Original Paper *Forma*, **16**, 155–171, 2001

Simulation on the Formation Process of Japanese *Hiragana* Characters by Deformation and Morphing

Huijun SUN[1], Lifeng HE[2], Tsuyoshi NAKAMURA[1] and Hidenori ITOH[1]

[1]*Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan*
[2]*Aichi Prefectural University, Nagokute-cho, Aichi 480-1198, Japan*

(Received August 9, 2001; Accepted October 5, 2001)

Keywords: Origin of Japanese *Hiragana* Characters, Character Deformation, Morphing, B-Spline Curve, Energy Minimization Principle

Abstract. This paper introduces a method for simulating the origin of Japanese *Hiragana* characters from their corresponding Chinese characters. By demolishing the Chinese characters, i.e., connecting some of their strokes with straight lines and deforming them, the square style Chinese characters are first transformed into their cursive writing style ones. Then, B-spline morphing is used to simulate the evolution process from the generated cursive writing style Chinese characters to the corresponding Japanese *Hiragana* characters, where the energy minimization principle is used to find the optimal *smooth coefficient* which makes the total morphing energy minimal. The experimental results show such a method as a reasonable way to simulate the origin of Japanese *Hiragana* characters.

1. Introduction

It is well-known that the shapes of Japanese *Hiragana* characters originated from the corresponding Chinese characters (KABA, 1979). Some examples are shown in Fig. 1.

To comprehend the origin of Japanese *Hiragana* characters is very helpful for the beginners to study Japanese. In fact, such originating process are introduced in the Japanese lessons of Japanese primary schools. However, as we know from Fig. 1, there exist big gaps between the Chinese characters and their corresponding Japanese *Hiragana* characters. It is difficult to directly image the evolution process from the Chinese characters to their corresponding Japanese *Hiragana* characters. Therefore, the successive simulation of the evolution process from a Chinese character to its corresponding Japanese *Hiragana* character is expected. However, there is almost no report on this topic.

This paper considers a method for automatically simulating the origin of Japanese *Hiragana* characters from their corresponding Chinese characters. After users input a pair of a Chinese character and its corresponding Japanese *Hiragana* character, a sequence of characters that simulates the evolution process from the Chinese character to its corresponding Japanese *Hiragana* character is automatically generated. This simulation

安	以	宇	衣	於	加	幾	久	計	己
あ	い	う	え	お	か	き	く	け	こ

Fig. 1. The bottom is some of Japanese *Hiragana* characters and the above is the corresponding Chinese Characters.

process was designed to automatically with a minimum of manual intervention, and the only manual intervention we would like to require is the input of sample Chinese characters and Japanese character.

The simulation consists of two steps. In the first step, the square style Chinese characters are deformed to their cursive writing style ones. In the second step, a selected cursive writing style character is used to generate the intermediate characters between the cursive writing style character and the Japanese *Hiragana* character by B-spline morphing.

In order to deform a square style Chinese character to its cursive writing style ones, according to the geometric properties of the Chinese character, some of its strokes are first connected with straight lines to form quasi-cursive style one. Then, by a transformation matrix, which is defined by a *smooth coefficient*, that can be selected in a certain range, the corresponding cursive writing style character is derived (TAKEUCHI, 1998). The bigger a smooth coefficient is, the smoother a derived cursive writing style character is.

Although the derived cursive writing style characters appear, more or less, similar to the corresponding Japanese *Hiragana* character, they are still different to that Japanese *Hiragana* character. Therefore, we use morphing to simulate the evolution process from a selected cursive writing style Chinese character to its corresponding Japanese *Hiragana* character. Our morphing problem can be formulated as follows. Given two sample characters C_0, C_1 and a control parameter $t \in [0,1]$, construct an intermediate character C_t, which is more similar to C_0 as $t \to 0$ and more similar to C_1, as $t \to 1$.

Moreover, with different smooth coefficients, from a square style Chinese character, we can derive different cursive writing style characters. The necessary energies for morphing these cursive writing style characters to the corresponding Japanese *Hiragana* character are also different. Obviously, the smaller the energy is, the better a result would be. Therefore, we use the energy minimization principle to select the most suitable smooth coefficient that makes the morphing energy minimal. In this paper, the “Chinese character” and “*hiragana* character” are defined as the inputed handwriting square style Chinese characters and the Japanese *Hiragana* characters, respectively.

The rest of this paper is constructed as follows. In the next section, we introduce how to generate the cursive writing style Chinese characters from their square style ones by matrix transformation. Section 3 we considers the morphing problem between a cursive writing style Chinese character and its corresponding Japanese *Hiragana* character. How to derive the optimal smooth coefficient by energy minimization principle is introduced in Section 4. Then, we describe the system and show some experimental results in Section 5. Finally, conclusions and future works are summarized in Section 6.

2. A Deformation Method for Generating Cursive Writing Style Chinese Characters

In this paper, we assume that the width of the strokes of input characters be one pixel.

In this way, each stroke is a serial of successive segments from an end point to another end point, and the connectivity of each pixel in a stroke except the end points is 2-neighbor.

In our system, cursive writing style Chinese characters are generated from the corresponding square style Chinese ones by deformation. Since a square style Chinese character consists of strokes, which are considered as curves in our system, a deformation of a character can be reduced into a deformation of its strokes. Moreover, since a curve (stroke) can be approximated to a zigzag curve, i.e., a successive segment, where each of the vertex points in the zigzag curve is one of the discrete points from the original curve, a deformation of a stroke can be considered as a deformation of the corresponding zigzag curve. In this way, if we can define a transformation function for all zigzag curves derived from a character, we can obtain a deformed character.

2.1. Modeling curves

Suppose $(c_0, ..., c_k, ..., c_n)$ is the order set of the vertex points in a zigzag curve, where for each c_k $(0 \le k \le n)$, $c_k = a_k + j \cdot b_k$. The segment from point c_{k-1} to point c_k, denoted by w_k, can be defined as

$$w_k = c_k - c_{k-1}.$$

In this way, the zigzag curve can be modeled as

$$\{w_1, w_2, ..., w_n\}.$$

Notice that each segment w_i $(1 \le i \le n)$ is considered as a vector.

2.2. Transformation matrix for deforming curves

Suppose that X be the original curve expressed by the segment set $\{w_1, w_2, ..., w_n\}$ and C a deforming transformation matrix. Then, the transformed curve Y, denoted as $\{w_1', w_2', ..., w_n'\}$, can be derived as follows.

$$Y^T = CX^T$$

where X^T and Y^T are the transposition matrix of X and Y, respectively.

TAKEUCHI (1998) introduced a method for defining a transformation matrix to deform a zigzag curve into smooth ones, where a deformation parameter, called *smooth coefficient**, is used to control the smooth degree of the transformed curve. For smooth coefficient m, the corresponding deforming transformation matrix C_m $(n \times n)$ is defined as follows.

---

*It is also called *Ren-hitsu degree* in Japanese.

$$C_m = \begin{bmatrix} 1 & \overbrace{1\ 1\ \cdots\ 1}^{m-1} & 0\ 0\ \cdots\ 0\ 0\ \cdots\ \cdots\ 0\ 0\ \cdots\ 0\ 0\ 0 \\ 1\ 1 & \overbrace{1\ \cdots\ 1\ 1}^{m-1} & 0\ \cdots\ 0\ 0\ \cdots\ \cdots\ 0\ 0\ \cdots\ 0\ 0\ 0 \\ & \ddots & \vdots \\ \overbrace{1\ \cdots\ 1\ 1}^{m-1} & 1\ \overbrace{1\ 1\ \cdots\ 1}^{m-1} & 0\ \cdots\ \cdots\ 0\ 0\ \cdots\ 0\ 0\ 0 \\ 0\ \overbrace{1\ 1\ \cdots\ 1}^{m-1} & 1\ \overbrace{1\ 1\ \cdots\ 1}^{m-1} & 0\ \cdots\ 0\ 0\ \cdots\ 0\ 0\ 0 \\ & \ddots & \vdots \\ 0\ \cdots\ \cdots & \cdots\ \cdots\ 0 & \overbrace{1\ \cdots\ 1\ 1}^{m-1}\ 1\ 1 \\ 0\ \cdots\ \cdots & \cdots\ \cdots\ 0 & 0\ \overbrace{1\ \cdots\ 1\ 1}^{m-1}\ 1 \end{bmatrix} \tag{1}$$

That is, if the original curve is $\{w_1, \ldots, w_k, \ldots, w_n\}$ and the smooth coefficient is m, then each segment of the transformed curve $\{w_1', \ldots, w_k', \ldots, w_n'\}$, say, w_k' ($1 \le k \le n$), is calculated by the sum of m–1 number segments before and after w_k, together with w_k itself. In the case where there are not enough segments before (after) w_k, all existing segments before (after) w_k are used to derive w_k'.

An example is shown as in Fig. 2, where the original curve is $\{w_1, w_2, w_3, w_4\}$ (Fig. 2(a)). If the smooth coefficient is selected as 2, the transformation matrix can be defined as:

$$C_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2}$$

The transformed curve is shown in Fig. 2(b).

2.3. *Deriving cursive writing style Chinese characters by transformation matrix*

Usually, a square style Chinese character consists of more than one stroke. Each stroke of a square style Chinese character can be considered as a continuous curve. To obtain a cursive style Chinese character from the square style one, according to the geometric properties of the square style character, we first connect some strokes with straight lines that interpolate from an end point of one stroke to the first point of another stroke to form a quasi-cursive style Chinese character.

For example, suppose that two strokes of the considered square style Chinese character be $\{u_1, \ldots, u_a\}$ and $\{v_1, \ldots, v_b\}$, respectively, and the straight line connecting u_a

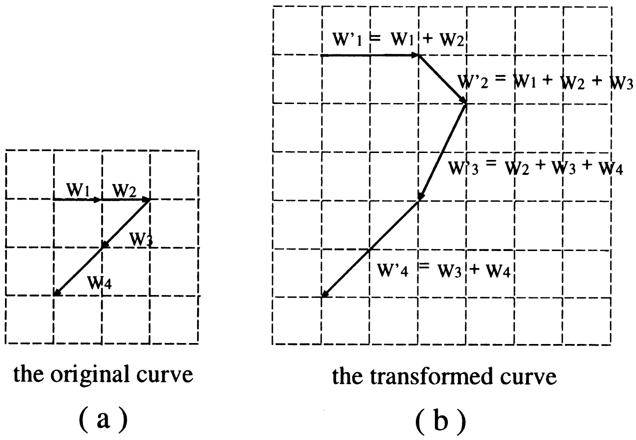

Fig. 2. Deforming a curve by a transformation matrix.

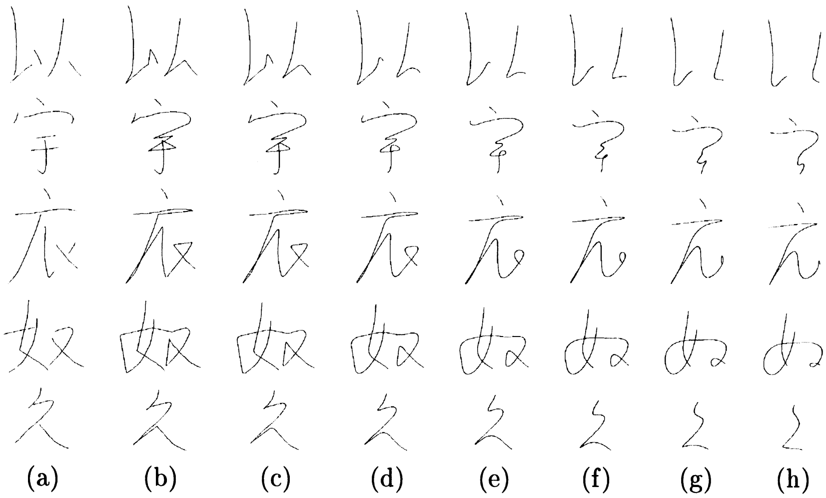

Fig. 3. Examples of Chinese character deformation from square style ones to cursive style ones, where (a) are the input square style characters; (b) the quasi-cursive style ones; (c), (d), (e), (f), (g) and (h) the generated cursive style ones by setting the *smooth coefficient* as 10, 20, 30, 40, 50 and 60, respectively.

and v_1 be $\{w_1, ..., w_c\}$, then, the new curve formed by the two curves is described as $\{u_1, ..., u_a, w_1, ..., w_c, v_1, ..., v_b\}$.

With a quasi-cursive style Chinese character A and a selected smooth coefficient m, we can use the transformation matrix C_m to deform all the curves contained in A to obtain the corresponding cursive writing style character. Some examples are shown in Fig. 3.

3. Generating Japanese *Hiragana* Characters from Their Corresponding Cursive Writing Style Chinese Characters

In our system, generating Japanese *Hiragana* characters from their corresponding cursive writing style Chinese characters is considered as a morphing problem between two sample characters, which can be further reduced to the morphing of the corresponding strokes of the two sample characters. Therefore, we will first consider how to generate versatile intermediate strokes from two sample strokes.

Morphing (metamorphosis) is used for deal with that, when given two objects, one object continuously deforms to the other. Morphing of objects are popular in animation seen in the entertainment and the broadcasting industry. Many algorithms have been proposed for the morphing of images (BEIER and NEELY, 1992), polygons (SEDERBERG and GREENWOOD, 1992; SEDERBERG *et al.*,1993; SHAPIRO and RPPOPORT, 1995), polyhedra (BETHEL and UELTON, 1989) and volume data (HUGHES, 1992; HE and KAUFMAN, 1994).

Our algorithm for morphing object is simpler than the existing morphing schemes, and the results seem quite favorable. The key for the success of our algorithm is the B-spline shape representation, which captures the geometric properties of a shape detailedly, a crucial requirement for aesthetic shape deformations.

3.1. Stroke representation

In order to generate intermediate morphing strokes from two sample strokes, a stroke is described by its corresponding B-spline curve. We first review the definition of B-spline curve and then introduce how to derive the B-spline curve of a stroke.

It is well-known that a B-spline curve with degree p can be defined by a set of control points $\{c_0(x_0, y_0), c_1(x_1, y_1), ..., c_n(x_n, y_n)\}$ and a knot vector $U = \{u_0, u_1, ..., u_m\}$ in the following form (GORDON and RIESENFELD, 1974):

$$x(u) = \sum_{i=0}^{n} N_{i,p}(u)x_i \tag{3}$$

$$y(u) = \sum_{i=0}^{n} N_{i,p}(u)y_i \tag{4}$$

where $u_0 \le u_1 \le u_2 \le \cdots \le u_m$ and the half-open interval $[u_i, u_{i+1})$ is called i-th knot span. Moreover, u is a variable in the range $u \in [u_i, u_{i+p+1}]$ $(i = 0, 1, ..., n)$, $(x(u), y(u))$ is a point in B-spline curve relative to u, and $N_{i,p}(u)$ is a B-spline basis function of degree p. Notice that n, m and p must satisfy the relation of $m = n + p + 1$.

The knots can be considered as division points that subdivide the interval $[u_0, u_m]$ into knot spans. All B-spline basis functions are supposed to have their domain on $[u_0, u_m]$. For convenience, we usually set $u_0 = 0$ and $u_m = 1$ so that the domain becomes a closed interval [0,1].

The i-th B-spline basis function of degree p, $N_{i,p}(u)$, is defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{6}$$

To compute $N_{i,p}(u)$ for degree p greater than 0, we use the triangular computation scheme, where all basis functions of degree 0 are on the first column. Then, from $N_{i,0}(u)$ and $N_{i+1,0}(u)$, according to the formula (6), $N_{i,1}(u)$ can be calculated. That is, we can compute $N_{0,1}(u)$, $N_{1,1}(u)$, and so on. All of those $N_{i,1}(u)$ are written on the second column. Once all $N_{i,1}(u)$ have been computed, we can use them to compute the third column. This process will continue until all required $N_{i,p}(u)$ are derived. An example is shown in the following diagram, where $p = 5$. Usually, such diagram is referred to as the Cox-de Boor recursion formula (DEBOOR, 1972).

$$\begin{array}{cccccc}
N_{0,0} & & & & & \\
 & N_{0,1} & & & & \\
N_{1,0} & & N_{0,2} & & & \\
 & N_{1,1} & & N_{0,3} & & \\
N_{2,0} & & N_{1,2} & & N_{0,4} & \\
 & N_{2,1} & & N_{1,3} & & N_{0,5} \\
N_{3,0} & & N_{2,2} & & N_{1,4} & \\
 & N_{3,1} & & N_{2,3} & \cdots & \\
N_{4,0} & & N_{3,2} & \cdots & & \\
 & N_{4,1} & \cdots & & & \\
N_{5,0} & \cdots & & & & \\
\cdots & & & & &
\end{array} \tag{7}$$

With B-spline curves, we can use lower degree curves to maintain a large number of control points. By B-spline curve's local modification property, the change of a control point only influences the shape of the corresponding part in a curve (GORDON and RIESENFELD, 1974). Moreover, since B-spline curves satisfy the strong convex hull property, they are suitable for shape control. Therefore, we can design and edit a curve through adjusting control points.

In our system, we first take the sample points by equal-distance sampling from an input stroke, then, use these points, i.e., $(x(u), y(u))$ in formulas (3) and (4), we can derive the control points, i.e., (x_i, y_i) in formulas (3) and (4), for the stroke. Moreover, we take degree p as 3. An example is shown in Fig. 4, where the control points form a control polygon.

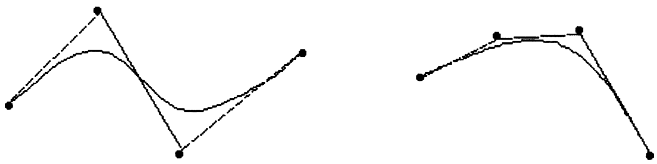

Fig. 4. Deriving B-spline curve control points and control polygon for strokes.

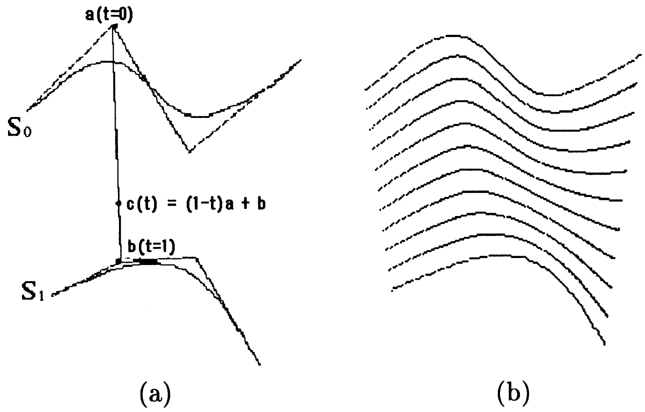

Fig. 5. Intermediate stroke generation.

3.2. Generating intermediate strokes

Now we consider how to generate intermediate morphing strokes from two sample strokes, that have been represented by B-spline curves with control points. Notice that the number of control points of the two sample strokes are made as same.

In principle, an intermediate B-spline curve (stroke) can be derived by affine transformation from every corresponding point (pixel) in the two input strokes. However, there are two problems. One is that there are too many input points in a stroke to process. Another is the numbers of pixels in two input strokes may be different, therefore, the process cannot be made easily.

In our system, we only make affine transformation for those derived corresponding control points of the two strokes to derive the control points for an intermediate stroke. Then, we use the derived control points to construct an intermediate stroke. Obviously, each derived intermediate B-spline curve still holds the affine invariance property (MOEBIUS, 1885).

Our morphing problem can be formulated as follows. Given two sample stokes S_0, S_1 and a control parameter $t \in [0,1]$, construct an intermediate morphing stroke S_t, with the increase of t from 0 to 1, S_t varies from S_0 to S_1.

Suppose that a and b are a couple of the corresponding control points in two corresponding sample strokes, say, S_0 and S_1, respectively, as shown in Fig. 5(a). We

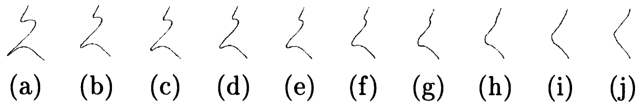

Fig. 6. Morphing results for two very different curves: (a) S_0 (the corresponding cursive writing style Chinese character "Jiu"); (b), (c), (d), (e), (f), (g), (h) and (i) are the generated intermediate S_t by setting the morphing parameter t to 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, respectively; (j) S_1 (Japanese *Hiragana* character "Ku").

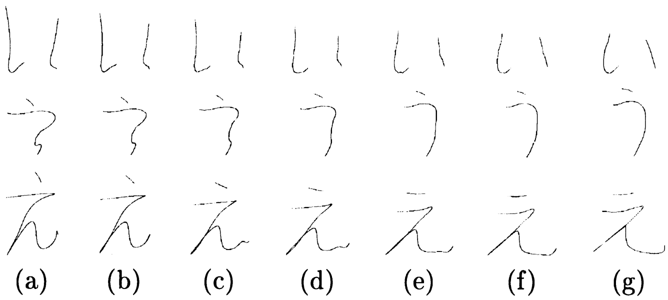

Fig. 7. Morphing results for characters: (a) sample characters (the corresponding cursive writing style Chinese characters); (b), (c), (d), (e), and (f) are the generated intermediate characters by setting the morphing parameter t to 0.2, 0.4, 0.6, 0.8, 0.9, respectively; (j) sample characters (the Japanese *Hiragana* character "i", "u" and "e").

interpolate a segment from a to b, therefore a point $c(t)$ in the derived segment can be described as $c(t) = (1 - t)a + t \cdot b$, where $t \in [0,1]$. In this way, for each t, we can derive all $c(t)$ for all control point couples in strokes S_0 and S_1. Then, take all those $c(t)$ as control points, according to formulas (3) and (4), we can interpolate them to form an intermediate stroke S_t of S_0 and S_1 relative to the parameter t. When t increase from 0 to 1, S_t has a resembling tendency from S_0 to S_1. Some example intermediate strokes generated from S_0 and S_1 given in Fig. 5(a) are shown in Fig. 5(b).

Our system is robust for our purpose, even the input two curves look very different, our system can give a reasonable results. Figure 6 shows such an example.

3.3. Generating intermediate characters

For a control parameter $t \in [0,1]$, from a cursive style Chinese character C_0 and a corresponding Japanese *Hiragana* character C_1, processing all corresponding strokes respectively, we can derive an intermediate character C_t, with the increase of t from 0 to 1, C_t varies from C_0 to C_1.

Figure 7 shows some morphing experimental results for some cursive style Chinese characters and corresponding Japanese *Hiragana* characters.

Since we use B-spline function of degree 3 as the base of B-spline curves, the generated strokes are much smoother than those generated by other methods (e.g., Lagrange polynomial interpolation) (MICHAEL, 1999). According to the lesser tendency to oscillate (minimum curvature property) property (least-squares) of B-spline function of degree 3, each intermediate stroke is more preferable to other forms of representations.

4. Finding the Optimal Smooth Coefficient by Energy Minimization Principle

As introduced above, to deform the square style Chinese characters to their cursive writing style characters, a transformation matrix, defined by a transformation parameter, i.e., *smooth coefficient* is used. With different smooth coefficients, we can derive different cursive writing style characters.

In order to achieve the best result, we should select the optimal one from the large number of generated cursive writing style character candidates to make morphing. In other words, we should decide the optimal smooth coefficient to generate the optimal cursive writing style character.

In our system, we use the energy minimization principle (KASS *et al.*, 1987) to decide the optimal smooth coefficient. We consider that the optimal smooth coefficient is the one that makes the total energy morphing from the cursive writing style character generated by using it to the corresponding Japanese *Hiragana* character minimal.

KASS *et al.*(1987) first established a well-known *snake* model to process vision tasks, and LAI and CHIN (1995) gave a more innovative model, the *g-snake* model. The common feature of both models is that, for each model, there is a defined internal energy to keep the stroke continuity, i.e., to keep the initial shape of a stroke, and an imaginary external force acted on the stroke to make it deform. The stroke is attracted by such actions to deform to a desirable final state, where the external energy, i.e., that make the origin state deforming to the desirable state is the absolute difference value of two internal energies of the two states.

Due to the different usage, the external force can, for example, comes from a user interface, automatic attentional mechanisms or high level interpretations etc.

In our case, we consider that the energy for character transformation consist of difference value of internal energies (the origin state and the desirable state) and morphing energy. We imagine that there is a force acted on a selected cursive writing style character C_0 to make it deform to the corresponding Japanese *Hiragana* character C_1.

For a problem, if its solution relative to the initial conditions can be determined directly, it is called a well-posed problem. Inversely, if there are too many solutions, i.e., the solution can not be determined directly, it is ill-posed. For example, one equation with two variables is a typical ill-posed problem. Our problem is also an ill-posed ones, because we have a solutions with each smooth coefficient m.

Regularization technique is a basic method to transform an ill-posed problem to a well-posed one. Here we use a simple model, the solution of linear equations, as our example to explain its basis concepts. This problem can be described as: for the given vector $\boldsymbol{x}$, to find the vector $\boldsymbol{y}$ to meet the following:

$$A\boldsymbol{y} = \boldsymbol{x} \tag{8}$$

where, A is a linear operator that contains coefficients in each equation.

When the inverse matrix A^{-1} of A does not exist, $\boldsymbol{y}$ cannot be directly determined for a given $\boldsymbol{x}$, the problem is ill-posed. In such situation, a reasonable way is introducing some constraints on $\boldsymbol{y}$ to make it to be able to directly determined. For example, finding $\boldsymbol{y}$ such that for any linear transformation B, $\boldsymbol{y}$ makes the normal second power B, i.e., $||B\boldsymbol{y}||^2$, minimum. The problem is formulated as find $\boldsymbol{x}$ in expression (9).

$$\min_y ||A\boldsymbol{y} - \boldsymbol{x}||^2 + \lambda||B\boldsymbol{y}||^2 \tag{9}$$

where, λ is Lagrangian multiplier. By such assumption, the solution can be directly determined, and then the problem is changed to be well-posed. Here, λ is also called regularization parameter.

Regularization has found a lot of applications in recognition aspect, such as edge detection, optimal flow detection, surface reconstruction, etc. In fact, the regularization technique is a problem to find a minimum sum energy. The details should be referred to POGGIO *et al.*(1985).

The *g-snake* model (LAI and CHIN, 1995) is an application of the minimum energy principle on a curve. In *g-snake* model, a curve $\boldsymbol{U}$ is represented as a link of point vector constructing it, i.e., $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n]$, where, for each i such that $1 \le i \le n$, $u_i \in E = \{(x,y)\colon x, y = 1, 2, \ldots, M\}$, thus $\boldsymbol{U} \in E^n$. Each element in $\boldsymbol{U}$ is also called *snaxel* (snake-pixel). According to the vector combination principle, each snaxel can be expressed as a linear combination of its two adjacent snaxel vectors

$$u_i = \alpha_i u_{i_\alpha} + \beta_i u_{i_\beta} \tag{10}$$

where the basis indices are given by:

$$i_\alpha = \begin{cases} i-1; & i > 1 \\ 3; & i = 1, \end{cases} \quad i_\beta = \begin{cases} i+1; & i < n \\ n-2; & i = n. \end{cases} \tag{11}$$

$\boldsymbol{U}$ is *nontrivial* if $\left| x_{i_\alpha} y_{i_\beta} - x_{i_\beta} y_{i_\alpha} \right| > 0$ for $1 \le i \le n$. The non-trivial condition ensures that u_{i_α} and u_{i_β} are linearly independent for all i.

Collecting and rearranging similar equations for all i, the shape equation of the stroke of an object can be written down as:

$$\boldsymbol{A}\boldsymbol{U}^T = 0 \tag{12}$$

where $\boldsymbol{A}$ is called shape matrix that contains the necessary information to describe the shape:

$$A = \begin{bmatrix} 1 & -\beta_1 & -\alpha_1 & 0 & \cdots & 0 \\ -\alpha_2 & 1 & -\beta_2 & 0 & 0 & \cdots \\ 0 & -\alpha_3 & 1 & -\beta_3 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -\alpha_{n-1} & 1 & -\beta_{n-1} \\ 0 & 0 & \cdots & -\beta_n & -\alpha_n & 1 \end{bmatrix}. \tag{13}$$

An internal energy E_{int} induced by shape matrix A is defined to represent such fluctuation:

$$E_{\text{int}}(\boldsymbol{U}) = \frac{\left(\boldsymbol{AU}^T\right)^T R^{-1}\left(\boldsymbol{AU}^T\right)^T}{l(\boldsymbol{U})} \tag{14}$$

where

$$l(\boldsymbol{U}) = \frac{1}{n}\sum_{i=1}^{n}\left\|u_{i+1} - u_i\right\|^2 \tag{15}$$

is a normalizing constant, and R = diag$\{\sigma_1^2, \sigma_2^2, ..., \sigma_n^2\}$ contains the deformation variances σ_i^2 that allows assignment of location dependent weightings on deformations.

Rewriting and denoting the *internal* energy as follow (LAI and CHIN, 1995):

$$E_{\text{int}}(u_i) = \frac{\left\|u_i - \alpha_i u_{i_\alpha} - \beta_i u_{i_\beta}\right\|^2}{l(\boldsymbol{U})}. \tag{16}$$

The external energy for deforming a cursive writing style character to the corresponding Japanese *Hiragana* character, i.e., the morphing energy, can be described as:

$$E_{\text{ext}} = \left\|E_{\text{int}_m} - E_{\text{int}_g}\right\| \tag{17}$$

where E_{int_m} is the internal energy of the cursive writing style character relative to the smooth coefficient m, and E_{int_g} is the internal energy of the corresponding Japanese *Hiragana* character.

Since E_{int_m} is a function of the smooth coefficient m, the morphing energy E_{ext} also depends on m. Obviously, the optimal smooth coefficient is the one that makes E_{ext}

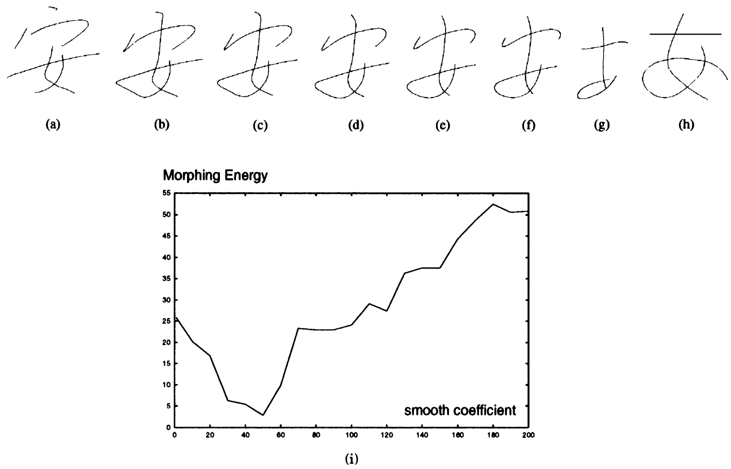

Fig. 8. Deriving the optimal smooth coefficient by energy minimization principle.

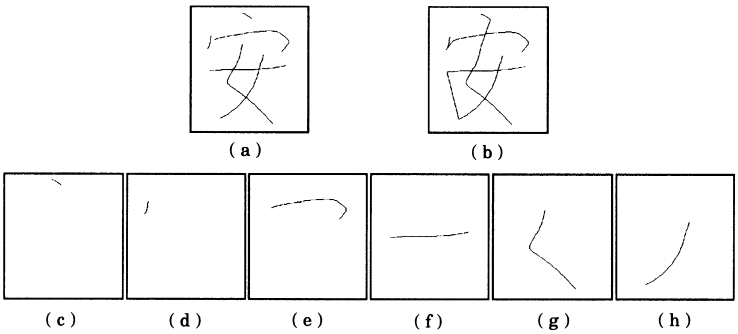

Fig. 9. An example for forming the quasi-cursive character, (a) the input square style Chinese character; (b) the corresponding quasi-cursive style one; (c), (d), (e), (f), (g) and (h) are the first, second, third, fourth, fifth and the sixth stroke of the Chinese character, respectively.

minimal. Therefore, we can calculate the optimal smooth coefficient by analyzing E_{ext} on m.

An example is shown in Fig. 8. Figure 8(a) is a Chinese character “an” and Fig. 8(h) is a corresponding Japanese character “a”. Figures 8(b), (c), (d), (e), (f) and (g) are the generated cursive writing style characters by setting the smooth coefficient m to 10, 20, 30, 40, 50, 100, respectively. Figure 8(i) shows the relationship between smooth coefficient m and morphing energy for transforming the cursive writing style character of “an” relative to m to the corresponding Japanese character “a”. From the chart we can find that when the smooth coefficient m is around 50, the morphing energy is minimal. Therefore, 50 is the optimal smooth coefficient for deriving the cursive style character to be used for morphing from the Chinese character “an” shown in Fig. 8(a).

5. System Description and Experimental Results

Our system has been implemented on a *Sun-microsystems* **ULTRA5** workstation. The experimental results show that our system is reasonable and robust.

The processing flow of our system to simulate the origin of a Japanese *Hiragana* character can be summarized as follows:

(1) Using an electronic pen, input a pair of a square style Chinese by character and a corresponding Japanese *Hiragana* character by the same person in respectively. The input characters are regularized as 256×256 pixel size, and each character is saved by strokes which are arranged by the input order.

(2) From the input square style Chinese character, constructing the quasi-cursive style character. According to the quasi-cursive style character generation rule, which is stored in a database established in advance, we connect some stroke with straight lines. For example, Fig. 9(a) is a square style Chinese character "an", (c), (d), (e), (f), (g) and (h) are its strokes in the input order. To form the quasi-cursive style one, shown in Fig. 9(b), the generation rule is: using straight lines to connect the end points of the first stroke, the second stroke and the sixth stroke to the starting points of the fifth stroke, the third strokes and the fourth stroke, respectively.

(3) Taking the smooth coefficient from 10 to 100, for every additional 10, we use it to generate a cursive writing style character of the square style Chinese character and calculate the needed morphing energy from the cursive writing style character to the input Japanese *Hiragana* character to derive the optimal smooth coefficient.

(4) Taking the cursive writing style character generated with the derived optimal smooth coefficient as C_0, the input Japanese *Hiragana* character as C_1, executing the morphing algorithm to derive the intermediate characters by setting the morphing parameter t to 0.2, 0.6 and 0.9, respectively.

(5) Outputting the results. The sequence of characters that simulates the evolution process from the input square style Chinese character to the input Japanese *Hiragana* character is output in the following order: the input square style Chinese character, the quasi-cursive Chinese character, then, the cursive Chinese characters, which are generated with the smooth coefficient from 10 to the optimal one for every additional 10, then, all intermediate characters generated by morphing, lastly, the Japanese *Hiragana* character.

Some experimental output results are shown in Fig. 10. The first row is the input Chinese characters; the 12th row is the corresponding input Japanese "Hiragana" characters; the 2nd row is the corresponding quasi-cursive character; from the 3rd row to the 8th row are the cursive character relative the different smooth coefficient 10 to the optimal one for every additional 10; from the 9th row to the 11th row is the intermediate characters generated by morphing from the optimal smooth coefficient, the morphing parameter is set to 0.2, 0.6 and 0.9. In this example the (a) column have the optimal smooth coefficient 50; the (b), (c), (d) columns have the optimal smooth coefficient 60; the column (e) have the 40 optimal smooth coefficient.

Fig. 10. Experimental output results for transformation: (a), (b), (c), (d) and (e) for Japanese *Hiragana* characters “a”, “i”, “u”, “e” and “o”.

6. Conclusions

In this paper, we proposed a method to simulate the origin of Japanese *Hiragana* characters from their corresponding square style Chinese characters. By character deformation, from square style Chinese characters, we can obtain the cursive writing style ones. Then, by character morphing, we can simulate the evolution process from the cursive writing style Chinese characters to their corresponding Japanese *Hiragana* characters.

In our system, we definded the deformation energy, and focus the relationship between deformation energy and the form of the goal character, from our experiment we have the result, that is the smaller the deformation energy, the smooth coefficient that will makes the generated characters more natural.

Since B-spline curve of degree 3 is applied for solving the morphing problem, it makes the character morphing more flexible. As a result, the generated intermediate characters are very smooth.

Our method can be also used to generate intermediate characters from sample cursive characters for general Chinese characters and Japanese *Kanji* characters (WANG *et al.*, 1997).

Moreover, our method is easily extended to generate intermediate characters from more than 2 sample characters. Using the printing fonts and handwriting fonts as input samples, we can use our system to generate fonts that are more artistic than that written by ourselves, but simultaneously retains our personality.

On the other hand, since we take the sampling points with equal-interval, when the number of sampling points is not enough, the generated fonts may lose the rhythm of the initial fonts. Moreover, our method is not suitable for the simulation of the origin of few Japanese *Hiragara* characters, for example “ki”, the seventh Japanese *Hiragara* characters in Fig. 1. We will try to overcome these shortcomings in our next system version.

REFERENCES

BEIER, T. and NEELY, S. (1992) Feature-based image metamorphosis, *Computer Graphics*, **26**, 35–42.

BETHEL, E. W. and USELTON, S. P. (1989) Shape distortion in computer-assisted keyframe animation, in *Proceedings of Computer Animation ’89*, Springer.

DEBOOR, C. (1972) On calculating with B-splines, *J. Approx. Theory*, 50–62.

GORDON, W. and RIESENFELD, R. (1974) B-spline curves and surfaces, in *Computer Aided Geometric Design* (eds. R. E. Barnhill and R. F. Riesenfeld), pp. 95–126, Academic Press, San Diego, CA, USA.

HE, T., WANG, S. and KAUFMAN, A. (1994) Wavelet-based volume morphing, in *Proceedings of Visualization ’94*, IEEE Computer Society.

HUGHES, J. (1992) Scheduled fourier volume morphing, *Computer Graphics*, **26**, 43–46.

KABA, T. (1979) *Characters of Japanese*, Iwanami New Books, Iwanami-shoten, Tokyo, Japan.

KASS, A., WITKIN, A. and TERZOPOULOS, D. (1987) Snakes: Active contour models, in *Proc. First Int’l Conf. Computer Vision*, pp. 259–269.

LAI, K. F. and CHIN, R. T. (1995) Deformable contours: Modeling and extraction, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**, No. 11, 1084–1090.

MICHAEL, U. (1999) Splines: A perfect fit for signal and image processing, *IEEE Signal Processing Magazine*, **16**, No. 6, 22–38, November 1999.

MOEBIUS, F. (1885) *Gesammelte Werke*, Verlag von s. Hirzel, Hirzel, Germany.

POGGIO, T., TORRE, V. and KOCH, C. (1985) Computational vision and regularization theory, *Nature*, **317**, No. 26, September, 314–319.

SEDERBERG, T. W. and GREENWOOD, E. (1992) A physically based approach to 2D shape blending, *Computer Graphics*, **26**, 25–34.

SEDERBERG, T. W., GAO, P., WANG, G. and MU, H. (1993) 2D shape blending: An intrinsic solution to the vertex path problem, *Computer Graphics*, **27**, 15–18.

SHAPIRO, M. and RAPPOPORT, A. (1995), Shape blending using the star-skeleton representation, *IEEE Computer Graphics and Applications*, March.

TAKEUCHI, Y. (1998) Deformation method for single string curve, *Information Processing Society of Japan*, **39**, No. 11, 2997–3008.

WANG, L., HE, L., NAKAMURA, T. and ITOH, H. (1997) A method of generating calligraphy of Japanese character using deformable contours, *IJCAI-1997*, **2**, 1050–1056.