ではあるが）このように捉えることができるだろう。

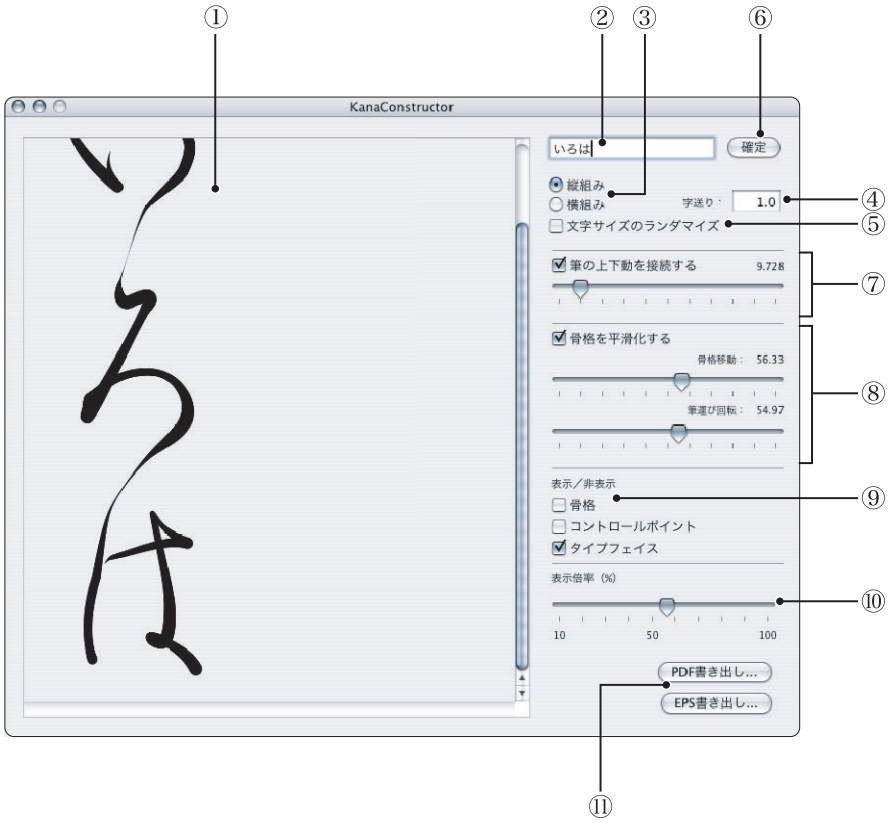
B–スプライン曲線：《接続感が良好（＝コーディングが簡単）で、文字のデザインが難解》

ベジェ曲線：《接続感が悪い（＝コーディングが複雑）で、文字のデザインが簡単》

最終的にはベジェ曲線を採用することになったが、それは書体制作において「数学的に、あるいはアルゴリズムの再現に適切という基準」ではなく、「最終的に文字をコントロールする人間が、もっとも理解しやすい形にするという基準」にすることが重要だと考えたためである。書体設計者はデザイナであってプログラマではない。デザイナが使い慣れている既存の書体設計ソフトウェアや Illustrator は、いずれも図形の表現にベジェ曲線を利用したものだ。また、ベジェ曲線はコントロールポイントの変化による曲線への影響を予測しやすく、直感的な操作に向いているという一般的な評価もある。

5.2 レイアウトアプリケーション “KanaConstructor”

5.2.1 インターフェイス



［5－3］「KanaConstructor」のスクリーンショット。

① 表示領域

生成された文字のタイプフェイスや骨格等はすべてここに描画される。

② テキストフィールド

扱いたい（描画させたい）文字列を入力する場所。

③ 縦組み・横組み選択

文字を縦方向に組むか、横方向に組むかを選択する。

④ 字送り設定

今回制作した書体は、仮想ボディの概念が無いため、本来、字送りという概念も存在しない。しかし、文字の生成結果のコントロールにはどうしても必要になるので、便宜上制作した。「StrokeEditor」の「骨格表示領域」の大きさを1として、ここで入力した数値が、そのまま文字のセンター・ポイントと次の文字のセンター・ポイントまでの距離となる。小さければ小さいほど、文字が「詰まった」状態で描画される。

⑤ 文字サイズのランダマイズ

おのおのの文字の大きさを、ある程度のランダム性を持って変化させるかさせないかを設定できる。

⑥ 確定ボタン

②〜⑤までの入力項目は、設計上、表示領域にダイナミックな反映がさせられないため設置した。②〜⑤までの項目に変化を加えたときは、このボタンを押すことで表示に反映させられる。

⑦ 筆の上下動を接続する

文字と文字の間の筆の上下動をなめらかに接続するかどうかを設定する。また、スライダによってその接続する度合いも0から100までの数値で設定できる。チェックボックスのチェックをはずせば、文字の終端と次の文字の始点は全く繋がらない。数値が高ければ高いほど、次の文字の始点までタイプフェイスが長く尾を引くような変化を生じる。

⑧ 骨格を平滑化する

文字の骨格データを平滑化するかどうかを設定できる。スライダの数値が双方とも0ならば、文字骨格は前後の文字の影響を受けない。「骨格移動」と「筆運び回転」は両方とも骨格の平滑化を成すための機能だが、それぞれ別のアルゴリズムによるもので、2つの数値の組み合わせによって、タイプフェイスの生成結果が変化する。アルゴリズムについては5.2.2を参照のこと。

⑨ 表示／非表示

骨格・コントロールポイント・タイプフェイスをそれぞれ表示するかしないかが設定できる。また、ここで表示状態になっているものは全て、PDF（またはEPS）書き出しの際に一緒に書き出される。

⑩ 表示倍率

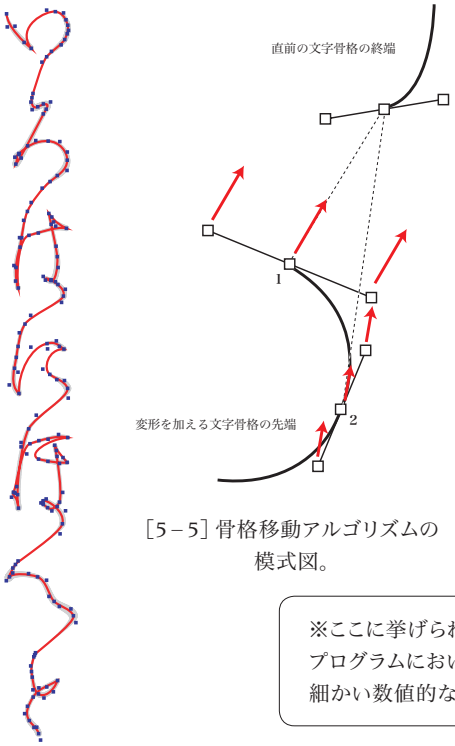
①の表示領域の表示倍率を10〜100%の間で設定できる。タイプフェイスの細かい部分を観察したり、全体を観察したいとき等、意図に合わせて自由に調整できる。

⑪ PDF 書き出し・EPS 書き出しボタン

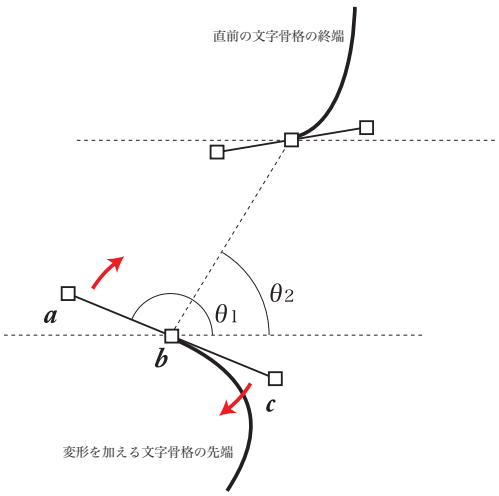
①の表示領域に表示されているタイプフェイスを、PDF（または EPS）形式で書き出すことができる。

5.2.2 文字骨格平滑化のアルゴリズム

骨格の接続、筆の上下動の平滑化は、パラメータの違いはあるものの、基本的には同じアルゴリズムにより行われている。すべての文字の骨格を繋げ、行全体を繋ぐ 1 本の骨格を作った［図 5－4］のち、「骨格移動」「骨格回転」のアルゴリズムにより平滑化を行う。5.1.1 のインターフェイスにおける、「骨格移動」と「筆運び回転」が、そのまま平滑化の強度に対応している。



［5－5］骨格移動アルゴリズムの模式図。



［5－6］骨格回転アルゴリズムの模式図。θ 1 と θ 2 の差を縮めるような挙動をする。

※ここに挙げられているものは模式図であり、実際のプログラムにおいては、ここに挙げたものに加えて、細かい数値的な調整は行っている。

［5－4］行全体の骨格とコントロールポイントを表示させた状態。
すべての文字の骨格が一本に繋がっている。

骨格移動：

各々の文字の骨格について、ベジェ曲線の始点（あるいは終点）のコントロールポイントと、前後のコントロールポイント（変曲点）の位置関係を維持したまま、前の文字の骨格の終点、後ろの文字の骨格の始点に「引っ張られるようにして」移動する［図 5－5］。移動距離は骨格の中心部に進めば進むほど対数的に減少する。この移動がコントロールポイントに及ぼす影響力は、KanaConstructor で設定されたパラメータ、文字の組み合わせによって前後するが、文字骨格の先端および終端から、およそ十数個の範囲になっている。

骨格回転：

「骨格移動」と同様、ベジェ曲線の始点（あるいは終点）のコントロールポイントと、前後のコントロールポイント（変曲点）の位置関係を維持したまま、前の文字の骨格の終点、後ろの文字の骨格の始点の方向へと回転させる［図5－6］。筆運びの接続感を上げるための処理である。「骨格移動」と同様、影響力は文字骨格の先端および終端から、およそ十数個の範囲になっている。

6 考察

前章の例はいずれも、今までの固定化された書体では再現できないものだ。もちろん、バランスよく、美しく見えるように調整を重ねたが、審美的な判断は個人差があるし、見慣れないものは即「美しくない」と考えられがちなので、生成結果の美しさについては個人的な判断は控えたい。しかし、前後の文字によって字形が変化する、という当初の目的は、一定の成果が上がったと考えている。

ここでは、筆者が今回の制作の中で興味を持った部分について数点、考察を加えたい。

6.1 制作したかな書体について

6.1.1 平滑化アルゴリズム

筆文字の崩し方は個人差が大きく、一概に決めることはできない。また必ずしも「元来の筆の動きを再現する」ことが良好な結果を生むとは限らないが、仮に、より筆の動きへと近づけるためには、平滑化のアルゴリズムにもさらに手を加える必要があるだろう。たとえば、文字骨格を1本としない、という方法も考えられる。筆文字の場合は、「お」や「む」の点などは、1行書き終わってから最後に付け足すという書き方がしばしば行われるためだ。

また、今回は接続、および平滑化アルゴリズムを一意に設定したが、手書きの文字を参考にするならば、当然ながら接続および平滑化の方法も複数、もしくは無限のヴァリエーションがあってしかるべきものだ。直前、直後だけでなく、より広範囲にわたった文字により影響を受け、変形させるという手法も考えられる。

だが、コンピュータプログラムとして実装するうえでは、何らかの形で接続方法も一意に（少なくとも、有限個には）定める必要がある。そうしないとプログラムとして表記できないからだ。

また、字形変化のための情報を、直前と直後の文字以上に広く取ろうとすると、計算量が等比的に増大してしまう。たとえば、すべての文章の内容を得てから、ひとつひとつの文字の字形を決定する、などという仕様は非現実的だ。どこかで線を引く必要がある。生成結果とコストパフォーマンスのバランスが取れる場所を模索する必要があるだろう。

6.2 固有性の与え方

6.2.1 手書き要素以外の方法

文字に固有性を与える方法として、本研究では個人の手書き文字の持つ固有性に注目し、「前後の文字」を拠り所として文字に固有性を持たせた。しかし、固有性を与える方法は、ほかにも様々なものが考えられるはずだ。3.1.2のBeowolfのように、ランダム性を使う方法、文章全体の中の位置を用いる方法、あるいは、文章の内容に踏み込んで何らかの変化をタイプフェイスに与える方法もあるかもしれない。今回の試みは、そうした数ある選択肢の中から、現実的に実装可能な方法を選び取ったものだ。印刷媒体の文字へ固有性を与える研究の、第一歩にすぎない。今後は、手書き文字の持つ要素とは別の基準をもとに文字に固有性を与える研究が生まれる可能性に期待したい。

6.2.2 DTPの構造上の問題

2章において、「日本の筆で書いた文字では、なだらかな大小の繰り返しや、左右の斜線による連綿など、文章に作者の感情が織り込まれた文字表現が行われていた」と書いた。しかし、当然のことながら、今回制作した書体では「作者の感情」なるものを再現することはできない。

現状のDTPのワークフローでは、基本的に作者の書いた文章はテキストデータとして組版者に渡され、レイアウトされることになる。もし「作者の感情」というものを組版にまで反映させたいならば、そうしたテキストデータに、何らかの方法で「感情を記録した」メタデータを添付する必要がある。

もちろん、そうした技術が実用化される可能性が無いわけではない。しかし、今やテキストデータというフォーマットはあまりに普及してしまい、それが特定の符号化方式に基づいた記録方法のひとつにすぎない、という意識すら失われている（原稿となるテキストデータでも、改行や段落分けは入れるだろう。本来は改行も段落分けも、厳密には組版の問題である。改行や段落分けには作者が介入するが、改ページ、改丁には介入しないのは、テキストデータというフォーマットと、作者と組版者相互間の暗黙の了解で決まってきたものに過ぎない）。

「感情」というものに限らず、固有性の拠り所として、筆者からテキスト以上の情報を得ようとする発想は自然なものだろう。だが、組版のためだけにメタデータを記録するようなフォーマットが普及するとは考えにくい。そのためにも、テキストデータが持つ情報の中だけから、何らかの方法で固有性を得る、という方向性が、今後の研究が進むべき方向性だと考えられる。

6.3 書体制作の意味の変化

今回、字形変化のアルゴリズムを設計するうえで、最も困難だったのが、「あらゆる文字の組み合わせに対して適応できる字形変化のアルゴリズムを見つけること」である。仮に「あ」と「い」が上手くなめらかに接続できるアルゴリズムを設計したとして、全く同じ方法で「い」と「う」が接続できるとは限らない。

書体を設計するうえで、いままでは、

「文字に汎用性を持たせる」

＝「文中のどこに置かれても違和感がない、汎用的な字形を考え、文字のかたちを決める」

という作業に重点が置かれていたが、これからは

「文字に固有性を持たせる」

＝「どのような文字の組み合わせに対しても適応する、字形変化のアルゴリズムを決める」

という作業へと変化するのではないだろうか。書体を設計するという行為が、汎用性を探ることから、固有性の「持たせ方」を探ることに変化するということだ。しかも、その固有性の「持たせ方」には、「どのような組み合わせにも対応できる」という、いままでとは別種類の汎用性が必要となると考えられる。

6.4 制作したソフトウェアについて

現状の DTP 環境で扱われているアウトライン型のフォントとは違うものを作るため、今回は行組版された PDF ファイルを制作し、それをレイアウトアプリケーション等で参照し、配置するという方法をとった。

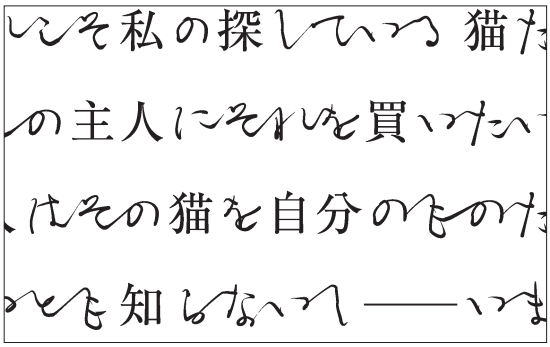
その方法自体は成功し、4章の作例も実際にレイアウトアプリケーションで作成したものだが、現状の仕様のままでは、DTP 環境に組み込むのは難しいだろう。理由は単純なもので、一度タイプフェイスが生成されると、再編集が不可能になってしまうことである。文字の差し替え、内容の変更があった場合は、タイプフェイス生成のソフトウェアまで戻り、PDF ファイルを書き出し直さなければならない。

本研究のように文字を動的に生成してしまうと、この文字はこの形、と一意に決めることができないので、テキストデータとタイプフェイス（ここでは「文字の形」そのもの）を分けて保存しておくことができない。これを解決するには、最終的にはレイアウトアプリケーションの中に、タイプフェイス生成エンジンを組み込み、アプリケーション内部では元になるテキストを保持したまま、接続と平滑化の結果をシミュレーションして表示するような機能がつく必要があるだろう。

本研究で扱った方法ではないにせよ、文字に固有性を持たせるためには、「文字1つにつきタイプフェイス1つ」という、現状のフォントデータの扱いを根本から見直す必要がある。今後のフォントベンダの動きに期待したい。

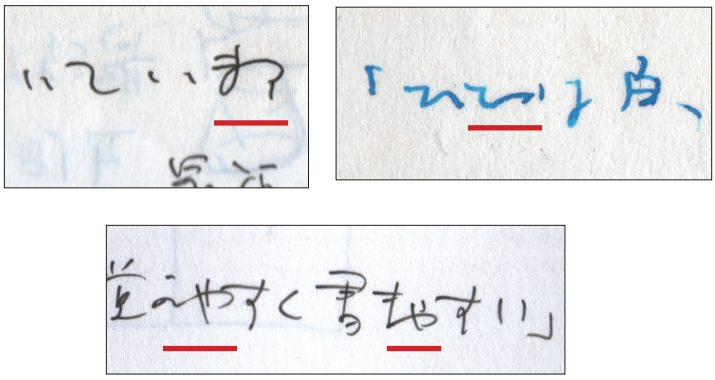
6.5 横組みについて

4.2で試みた、横組みの変化に興味深いものを感じた。ひとつは、見慣れないための違和感と同時に、ある種の新鮮さを感じたためだ。



［6－1］制作した横組みの組見本の一部

もうひとつは、この横組みの生成結果を見ると、「手書き文字を急いで書いた」ときと似た変形が見られたためだ。私は、今後こうした手書きの横書きがきっかけとなった「横組み専用の字形」というものが登場する可能性は十分にあると考えている。2章で触れたように、われわれの扱う文字、特にひらがなの字形がひとつに決まってしまったのは、活版が導入されて以降だ。それ以前の日本の文字は、たびたび形を変化させてきた（そもそも、ひらがなやカタカナですら、漢字から変化してできたものだ）。



［6－2］筆者の過去のノートより。左上は「ます」、右上は「とつ」、
下は「えや」「きや」が、変形して繋がってしまっている。

図6－2は筆者の使っていたノートの中から探して、抜き出してきたものである。急いで文字を書いているために、字形が変形してしまっている。「ます」が繋がってしまった例などは、思わず書いてしまった経験のある人も多いのではないだろうか。これは、横書きが一般化してはじめて生まれた、新しい字形変化の一例だと考えられる。

もともとひらがなの書き順は「上から下へ」を基本としており、縦書きを基準に書きやすく作られている。今後、ますます横書きが一般化するにつれ、自然と字形も変化していくことになるだろう。（義務教育で書き順を習っている以上、難しいだろうが）場合によっては、「左から右へ」を前提として、文字の書き順そのものも変わってしまう可能性すらあるだろう。

現状でも、PC 上では Web ブラウザやメーラ等、日常的に使うアプリケーションではほぼ全て横組みの日本語を扱っているし、実際に文字を書くときも、手紙を書く、などという特別な場合を除けばほとんど横書きにしているだろう。必然的に、日本語の組版はこれから横組みが中心へと変化していくはずだ。

活版の導入時期が遅いために歴史が浅い日本の文字組版の分野の中でも、横組みに関してはさらに歴史が浅い。始まったばかりだ、と言っても良いくらいだ。横組みに本研究のような筆文字の要素を取り入れることは、十分に大きな意味を持つだろう。より積極的に、種々の実験が行われるべきではないだろうか。

7 おわりに

今後、「固有性を持った文字」が普及する可能性はあるのだろうか。

手書きの影響を受けた文字は読みにくくなる危険をはらんでいるのかもしれない。2章でとりあげた古今和歌集なども、絵として、ものとして非常に美しいが、素人には資料を片手にしなければ読むこと自体難しい。

ひとつには、程度問題にすぎないという考え方がある。今回の研究では、ある程度の崩し方までに抑え、かつ、漢字は明朝体のものを流用することで可読性を保つことを狙っている。固有性を保持しつつ、誰しも読めうる状態を模索するという考え方である。

しかし、固有性を持った文字はもともと、読むことにある一定のハードルを設定してしまう指向性がある。

文章表現は、書くことも読むことも、印刷技術が普及する以前は一部の知識層のものであった。世界的にみれば識字率の向上が非常に早かった日本でも、一般市民の識字率は、江戸中期以降になってやっと、男性で50%、女性で15%程度だとされている。それまで書き文字とは、一部の人間の特殊な「たしなみ」だったのである。固有性をもった文字は、そうした状況に戻す方向性を、自ずと持っているといえる。

読める人が限定されてしまうような文字表現が大きく普及することは難しいと考えがちだが、固有性を持った文字が今までの印刷媒体の文字にはない、ある種の新鮮さを持つことは明らかだろう。2章で触れたように、印刷媒体の文字とはすなわち、汎用化された文字そのものだからだ。

たとえば読み解くのが難しい筆文字でも、読む相手がわかっていて、かつ相手が読めることが確実ならば使えるだろう。手紙などが良い例である。読者が特定できてさえいれば、読むことに対してある種の意図的なハードルを設定することができる。そのハードルとひきかえに、文字にある種の表現力が備わるならば、喜ばしいこととして、受け入れられる可能性はあるのではないか。

また、それと同時に、誰しもが確実に読める、という文字の用途も無くなることはないだろう。今後、書体開発の分野においては、誰にでも等しく閲覧できる「公的な文書のための汎用的文字」と、ある限定された用途に用いる「私的な固有文字」の二元化が起こるのではないだろうか。

謝　辞

この論文を書くにあたり、多くの方のお世話になりました。

まずは主査の佐藤雅彦先生、副査の安村通晃先生、有澤誠先生、相談に乗って頂いた永原康史先生、佐藤雅彦研究室の皆さん、そして、多くの美しい文字を残してくれた書体設計家の方たちへ、深く感謝致します。

参考資料

［日本語・和字・かな］

小松茂美著『かな　〜その成立と変遷〜』岩波新書　1968

紀田順一郎著『日本語大博物館』ジャストシステム　1994

板倉雅宣著『ヴィネット03号　和様ひらかな活字』朗文堂　2002

今田欣一著『ヴィネット05号　挑戦的和字の復刻』朗文堂　2002

味岡伸太郎著『日本語タイポグラフィの多様化の前提』リョービイマジクス　1992

永原康史著『日本語のデザイン』美術出版社　2002

［印刷技術・印刷史・書体史］

印刷博物誌編纂委員会編『印刷博物誌』紀伊國屋書店　2001

中根勝著『日本印刷技術史』八木書店　1999

藤枝晃著『文字の文化史』講談社　1999

京都造形芸術大学編『イラストレーションの展開とタイポグラフィの領域』角川書店　1998

河野三男著『タイポグラフィの領域』朗文堂　1998

白石和也・工藤剛・河地知木著『タイプフェイスとタイポグラフィ』九州大学出版会　1998

Manfred Klein・Yvonne Schwemer-Scheddin・Erik Spiekermann 著　組版工学研究会監訳『欧文書体入門　Type & Typographers』朗文堂　1992

［フォント関連技術］

Adobe Systems 著　アドビシステムズジャパン監訳『ページ記述言語 PostScript リファレンス・マニュアル』アスキー　1991

［曲線補完法］

桜井明監修　菅野敬祐・吉村和美・高山文雄著『C によるスプライン関数』東京電機大学出版局　1993

杉原厚吉著『グラフィックスの数理』共立出版　1995

［その他］

◎サンプル用かなデータの制作にあたって、リョービイマジクス株式会社「本明朝新かな」、大日本印刷株式会社「游築五号かな」、今田欣一氏制作のかな書体「さがの」を、骨格・筆運びの参考にしました。

◎組見本の文章は、青空文庫ウェブサイト（URI：http://www.aozora.gr.jp/）より、エドガー・アラン・ポー著　佐々木直次郎訳『黒猫』の一部を引用しました。

図版引用元リスト

（数字は図版番号）

1－1　印刷博物誌編纂委員会編『印刷博物誌』紀伊國屋書店　2001　P.879

1－2　印刷博物誌編纂委員会編『印刷博物誌』紀伊國屋書店　2001　P.855

1－3　永原康史著『日本語のデザイン』美術出版社　2002　P.110

1－5　田中一光・柏木博・東京都現代美術館・サントリーミュージアム〈天保山〉・朝日新聞著『田中一光回顧展　われらデザインの時代』朝日新聞社　2003　P.162／179

2－7　Emil Ruder『Typography』Ram Publications　1996　P.25

2－8　e 国宝（URI：http://www.emuseum.jp/）　東京国立博物館所蔵

2－9　株式会社林田プロジェクト・松本八郎・佐藤章・高田行庸・森純一郎・安斎徹雄・藤城雅彦・島田拓史企画編集『プリンティングカルチャー』ミズノプリンティングミュージアム　1993　P.119

2－10　板倉雅宣著『ヴィネット03号　和様ひらかな活字』朗文堂　2002　P.30

2－11　Adobe Systems『OpenType User Guide for Adobe Fonts』Adobe Systems　2003　P.10／11

3－1　Jan Tschichold 著　菅井暢子訳『書物と活字』朗文堂　1998　P.129

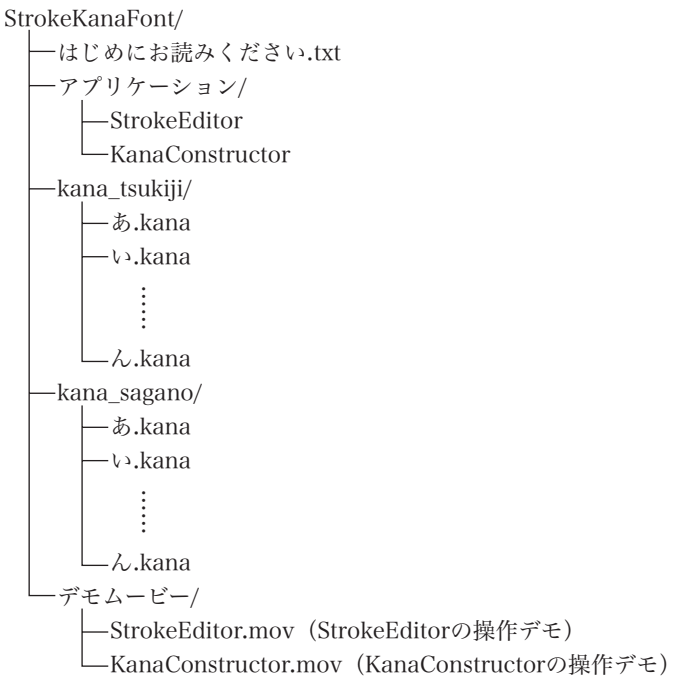
3－3　Manfred Klein・Yvonne Schwemer-Scheddin・Erik Spiekermann 著　組版工学研究会監訳『欧文書体入門　Type & Typographers』朗文堂　1992　P.29

3－4　猪股裕一監修『アドビ公認　デジタル・パブリッシング・ガイド』エムディエヌコーポレーション　1999　P.127

付　録

本論文には、付録として CD を添付しています。
制作したソフトウェアのデモ・ムービーと、ソフトウェア本体、制作したかな文字のデータが収録されています。

［CD の内容］



［注意事項］

◎本ソフトウェアは、研究目的で制作されたものであり、基本的に他者が利用することを想定していません（一部、動作が不安定な部分があります）。研究目的で参照する以外の目的で利用しないでください。
◎本ソフトウェアの複製・再配布を禁じます。
◎添付のかなデータを利用して制作された文字（生成されたタイプフェイス）を、あらゆる形式において、他の場所へ転用することを禁止します。
◎本ソフトウェアを用いた事による不利益・損害等は、作者はその責任を負いません。

［利用方法］

◎「StrokeEditor」「KanaConstructor」がアプリケーション本体です。ハードディスク上の任意の場所にコピーしてから起動してください。
◎「kana_tsukiji」「kana_sagano」ディレクトリには、それぞれ制作した「築地かな」「さがのかな」の文字データ（.kana ファイル）が入っています。使用したい方のディレクトリを、使用する

Macintosh のホームディレクトリの直下にコピーしたのち、ディレクトリ名を「kana」へリネームしてください。自分で新たなかな文字のデータを制作した場合も、ホームディレクトリ直下へ入れた kana ディレクトリの中へ入れてください。

◎論文第5章に、簡単な機能解説を記載しています。

◎かなデータは、すべてのかな文字について制作したわけではありません。片仮名や濁音・半濁音のうちいくつかはデータの無い文字があります。含まれない文字は空白として処理されます。

［動作環境］

◎ Mac OS X 10.2以降が動作する Macintosh で、日本語入力が可能な環境であること。作者は PowerMacG5/1.6GHz、Mac OS X 10.3.2の環境で制作し、動作を確認しました。