Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Modeling and Manipulating Dynamic Font-based Hairy Brush Characters using Control-Theoretic B-spline Approach

Xie Zhenyu * **Hiroyuki Fujioka** ** **Akinori Hidaka** *** **Hiroyuki Kano** ***

* *Graduate School of Engineering, Fukuoka Institute of Technology, Fukuoka 811-0295, Japan.*
** *Department of System Management, Fukuoka Institute of Technology, Fukuoka 811-0295, Japan.*
*** *Division of Science, Tokyo Denki University, Saitama 350-0394, Japan.*

Abstract: In this study, we consider a problem of modeling and manipulating hairy-brush characters based on the so-called 'dynamic font' method with control-theoretic B-spline approach, in which the characters are constituted as a result of trajectory curves using normalized uniform B-splines as basis function. First, typefaces of pre-designed dynamic font characters with one-pixel thickness are transformed to hairy-brush ones by introducing a deep learning method 'Pix2Pix'. We then develop a method for modeling hairy-brush characters by formulating the problem as an optimal function-approximation problem, which minimizes an input energy of writing trajectory. Also, manipulations for generating cursive words as seen in Japanese calligraphy in a systematic way are described. A design example of cursive words is included.

Keywords: modeling and manipulation; hairy-brush characters; dynamic font; control-theoretic B-splines; optimal function approximation; deep learning.

1. INTRODUCTION

Spline functions have been studied and used extensively in many industrial applications -such as robotic path following problem (Gill (2015)) and image registration (Sun (2017)), etc. Egerstedt and Martin have studied the theory of such splines from the viewpoint of optimal control theory (e.g. (Egerstedt and Martin (2010))), which has been called as 'dynamic splines'. Inspired by their works, we have also studied B-splines from that viewpoint (e.g. (Kano (2003))), which will be referred to as 'control-theoretic B-splines' in this paper.

As an application using such control-theoretic B-splines, this paper focuses on the problem of generating and manipulating characters based on the so-called 'dynamic font' method (Takayama (1996)). The dynamic font method has been developed by mimicking the writing process by humans, in which characters are generated by moving a writing device on a writing plane continuously in both time and space. Unlike the ordinary design methods -such as dot matrix, outline vector, and skeleton vector methods, etc. (e.g. (Uehara (1990))), this method is powerful, particularly when we want to generate and manipulate characters in Japanese calligraphy where the thickness of the stroke is important. For example, we have developed a scheme for generating cursive characters by employing control-theoretic B-spline function approximation (Fujioka (2006)). But, in these design, hairy-brush ink effect on characters as seen in a real calligraphy has never been considered.

A typical approach for generating such a hairy-brush ink effect on characters may be to employ model-based one. For example, Wong and Ip (Wong (2000)) have developed a writing device model that can create characters with hairy-brush ink effects by using a parameterized model which captures (i) the writing brush 3D geometry, (ii) the brush hair properties and (iii) the variations of ink deposition along a stroke trajectory. Similar works have been exhibited by various groups (e.g. (Mi (2004), Xu (2002))). However, optimally setting the parameters required in the models may be a complex task for finely generating hairy-brush ink effect. Also, excessive computational time is required as the fineness of hairy-brush ink becomes high. Another approach for the above issue is to introduce a deep-learning approach. In Lyu's work (Lyu (2017)), the calligraphy synthesis problem has been treated as an image-to-image translation problem and then a deep neural network-based model is developed to generate calligraphy images from standard character font images directly. Once such a network is trained, the fineness of hairy-brush ink may be achieved independently with the computational time. However, manipulating such characters may be difficult since these characters are treated as planar static patterns.

This paper is a continuation of our studies on the dynamic font method in (Fujioka (2006)). In particular, we here

Copyright lies with the authors

consider the problem of constructing and manipulating (in particular, reconstructing cursive) dynamic font-based hairy-brush characters using the control-theoretic B-spline approach. Therein, introducing a deep learning approach, the pre-designed characters with one-pixel thickness are transformed to hairy-brush ones with some ink effects. As a deep learning approach, we here employ 'Pix2Pix' (Isola (2017)) which is a conditional Generative Adversarial Network (cGAN) model for general-purpose image-to-image translation. We then develop a method of modeling such hairy-brush characters in a scheme of dynamic font by employing the control-theoretic function approximation theory. The basic idea may be similar to Lyu's work (Lyu (2017)) using the deep learning approach, but the big difference is on the manipulability based on the dynamic font method. Therefore, we present that the characters modeled in this way are connected and reconstructed in a systematic way to yield cursive words as seen in Japanese calligraphy.

The paper is organized as follows. In Section 2, we briefly present the dynamic font method and deep learning method 'Pix2Pix' as preliminaries. Then in Section 3, we develop a scheme for modeling the dynamic font-based hairy-brush characters based on the control-theoretic B-spline smoothing. Also, the method for manipulating characters is described in Section 4. In Section 5, a design example is included, and concluding remarks are given in Section 6.

2. PRELIMINARIES

As preliminaries, we briefly present the dynamic font (Takayama (1996)) and Pix2Pix (Isola (2017)), which will be used to develop a scheme for modeling hairy-brush characters in Section 3.

2.1 Dynamic Font

Figure 1 illustrates a dynamic font model for generating characters. We here consider a virtual writing device modeled by a circular cone moving in 3-dimensional space O-XYZ and a virtual writing plane O-XY. Characters are then obtained by moving the device in the space according to the designed writing motion and then by deriving the cross-sectional area on the writing plane. As the cone, say its tip, moves along a trajectory, the circle moves in the plane and results in a character.

Using such a method, we here suppose that a character with one-pixel thickness is pre-designed. Then, the corresponding writing motion $x_d(t) = [X_d(t)\ Y_d(t)]^T \in \mathbf{R}^2,\ t \in [t_0, t_m]$ on 2-dimensional plane O-XY is designed as

$$x_d(t) = \sum_{i=-3}^{m-1} p_i B_3(\alpha(t - t_i)). \quad (1)$$

Here, $B_3(\cdot)$ is a normalized, uniform cubic B-splines (e.g. (Takayama (1996))), m is an integer that determines the time interval of motion, p_i's are 2-dimensional weighting vectors called "control points", and $\alpha(> 0)$ is a scalar for scaling the interval between equally-spaced knot points t_i with $t_{i+1} - t_i = 1/\alpha$.

Remark 1. As we see from (1), once an appropriate α and m are chosen, designing characters reduce to that of a sequence of control points called as "control polygon" $\mathcal{M}_d$ defined by

$$\mathcal{M}_d = p_{-3}\ p_{-2}\ \cdots p_{m-1}. \quad (2)$$

The control polygon $\mathcal{M}_d$ represents a geometrical outline for writing motion. Thus, $\mathcal{M}_d$ can be regarded as a formal representation of the writing motion $x_d(t)$ and hence the character.

2.2 Pix2Pix

We next present "Pix2Pix" architecture (Isola (2017)), which will be used to transform the pre-designed characters with one-pixel thickness in Section 2.1 to hairy-brush ones.

The architecture is illustrated in Fig. 2, in which a conditional generative adversarial network (cGAN) is used to learn a function to map from an input image (i.e. pre-designed character) to an output image (i.e. hairy-brush character).

The network consists of two main pieces, i.e. 'Generator' G and 'Discriminator' D. The Generator transforms the input image x to the output image G(x), where U-net in Ronneberger's work (Ronneberger (2015)) is employed as the generator. On the other hand, the Discriminator measures the similarity of the input image x to either a target image y from the dataset or an output image G(x) from the generator and then discriminates if this was produced by the generator. These G and D are trained

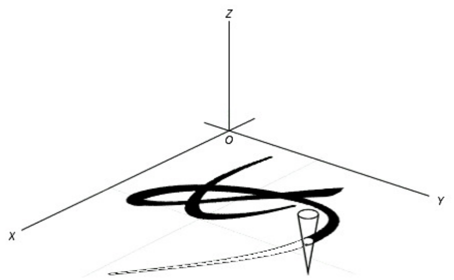

Fig. 1. Dynamic font model.

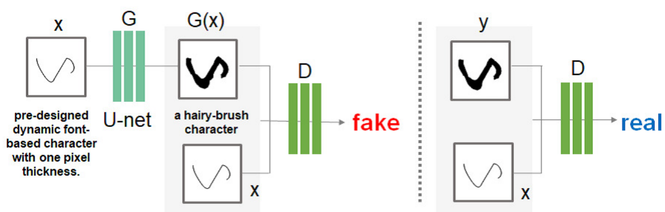

Fig. 2. Pix2Pix architecture.

(a) output (b) input

Fig. 3. An example of training dataset.

simultaneously in an adversarial process so that G seeks to better fool the discriminator D and also D seeks to better identify the counterfeit images, i.e. G(x).

As training dataset, we here used 2073 paired character images, where Kouzan hairy-brush font images (Aoyanagi (2016)) with gray-scale values and the corresponding binary images thinned by Zhang-Suen method (Gonzalez (2004)) are used as output and input images, respectively. An example of output and input images is illustrated in Fig. 3.

3. MODELING HAIRY-BRUSH CHARACTERS USING CONTROL-THEORETIC B-SPLINE APPROACH

We are now in the position to develop a method to model the dynamic font-based hairy-brush characters by employing a control-theoretic function approximation theory.

3.1 Problem Statement

Now suppose that we are given some pre-designed characters whose typefaces were transformed by employing Pix2Pix in Section 2.2 (see Fig. 4). From the relation of training dataset, we may regard the pre-designed character as the skeleton of the transformed character (i.e. red lines in Fig. 4).

Let $\mathbf{n}(t) \in \mathbf{R}^2$ be a unit normal vector, denoted as the red arrow in Fig. 4, to the motion of pre-designed characters on O-XY plane, say $x_d(t)$ in (1). Then we can compute $\mathbf{n}(t)$ as

$$\mathbf{n}(t) = \frac{1}{\sqrt{(\dot{X}_d(t))^2 + (\dot{Y}_d(t))^2}}[-\dot{Y}_d(t)\ \dot{X}_d(t)]^T, \qquad (3)$$

where $\dot{X}_d(t)$ and $\dot{Y}_d(t)$ denote $\frac{dX_d(t)}{dt}$ and $\frac{dY_d(t)}{dt}$ respectively. Moreover, we obtain a set of hairy-brush ink data by scanning the transformed characters to the normal direction $\mathbf{n}(s_i)$ from $x_d(s_i)$ at a sampling time $s_i \in [t_0, t_m]$ for $i = 1, 2, \cdots, N$ as follows.

Let $x_d^k(s_i) = [X_d^k(s_i)\ Y_d^k(s_i)]^T \in \mathbf{R}^2$ be a point given by

$$x_d^k(s_i) = x_d(s_i) + ak \cdot \mathbf{n}(s_i) \qquad (4)$$

with $a(> 0) \in \mathbf{R}$ and $v^k(s_i) \in \mathbf{N}$ be the gray-scale value on $x_d^k(s_i)$ at a sampling time s_i. Here, $k \in [k_L, k_U] \subset \mathbf{Z}$ is an index of scanning line given by $x_d^k(t)$. Also, $k_L, k_U \in \mathbf{Z}$ are set so that the region between $x_d^{k_L}(t)$ and $x_d^{k_U}(t)$ covers the whole character.

We thus measure and store a set of hairy-brush ink data for each k, denoted as $\mathcal{D}_k$, as

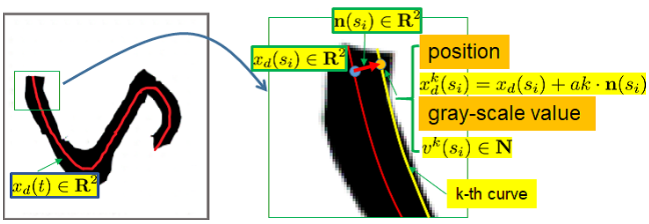

Fig. 4. Overview of pre-designed character.

$$\mathcal{D}_k = \{(d_i^k, s_i) : d_i^k = [(x_d^k(s_i))^T\ v^k(s_i)]^T \in \mathbf{R}^3,$$
$$s_i \in [t_0, t_m], i = 1, 2, \cdots, N\}. \qquad (5)$$

Then, we consider the following problem based on the idea of 'dynamic font' in Section 2.1:

Problem 1. Suppose that a set of hairy-brush ink data for each k, i.e. $\mathcal{D}_k$ in (5) is given and that an appropriate α and m are chosen. Construct the corresponding hairy-brush ink motion $x_k(t) = [X_k(t)\ Y_k(t)\ v_k(t)]^T \in \mathbf{R}^3,\ t \in [t_0, t_m]$ for index k, which can be represented with the same form as (1), i.e.

$$x_k(t) = \sum_{i=-3}^{m-1} p_i^k B_3(\alpha(t - t_i)), \qquad (6)$$

where $p_i^k \in \mathbf{R}^3$ are control points .

Equivalently, letting $\mathcal{M}_k$ be a control polygon given as

$$\mathcal{M}_k = p_{-3}^k p_{-2}^k \cdots p_{m-1}^k, \qquad (7)$$

we may notice from Remark 1 in Section 2.1 that Problem 1 can be regarded to find $\mathcal{M}$ given by

$$\mathcal{M} = \biguplus_{k=k_L}^{k_U} \mathcal{M}_k, \qquad (8)$$

which is a formal representation of the hairy-brush characters. Also, $\biguplus$ denotes the multiset union. Then, hairy-brush characters are constructed by filling the point $(X_k(t)\ Y_k(t))$ with the gray-scale value $v_k(t)$ for $\forall t \in [t_0, t_m]$ and $\forall k \in [k_L, k_U]$, which are determined from $\mathcal{M}$.

3.2 Control-Theoretic B-splines

For solving Problem 1, we here introduce the control-theoretic B-spline approximation theory (Fujioka (2006)). For the sake of simplicity, we here assume as follows:

(A1) The parameter α is set as $\alpha = 1$, and $t_i = i$ for $i = -3, -2, \cdots, m$.

(A2) Each element in $x_k(t) \in \mathbf{R}^3$ is modeled independently, and using notational abuse, p_i^k, d_i^k are assumed to be one-dimensional with the understanding that they represent one of the three elements.

Now, let us consider a third order linear system

$$\frac{d^3}{dt^3} x_k(t) = u_k(t),\ x_k(-3) = \dot{x}_k(-3) = \ddot{x}_k(-3) = 0, \qquad (9)$$

and a restricted set of controls

$$\mathcal{U}_k = \left\{ u_k(t)\ :\ u_k(t) = \sum_{i=-3}^{m-1} p_i^k \frac{d^3}{dt^3} B_3(t - i),\ p_i^k \in \mathbf{R} \right\}. \qquad (10)$$

Here, we assume $m \geq 5$ and that the first and last three coefficients,

$$p_k^I = \begin{bmatrix} p_{-3}^k \\ p_{-2}^k \\ p_{-1}^k \end{bmatrix},\ p_k^F = \begin{bmatrix} p_{m-3}^k \\ p_{m-2}^k \\ p_{m-1}^k \end{bmatrix} \qquad (11)$$

are prescribed.

Remark 2. In (11), we here prescribe p_k^I and p_k^F as

$$p_k^I = d_1^k \cdot \mathbf{1}_3,\ p_k^F = d_N^k \cdot \mathbf{1}_3 \qquad (12)$$

with $\mathbf{1}_3 = [1\ 1\ 1]^T \in \mathbf{R}^3$, which implies that an motion $x_k(t)$ in (6) coincides with the data and is stationary on starting and ending, i.e.

$$x_k(0) = d_1^k,\ \dot{x}_k(0) = \ddot{x}_k(0) = 0$$
$$x_k(m) = d_N^k,\ \dot{x}_k(m) = \ddot{x}_k(m) = 0. \tag{13}$$

We then consider an approximation problem formulated as the following control problem:

Problem 2. Find an optimal input $u_k(t)$ and solution $x_k(t)$ such that

$$\min_{u_k \in \mathcal{U}_k} J_k(u_k) \tag{14}$$

where

$$J_k(u_k) = \lambda \int_{-\infty}^{+\infty} u_k^2(t)dt + \sum_{i=1}^{N} w_i (x_k(s_i) - d_i^k)^2 \tag{15}$$

with $\lambda > 0$ and $0 < w_i \leq 1$ for $i = 1, 2, \cdots, N$.

Remark 3. In Problem 2, we note that λ and w_i are introduced in order to adjust the weights on input energy and the goodness of fit to the given data $\mathcal{D}_k$ in (5).

3.3 Optimal Solution

Note that the Problem 2 is of finite-dimensional since the space of control $\mathcal{U}_k$ is finite, and can be solved as follows (see (Fujioka (2006)) for the details): Letting $p_k, b_3(t) \in \mathbf{R}^M (M = m+3)$ be vectors defined as

$$p_k = [p_{-3}^k\ p_{-2}^k\ \cdots\ p_{m-1}^k]^T \tag{16}$$
$$b_3(t) = [B_3(t+3)\ B_3(t+2)\ \cdots\ B_3(t-(m-1))]^T, \tag{17}$$

we can rewrite (6) as

$$x_k(t) = p_k^T b_3(t). \tag{18}$$

Using such an expression, the cost $J_k(u_k)$ in (15) can be expressed in terms of p_k as $\bar{J}_k(p_k)$,

$$\bar{J}_k(p_k) = \lambda p_k^T Q_M p_k + (Bp_k - d_k)^T W (Bp_k - d_k) \tag{19}$$

which now is to be minimized with respect to

$$\hat{p}_k = \left[p_0^k\ p_1^k\ \cdots\ p_{m-4}^k\right]^T \in \mathbf{R}^{M-6} \tag{20}$$

since p_k^I and p_k^F in (11) are prescribed. Here, the matrices $B \in \mathbf{R}^{N \times M}$ and $W \in \mathbf{R}^{N \times N}$, and the vector $d_k \in \mathbf{R}^N$ are defined by

$$B = [b_3(s_1)\ b_3(s_2)\ \cdots\ b_3(s_N)]^T, \tag{21}$$
$$W = \mathrm{diag}\{w_1\ w_2\ \cdots\ w_N\}, \tag{22}$$
$$d_k = [d_1^k\ d_2^k\ \cdots d_N^k]^T. \tag{23}$$

Also, the matrix $Q_M \in \mathbf{R}^{M \times M}$ is a Grammian defined as

$$Q_M = \int_{-\infty}^{+\infty} b_3^{(3)}(t) \left(b_3^{(3)}(t)\right)^T dt, \tag{24}$$

and can be computed using B-splines explicitly.

In view of (11), p_k in (16) is written as

$$p_k = \begin{bmatrix} p_k^I \\ \hat{p}_k \\ p_k^F \end{bmatrix}. \tag{25}$$

What we want to do here is to optimize $\bar{J}_k(p_k)$ with respect to $\hat{p}_k$. For this purpose, let Q_M be decomposed as

$$Q_M = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12}^T & Q_{M-6} & Q_{23} \\ Q_{13}^T & Q_{23}^T & Q_{33} \end{bmatrix}, \tag{26}$$

where $Q_{11}, Q_{13}, Q_{33} \in \mathbf{R}^{3\times3}$, $Q_{12} \in \mathbf{R}^{3\times(M-6)}$, $Q_{23} \in \mathbf{R}^{(M-6)\times3}$, and $Q_{M-6} \in \mathbf{R}^{(M-6)\times(M-6)}$ is the principal minor of Q_M of size $M-6$, thus

$$Q_{M-6} > 0. \tag{27}$$

Then straightforward calculation yields

$$p_k^T Q_M p_k = \hat{p}_k^T Q_{M-6} \hat{p}_k + 2(Q_{12}^T p_k^I + Q_{23} p_k^F)^T \hat{p}_k + \text{const.} \tag{28}$$

Next we consider the second term in $\bar{J}_k(p_k)$. In accordance with (26), the matrix B in (21) is also partitioned as

$$B = \left[\, B_I\ \hat{B}\ B_F \,\right], \tag{29}$$

where $B_I, B_F \in \mathbf{R}^{N\times3}$ and $\hat{B} \in \mathbf{R}^{N\times(M-6)}$. We then have

$$\begin{aligned}(Bp_k - d)^T W (Bp_k - d_k) &= \hat{p}_k^T \hat{B}^T W \hat{B} \hat{p}_k \\ &+2\left(B_I p_k^I + B_F p_k^F - d\right)^T W \hat{B} \hat{p}_k + \text{const.}\end{aligned} \tag{30}$$

Thus the cost $\bar{J}_k(p_k)$ can be expressed in terms of $\hat{p}_k$ as $\bar{J}_k(\hat{p}_k)$,

$$\begin{aligned}\bar{J}_k(\hat{p}_k) = \hat{p}_k^T &\left(\lambda Q_{M-6} + \hat{B}^T W \hat{B}\right) \hat{p}_k \\ &+2\left[\lambda\left(Q_{12}^T p_k^I + Q_{23} p_k^F\right)\right. \\ &\left.+\hat{B}^T W \left(B_I p_k^I + B_F p_k^F - d_k\right)\right]^T \hat{p}_k \\ &+\text{const.}\end{aligned} \tag{31}$$

Since $\lambda Q_{M-6} + \hat{B}^T W \hat{B} > 0$, $\bar{J}_k(\hat{p}_k)$ has a unique minimum when

$$\begin{aligned}\hat{p}_k^\star = &\left(\lambda Q_{M-6} + \hat{B}^T W \hat{B}\right)^{-1} \\ &\times \left[\hat{B}^T W (d_k - B_I p_k^I - B_F p_k^F) - \lambda\left(Q_{12}^T p_k^I + Q_{23} p_k^F\right)\right].\end{aligned} \tag{32}$$

Thus, there always exists a unique optimal solution to the Problem 2, and the optimal control $u_k^\star(t)$ and the optimal solution $x_k^\star(t)$ are given by

$$u_k^\star(t) = \sum_{i=-3}^{m-1} p_i^{k\star} \frac{d^3}{dt^3} B_3(t-i), \tag{33}$$

$$x_k^\star(t) = \sum_{i=-3}^{m-1} p_i^{k\star} B_3(t-i). \tag{34}$$

Note here that $p_k^\star = [p_{-3}^{k\star}\ p_{-2}^{k\star}\ \cdots\ p_{m-1}^{k\star}]^T \in \mathbf{R}^M$ is obtained by p_k in (25) with $\hat{p}_k$ substituted by $\hat{p}_k^\star$ in (32). By computing $p_k^\star$ for $\forall k \in [k_L, k_U]$, we then get $\mathcal{M}$ in (8) as a model of hairy-brush dynamic font-based characters.

4. MANIPULATING CHARACTERS

As seen in the previous section, the control polygon $\mathcal{M}_k$ uniquely determines the motion $x_k(t)$, hence the motion on k-th line of hairy-brush ink effect. This fact implies that manipulating characters can be defined as operations on control polygons. Here, we develop such operations, where we restrict ourselves to constituting words from some modeled characters.

Now suppose that some characters are modeled in a default position and in default size in XY-plane, or equivalently, the corresponding control polygons $\mathcal{M}$ in (8) in a default position and in default size in XY-plane.

First, we introduce a translation operation $\mathcal{T}(\cdot;\cdot)$ such that, for given $\mathcal{M}_k$ in (7) and a matrix $A_k \in \mathbf{R}^{3\times M}$ given as

$$A_k = [a^k_{-3}\ a^k_{-2} \cdots\ a^k_{m-1}]$$

with column vectors $a^k_i \in \mathbf{R}^3$ for $i = -3, -2, \cdots, m-1$,

$$\mathcal{T}(\mathcal{M}_k; A_k) = (p^k_{-3} + a^k_{-3})(p^k_{-2} + a^k_{-2}) \cdots (p^k_{m-1} + a^k_{m-1}). \tag{35}$$

Note here that the control polygon $\mathcal{M}_k$ can be translated by the amount A_k to result in a polygon $\mathcal{T}(\mathcal{M}_k; A_k)$. In particular, setting $a^k_{-3} = a^k_{-2} = \cdots = a^k_{m-1} = a$ with a constant vector $a^k \in \mathbf{R}^3$ whose third element is zero, we can simply translate the k-th line of hairy-brush ink, say characters, on the XY-plane. In addition, we note that the third element of vector a^k_i for $i = -3, -2, \cdots, m-1$ can be used to adjust the gradation on the hairy-brush ink.

Remark 4. In (35), setting a^k_i as $a^k_i = \beta(p^k_i - p^0_i)$ for a proper real value $\beta(\geq 0)$, it is noted that the width of characters can be manipulated on the XY-plane.

A natural way to construct a word $\mathcal{M}$ consisting of n characters may be by simply concatenating translated polygons of k-th hairy-brush line, i.e.

$$\mathcal{M}_k = \mathcal{T}(\mathcal{M}^{[1]}_k; A_k)\mathcal{T}(\mathcal{M}^{[2]}_k; A_k) \cdots \mathcal{T}(\mathcal{M}^{[n]}_k; A_k), \tag{36}$$

where $\mathcal{M}^{[i]}_k$ denotes the control polygon on k-th hairy-brush line of i-th character for $i = 1, 2, \cdots, n$. However, the number of control polygon $\mathcal{M}^{[i]}_k$ for each characters, that is the range of k (i.e. k_L and k_U), may be different. Due to that fact, the concatenation as in (36) may be generally unavailable. We therefore adjust the numbers of control polygon $\mathcal{M}^{[i]}_k$ as follows: Let $k^{\min}_L$ and $k^{\max}_U$ be minimum and maximum index of $k^{[i]}_L$ and $k^{[i]}_U$ of characters $\mathcal{M}^{[i]}$ for $i = 1, 2, \cdots, n$, where $k^{[i]}_L$ and $k^{[i]}_U$ denote the lower and upper indices of hairy-brush ink line of i-th character. Then, characters $\mathcal{M}^{[i]}$ is modified as $\bar{\mathcal{M}}^{[i]}$,

$$\bar{\mathcal{M}}^{[i]} = \mathcal{M}^{[i]} \biguplus \mathcal{M}^{[i]}_L \biguplus \mathcal{M}^{[i]}_U \tag{37}$$

with

$$\mathcal{M}^{[i]}_L = \underbrace{\mathcal{M}^{[i]}_{k_L} \biguplus \mathcal{M}^{[i]}_{k_L} \biguplus \cdots \biguplus \mathcal{M}^{[i]}_{k_L}}_{k^{[i]}_L - k^{\min}_L} \tag{38}$$

$$\mathcal{M}^{[i]}_U = \underbrace{\mathcal{M}^{[i]}_{k_U} \biguplus \mathcal{M}^{[i]}_{k_U} \biguplus \cdots \biguplus \mathcal{M}^{[i]}_{k_U}}_{k^{\max}_U - k^{[i]}_U}. \tag{39}$$

Thus, letting $\bar{\mathcal{M}}^{[i]}_k$, $i = 1, 2, \cdots, n$ be control polygon on k-th hairy-brush line of i-th character $\mathcal{M}^{[i]}$, the word $\mathcal{M}$ is constructed as the multiset union of $\mathcal{M}_k$, $k \in [k^{\min}_L, k^{\max}_U]$

$$\mathcal{M}_k = \mathcal{T}(\bar{\mathcal{M}}^{[1]}_k; A_k)\mathcal{T}(\bar{\mathcal{M}}^{[2]}_k; A_k) \cdots \mathcal{T}(\bar{\mathcal{M}}^{[n]}_k; A_k). \tag{40}$$

In the sequel, we will refer such a word $\mathcal{M}$ as 'standard word'.

Remark 5. The conditions in (13) imply that the writing motion of each modeled character (i.e. $x_k(t)$) stops at the beginning and ending of its character. Therefore, in order to get a word with smoother interconnections in (40), we here remove the first and/or last two control points from the connected side of $\bar{\mathcal{M}}^{[i]}_k$, $i = 1, 2, \cdots, n$ and $\forall k \in [k^{\min}_L, k^{\max}_U]$ and concatenate each control polygon as in (40).

In addition, by employing the approximation method in Problem 2, we can reconstruct the words consisting of $\mathcal{M}_k$ in (40) to cursive words with omitted running style as often seen in Japanese calligraphy. Specifically, let $x^c_k(t)$ be a motion of k-th line constituting such cursive hairy-brush words and be given by

$$x^c_k(t) = \sum_{i=-3}^{m-1} \tau^k_i B_3(t - i) \tag{41}$$

with control points $\tau^k_i \in \mathbf{R}$, $i = -3, -2, \cdots, m-1$. Also, the control point vector $\tau_k = [\tau^k_{-3}\ \tau^k_{-2}\ \cdots\ \tau^k_{m-1}]^T \in \mathbf{R}^M$ be partitioned compatibly with p_k in (25) as

$$\tau_k = \begin{bmatrix} \tau^I_k \\ \hat{\tau}_k \\ \tau^F_k \end{bmatrix},\ \tau^I_k = \begin{bmatrix} \tau^k_{-3} \\ \tau^k_{-2} \\ \tau^k_{-1} \end{bmatrix},\ \tau^F_k = \begin{bmatrix} \tau^k_{m-3} \\ \tau^k_{m-2} \\ \tau^k_{m-1} \end{bmatrix}, \tag{42}$$

where $\hat{\tau}_k = [\tau^k_0\ \tau^k_1\ \cdots\ \tau^k_{m-4}]^T \in \mathbf{R}^{M-6}$. In Problem 2, we impose the start and terminal conditions as

$$p^I_k = \tau^I_k,\ p^F_k = \tau^F_k \tag{43}$$

and let the data set $\mathcal{D}_k$ in (5) be given as

$$s_i = i,\ d_i = x_k(s_i),\ i = 1, 2, \cdots, m-2(= N). \tag{44}$$

Note here that p^I_k and p^F_k are first and last three control points of standard word, i.e. $\mathcal{M}_k$ in (40), and are given with a form in (11). The condition (43) implies that the reconstructed motion $x^c_k(t)$ has the same property as the motion of modeled word $x_k(t)$ at the start and terminal points, i.e.

$$x^c_k(t) = x_k(t),\ \dot{x}^c_k(t) = \dot{x}_k(t),\ \ddot{x}^c_k(t) = \ddot{x}_k(t) \text{ for } t = 0, m. \tag{45}$$

Substituting (43) and (44) to (32), we thus get $\hat{\tau}^\star_k = [\tau^{k\star}_0 \cdots \tau^{k\star}_{m-4}]^T \in \mathbf{R}^{M-6}$. Therefore, letting $S(\cdot; \lambda; W)$ be an operator for reconstructing standard words to cursive words with the parameters λ and w_i, $i = 1, 2, \cdots, N$, then we have

$$S(\mathcal{M}_k; \lambda; W) = p^k_{-3}p^k_{-2}p^k_{-1}\tau^{k\star}_0 \cdots \tau^{k\star}_{m-4}p^k_{m-3}p^k_{m-2}p^k_{m-1}. \tag{46}$$

5. A DESIGN EXAMPLE

We here show an example of modeled characters and cursive words produced by using our proposed method in Sections 3 and 4.

Figure 5 shows the results for the modeled characters 'i' and 'shi'. Here, (a) and (b) are the 'pre-designed' and 'transformed' characters that are obtained by the dynamic font method and Pix2Pix in Section 2. Also, (c) is the corresponding modeled characters. We here set a and $[k_L, k_U]$ as $a = 1$ and $[k_L, k_U] = [-24, 18]$ for 'i' and $[k_L, k_U] = [-15, 13]$ for 'shi' respectively. Also, m is set as $m = 15$ for 'i' and $m = 6$ for 'shi', respectively.

The method described in Section 4 is then applied to the word 'i-shi' constructed by concatenating 'i' and 'shi' in Fig. 5 (c). The results are shown in Fig. 6, where the parameter λ is set as (a) $\lambda = 0.1$, (b) $\lambda = 1$ and (c) $\lambda = 5$

(a) pre-designed (b) Pix2Pix (c) modeled

Fig. 5. (a) pre-designed characters, (b) characters transformed by Pix2Pix and (c) modeled dynamic font-based characters.

(a) $\lambda = 0.1$ (b) $\lambda = 1$ (c) $\lambda = 5$

Fig. 6. Reconstructed words with (a) $\lambda = 0.1$, (b) $\lambda = 1$ and (c) $\lambda = 5$.

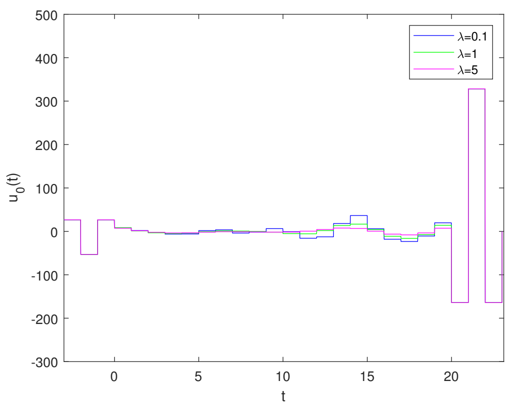

Fig. 7. Optimal control input $u(t)$ for $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 5$.

all for $W = I$. We may observe that hairy-brush characters (i.e. Fig. 5 (c)) and cursive words (i.e. Fig. 6 (a)-(c)) as seen in real Japanese calligraphy have been produced. As for the generation of cursive words, we see that the smoothness of not only respective characters but also their connection increases as λ increases. This indicates that the degree on omitted running style of reconstructed cursive words can be arbitrarily adjusted by λ. In addition, the optimal control input $u_k^\star(t)$ for $k = 0$ is plotted in Fig. 7. Then, we obviously see that the control input becomes smaller as λ increases.

6. CONCLUDING REMARKS

We considered a problem of modeling and manipulating hairy-brush characters based on the dynamic font method. We, in particular, developed a scheme for modeling hairy-brush characters by formulating the problem as an optimal function-approximation problem. Then, a central issue was to determine control points for generating optimal input and solution. In addition, we derived some operations in order to generate cursive words in a systematic way, and it has been applied successfully as shown in a design example. We here considered only the case of Japanese Kana characters, but the scheme can be used in other languages as well.

REFERENCES

Aoyanagi K., Kouzan Hairy-Brush Font, https://opentype.jp/kouzanmouhitufont.htm.

Egerstedt M. and C. Martin, *Control Theoretic Splines: Optimal Control, Statistics, and Path Planning*, Princeton University Press, 2010.

Fujioka H. and H. Kano et al, Constructing and Reconstructing Characters, Words and Sentences by Synthesizing Writing Motions, *IEEE Trans. Systems, Man and Cybernetics, Part A*, Vol.36, No.4, pp.661-670, 2006.

Gill R. J., D. Kulić and C. Nielsen, Spline Path Following for Redundant Mechanical Systems, *IEEE Trans. Robotics*, Vol.31, Issue 6, pp.1378 - 1392, 2015.

Gonzalez R. C., R. E. Woods and S. L. Eddins, *Digital Image Processing Using MATLAB*, Pearson Education Inc., 2004.

Isola P., J-Y. Zhu, T. Zhou, A. A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp.5967-5976, Honolulu, HI, USA, July 21-26, 2017.

Kano H., M. Egerstedt, H. Nakata, and C.F. Martin, B-Splines and Control Theory, *Applied Mathematics and Computation*, Vol.145, No.2-3, pp.265-288, 2003.

Lyu P., X. Bai, C. Yao, Z. Zhu, T. Huang and W. Liu, Auto-Encoder Guided GAN for Chinese Calligraphy Synthesis, *arXiv.org*, Computer Science, arXiv:1706.08789v1, June, 2017.

Mi X.-F., M. Tang and J-X Dong, Droplet: A Virtual Brush Model to Simulate Chinese Calligraphy and Painting, *J. Comput. Sci. and Technol.*, Vol.19, No.3, pp.393-404, 2004.

Ronneberger O., P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, *arXiv.org*, Computer Vision and Pattern Recognition, arXiv:1505.04597, May, 2015.

Sun W., W. J. Niessen and S. Klein, Randomly Perturbed B-Splines for Nonrigid Image Registration, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.39, Issue 7, pp.1401 - 1413, 2017.

Takayama K. and H. Kano et al, Dynamic Font: A New Representation Technology, *FUJITSU Sci. and Tech. J.*, Vol.32, No.2, pp.192-202, 1996.

Uehara T., Current Technology and Problems in Computer Font (in Japanese), *Trans. Information Processing Society of Japan*, Vol.31, No.11, pp.1570-1580, 1990.

Wong H. T.F. and H.S. I. Horace, Virtual brush: a model-based synthesis of Chinese calligraphy, *Computers & Graphics*, Vol.24, Issue 1, pp.99-113, 2000.

Xu S., M. Tang, F. C.M. Lau, A Solid Model Based Virtual Hairy Brush, *HKU CSIS Technical Report*, TR-2002-04, 2002.