

B-スプライン関数を利用した連綿体かなフォントの生成*

3R-1

中田 浩暁 狩野 弘之†

東京電機大学 大学院理工学研究科‡

1. まえがき

文字フォントの生成方式は、ドット行列方式、輪郭ベクトル方式、骨格ベクトル方式、の3つに大別されてきた。これらの方式はいずれも、個々の文字を静的 (static) なパターンと捉えて生成する方法である。これに対して、人が文字を書くときのように筆記具を空間的かつ時間的に連続に移動させた結果として、すなわち動的 (dynamic) に生成されるという考えに基づいて“ダイナミックフォント”が考えられた [1]。本研究では、この生成方式を用いたフォントの設計を対話的に行なうフォントエディタを開発した。また、実際にこのエディタを用いて、続け字を可能とする毛筆体の連綿体かなフォントの試作を行なった。さらに、平滑化スプライン [2] を応用して、フォントに対する平滑化の演算を定義し、省略した字形を得る方法を提案する。

2. フォントの生成法

フォント生成の概念図を図 1 に示す。すなわち仮想的な筆記面と筆記具を考え、筆記面は $O-XY$ 平面、筆記具の形状は円錐で、その軸は Z 軸に平行であるとす。フォントは、筆記具が設計された書字運動をしたときの筆記面との交差領域の痕跡として形成されていく。

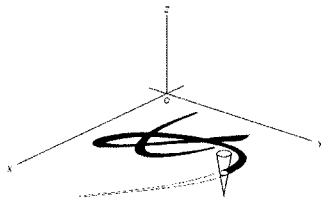


図 1: フォント生成の概念図

筆記具先端の軌道 $p(t)$ は次式によって設計される。

$$p(t) = \sum_{i=l-k+1}^{m+k-1} p_i^* B_k(\alpha(t-t_i)) \quad (1)$$

ここで、 $B_k(\cdot)$ は整数の節点をもつ正規化された一様な k 次 B-スプライン関数であり、これにより生成される 3 次元運動は“単位運動”とよばれている。また α は実

*Generation of Japanese Cursive Fonts Using B-Spline Functions

†Hiroaki Nakata, Hiroyuki Kano

‡Graduate School of Science & Engineering, Tokyo Denki University

時間における節点 t_i の間隔を調節する定数、 p_i^* は 3 次元重みベクトル (制御点)、 m, l は制御点の個数などに関係する整数である。制御点の系列

$$M = p_{l-k+1}^* \cdots p_{m+k-1}^* \quad (2)$$

は“制御多角形”を指定し、幾何学的には運動軌道の輪郭を表す。書字運動は単位運動に対する重み係数を指定することによって生成され、従って制御多角形をその運動に対する形式的な表現、フォントの設計は制御多角形の設計問題、と見なすことができる。

3. フォントエディタ

以上より、フォントの形状を設計することは制御多角形を決定することに帰着し、そこでフォントの設計を対話的におこなうソフトウェア、すなわちフォントエディタを Linux OS で動作する X-Window 上で開発した。(図 2 に実際の画面を示す)

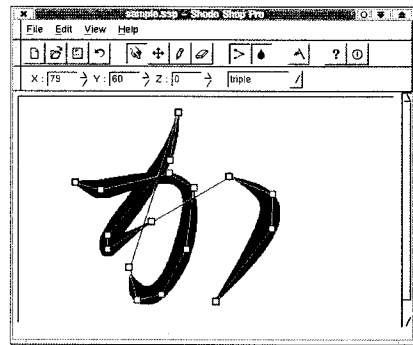


図 2: フォントエディタ

エディタによって、制御点の追加や削除、および移動 (X, Y, Z 方向の重み係数の変更) といった作業を主にマウスを用いておこなうことができる。さらに、拡大・縮小、平行移動、および回転といったフォントに対する種々の演算は制御多角形 M に対する演算として形式的に実現され [1]、これらの演算もエディタの機能として含まれている。

生成されるフォント形状の、計算と描画に要する時間は十分に短く、これによりユーザーは字形を対話的にデザインすることが可能である。また、編集した個々のフォントは制御多角形として簡潔に表現し、保存することができ、ここでは XML フォーマットにしたがって制御点列のデータを記述している。

4. 連綿体かなフォント

前節のエディタを用いて、続け字を可能とする毛筆体の連綿体かなフォント(48文字)を試作した。本生成方式で文字同士の連綿を行なうには、各文字に対応する制御多角形同士をどのように接続するかについてのみ考えればよい。書道において、連綿は上の字の収筆および下の字の起筆の形状、位置、方向などによって規定される[3]。ここでは連綿のパターンを分類する方法[4]を参考にし、続け方判定テーブルを参照して判定できるようにした。

図3に文字‘と’、‘り’ (および各々の制御多角形 M_1, M_2) とそれらを連綿させた例を示す ($k = 3$ とした)。この場合、単語‘とり’は制御多角形 $M = M_1' M_2'$ で生成されている。ただし、 M_1', M_2' は M_1, M_2 の平行移動と、接続部での制御点の変更によって得られる。‘と’の収筆部と‘り’の起筆が自然に接続されていることがわかる。

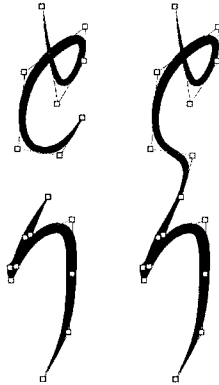


図3: 連綿体かなフォント‘と’, ‘り’, および‘とり’

5. フォントの平滑化

書道においてしばしば見られる、かなの省略した字形を体系的な方法で求めることはフォントのデザイナーにとって有益であると考えられる。そこで、データ点のスムージングに用いられる平滑化スプライン[2]を応用し、フォントにおける平滑化の演算を提案する。

(1)で与えられる標準のフォントに対して、次のようなフォント生成軌道のファミリーを考える。

$$\mathcal{X} = \left\{ x(t) : x(t) = \sum_{i=l-k+1}^{m+k-1} \tau_i B_k(\alpha(t-t_i)) \right\} \quad (3)$$

ここで、 $\{\tau_{l-k+1}, \dots, \tau_{m+k-1}\}$ は平滑化されたフォントに対応する制御点である。なお、以下では3次元軌道の各要素に対して独立に平滑化を適用する。このとき、平滑化の演算は次式の最適化問題によって定式化される。

$$\min_{x \in \mathcal{X}} \left(\lambda \int_{-\infty}^{\infty} \{x''(t)\}^2 dt + \sum_{i=l-k+1}^{m+k-1} w_i (p_i - \tau_i)^2 \right) \quad (4)$$

ここで、 λ は書字運動 $x(t)$ の滑らかさを決定する正定数スカラー、 $w_i \in [0, 1]$ は各 i 番目の制御点の信頼度を

表すパラメーターである。(4)に対する解は以下によって一意に与えられる。

$$\tau = (\lambda V + W)^{-1} W p. \quad (5)$$

$\tau = [\tau_{l-k+1} \dots \tau_{m+k-1}]^T$, $p = [p_{l-k+1} \dots p_{m+k-1}]^T$, $W = \text{diag}\{w_{l-k+1}, \dots, w_{m+k-1}\}$, そして $V = [v_{i,j}]$ は B-スプライン関数 $B_k(\cdot)$ から一意に決定される正定値のグラミアンである。

以下に平滑化によって変形されたフォントの例を示す。2つのフォント‘か’、‘き’に対する制御多角形をそれぞれ M_1, M_2 とするとき、図4の左は制御多角形 $M = M_1 M_2$ から構成されるフォント‘かき’であり、中央と右は M に対してそれぞれ $\lambda = 0.5, 2.0$ (いずれも $W = I$) として平滑化の演算を適用した結果である。

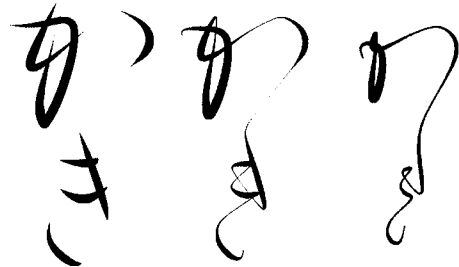


図4: 平滑化操作による書体の変形

6. まとめ

ダイナミックフォントの生成方式に基づき、フォントの設計を支援するツールを作成し、さらに連綿体かなフォント、および平滑化の演算によって省略形のフォントを生成する方法を提案した。この方法の特徴は、個々のフォントや連綿体フォント、さらに様々な変形フォントを統一的な枠組のもとに生成できることである。

参考文献

- [1] K.Takayama and H.Kano: A New Approach to Synthesizing Free Motions of Robotic Manipulators Based on a Concept of Unit Motions, *IEEE Trans. SMC.*, Vol. 25, No. 3, pp. 453-463, March, 1995.
- [2] B. W. Silverman, Spline Smoothing: The Equivalent Variable Kernel Method, *The Annals of Statistics*, Vol. 12, No. 3, pp. 898-916, 1984.
- [3] 榎倉香邨: 書道かな, 日本放送協会出版, 1998.
- [4] 戸倉毅, 鈴木隆子, 中村浩子, 牧野優子, 高倉穂: つづけ字を可能とする毛筆体文字生成システム, 情報処理学会論文誌, Vol. 29, No. 1, pp. 20-28, Jan., 1988.