Writer Dependent Online Handwriting Generation with Bayesian Network [*]

Hyunil Choi, Sung Jung Cho†, and Jin H. Kim

CS Div., EECS Dept., KAIST,
373-1 Kusong-dong, Yousong-ku, Daejon, Korea
{hichoi, jkim}@ai.kaist.ac.kr

† Samsung Advanced Institute of Technology,
San 14-1, Nongseo-ri, Giheung-eup, Yongin-si, Gyeonggi-do, Korea
sung-jung.cho@samsung.com

Abstract

In this paper, we propose a method to generate writer dependent(WD) handwritings. We modelled the shape of character both globally and locally with probabilistic relationships between character components. Then writer independent(WI) model was trained with lots of data. Once WI model was built, the model was adapted to a training example to maximize likelihood of the example by minimization of squared error between model and instance. The experimental results of WI numeral character generation showed that global shape consistencies and variabilities of local shape were preserved. The relationships from WI model were still valid in WD models by proposed adpatation technique so that we could generate natural-looking writer specific handwritings.

1 Introduction

The recent emergence of pen computers makes handwriting more convenient and natural input method in human-computer interaction (HCI). Computer users are getting familiar to input messages with handwriting. If an user has a font which shows his/her own writing style, messages can be inputted faster than handwriting, looked friendly. From this requirement, we made a system which generates handwriting when a person types text strings. It is analogous to the text-to-speech system in the speech recognition literature. However, it is different in that handwritten characters, instead of speech, are generated, and they are trainable by writers.

There are several possible applications of character generations. We could make handwriting fonts automatically. With some handwriting control parameters like cursiveness to the baseline system, various fonts can be made. If the system is capable of simulating a specific writer's handwritings, it is also possible to make personalized font. Suppose that an user has a computer font which shows his own writing style. He can use the font in writing E-mail or greetings. We can say the counterparts might feel friendly towards him.

Handwriting generation task has been studied for a long time. The motor model [8] views handwriting process as a result of motor processes that control writing method. Although it established unified framework for various kinds of handwriting generation models, strong assumptions on both linearity of subsystems and simplification of complex muscle movement are impractical. Based on the motor models, *delta-lognormal theory* [7] enables parametric representation of handwriting generation. Motor model based approaches can be used to simulate handwriting in terms of motor aspect. But it is commonly known that the dynamic-inverse problem is hard to solve [9].

For the pratical use of handwriting generation, learning-based approach is another promising methodology. By collecting handwriting examples from a writer, the system learns the writer's writing style. In [3], common glyphs are predefined and the system keeps handwritten glyphs of the writer. When texts are given, the system concates corresponding glyphs to make handwriting. In spite of its simplicity, it is hard to extend the system if a language has many classes. Wang [10] extracts strokes of letter and ligature using tri-unit handwriting model. By minimizing deformable energy, synthesized handwriting is generated. Recently, curve analogies [4] enable us to simulate a transfor-

[*] This work has been partially supported by the Korean Ministry of Science and Technology for National Research Laboratory Program.

mation from a source shape to corresponding target shape. The results showed fluent simulation result when it was applied to handwriting. But the method has high complexity so that the practical use is hard to be taken.

We extend previous work [2] to generate writer specific handwriting. We model shape of character both globally and locally with probabilistic relationships between character components. Global shape is modelled with several strokes, local shape is represented as points in a stroke. Once we have trained model for a letter, the most probable(MP) components produce a handwriting. We take the idea of writer adaptation in character recognition fields to generate the writer dependent(WD) handwritings. We built WI model first, then adapted the model to given example to maximize likelihood of the example. The proposed method has advantages in that, by modelling coarse-to-fine level of character shapes with probabilistic relationships, it is possible to maintain global shape consistency as well as variability of local shape. For the WD handwriting generation, the system requires only one training example per class to simulate specific writer's handwritings.

This paper is organized in four parts. The Bayesian network-based character modelling was explained in section 2. A handwriting generation algorihtm with WI models was introduced in section 3. The adaptation of WI model to the instance was described in section 4. Then we showed the generated handwritings using our system followed by the conclusions of this paper.

2 Writer Independent Handwriting Generation

2.1 Character Model

A character can be decomposed into three different level components: letter, strokes and points. A letter is structurally composed of several strokes. A stroke is a straight or nearly straight trace that has distinct directions from connected traces in writing order, and is composed of points. A point, the primitive component, is represented by its (x, y) coordinate.

We proposed a Bayesian network framework for explicitly modelling components and their relationships of a letter [1]. A letter is modelljudgmented with hierarchical components: stroke models and point models. Each model is constructed with subcomponents and their relationships. A point model, the primitive one, is represented by a 2-D Gaussian for positions on X-Y plane. Relationships are modelled with position-dependencies between components.

A Bayesian network [5] is a probabilistic graph for representing random variables and their dependencies. The joint probability of random variables $\{X_1, \ldots, X_N\}$ in a Bayesian network S is calculated by the multiplication of

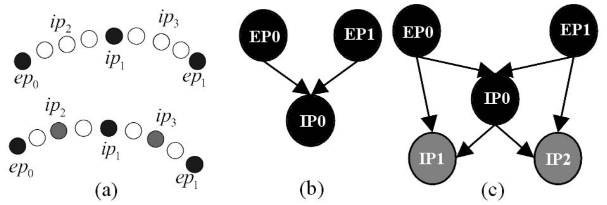

Figure 1. Recursive construction of a stroke model.

local conditional probabilities of all nodes. Let a node X_i in S denote the random variable X_i and $pa(i)$ denote the parent nodes of X_i, from which dependent arcs come to the node X_i. The joint probability of $\{X_1, \ldots, X_N\}$ becomes $P(X_1, \ldots, X_N) = \prod_{i=1}^{n} P(X_i|pa(i))$.

To model continous variables, conditional Gaussian distributions [6] were used. The mean of a random variable is assumed to be determined from the linearly weighted sum of parent variable values. When a multivariate random variable X depends on $X_1, \ldots, X_N$, the conditional probability distribution is given as follows:

$$P(X = \boldsymbol{x}|X_1 = \boldsymbol{x}_1, \ldots, X_n = \boldsymbol{x}_N) = G(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad (1)$$

where G, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ are Gaussian, mean vector, and covariance matrix, respectively. The mean vector is determined from the parent variable values $\boldsymbol{z} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T, 1]$ with a linear regression matrix $\boldsymbol{W}$, i.e., $\boldsymbol{\mu} = \boldsymbol{W}\boldsymbol{z}$, where $\boldsymbol{W}$ is a $d \times k$ matrix, d is the dimension of X, and k is the dimension of $\boldsymbol{z}$.

A point has the attribute of (x, y) position on the 2-D plane. So, a point model has 2-D Gaussian distribution for modelling 2-D point positions. It is represented by one node in Bayesian networks. When a point $\boldsymbol{q} = (x, y)$ depends on other points $\boldsymbol{q}_1 = (x_1, y_1), \ldots, \boldsymbol{q}_n = (x_N, y_N)$, its matching probability $P(\boldsymbol{q}|\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N)$ is given from the conditional Gaussian distribution (Eq. (1)) by setting $\boldsymbol{q} = (x, y)$ and $\boldsymbol{z} = [\boldsymbol{q}_1^T, \ldots, \boldsymbol{q}_N^T, 1]$.

A stroke is composed of points. Therefore, a stroke model is composed of point models with their relationships, called within-stroke relationships (WSRs). A stroke model is constructed by recursively adding the mid point models and specifying WSRs. At the mid point, the lengths of the left and the right partial strokes are equal. The WSR is defined as the dependency of a mid point from two end points of a stroke. Fig. (1) shows an example of the recursive construction of a stroke model. In Fig. 1 (a), an example of stroke instances was given. At the first recursion ($d = 1$), IP_1 is added for modelling $\boldsymbol{ip}_1$ with the WSR from EP_0 and EP_1 (Fig. (1) (b)). At $d = 2$, IP_2 and IP_3 are added

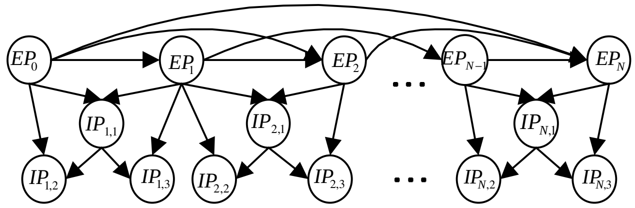

Figure 2. Bayesian network representation of a letter model.

for the left and the right partial strokes (Fig. (1) (c)). This recursion stops when the covariances of newly added point models become smaller than some threshold.

A letter consisted of strokes which have relationships. Therefore, a letter model consists of stroke models with inter-stroke relationships (ISRs). ISRs are represented with dependencies of stroke-end-points(EPs). Ideally, a stroke gets influence from all the points of other strokes. However, representing all the relationships is too complex and redundant. So, we encapsulate them as relationships of EPs. A letter model is constructed by concatenating stroke models according to their writing order and specifying ISRs.

Figure 2 shows the Bayesian network based letter model with N strokes and the stroke recursion depth $D = 2$. EP_i are stroke-end-point models and $IP_{i,j}$ are point-models within the i-th stroke. The right end point of the previous stroke is shared with the left one of the following stroke. ISRs are represented by the arcs between EP_i, and WSRs are represented by incoming arcs to $IP_{i,j}$.

2.2 Handwriting Generation Algorithm

Since the Bayesian network is a generative model, we can choose the MP characters from the Bayesian network based letter models. The character can be interpreted as the most representative character pattern that each model has. If a model successfully learns the concept of characters from training data, then it can generate natural shapes and vice versa.

A WI character is generated from the proposed character model by generating points according to their dependency order. The points without any dependency are generated at first. Then, points that depend on previously generated points are sequentially generated up to predefined recursion level D. Since each point model has covariance matrix, we can randomly add some noise. Note that in case of adding noises, dependencies are still valid so that generated handwriting would have natural looking. Alg. (1) shows proposed method to produce MP points from a model.

Algorithm 1 Hierarchical Handwriting Generation Algorithm

```
{Global Shape Generation}
for $i = 0$ to $N$ do
   Generate $\boldsymbol{ep}_i^* = \arg\max_{\boldsymbol{ep}_i} P(EP_i|pa(i))$
   Add random noise $\boldsymbol{ep}_i^* \leftarrow \boldsymbol{ep}_i^* + \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{EP_i})$
end for
{Local Shape Generation}
for $i = 0$ to $N$ do
   for $d = 1$ to $D$ do
      for $j = 2^{d-1}$ to $2^d - 1$ do
         Generate $\boldsymbol{ip}_{i,j}^* = \arg\max_{\boldsymbol{ip}_{i,j}} P(IP_{i,j}|pa(i,j))$
         Add random noise $\boldsymbol{ip}_{i,j}^* \leftarrow \boldsymbol{ip}_{i,j}^* + \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{IP_{i,j}})$
      end for
   end for
end for
```

3 Writer Dependent Handwriting Generation : Writer Adaptation

3.1 Character Model Adaptation

In this section, we describe a method to adapt individual handwriting style to our WI model. The reason why we utilize WI model in WD handwriting generation is because the WI model parameters were estimated from lots of examples. It means they have been keeping reliable relationships between components so that generated handwriting seems to be natural. Another reason is that simulation of writing variation is possible while keeping consistency of shape by utilizing probability distribution.

When a training example is given, we first match the instance with the model to find out strokes [1]. This step mainly investigates global shape(EPs) in terms of geometric relationships of strokes. Once the MP strokes are found, local shape(IPs) within a stroke is matched in each stroke. Having both global and local shape, we adapt the parameters of global shape to the instance and those of local shapes successively. After adaptation is hierarchically performed, we can generate the MP character to the instance with relationships in WI model.

The adaptation is performed on model parameters $\boldsymbol{\theta}$ that are linear regressive Gaussian(Eq. (1)). It has two parameters: $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \mathbf{\Sigma}\}$, where $\boldsymbol{\mu} = \boldsymbol{W}\boldsymbol{z}$ is mean vector, $\mathbf{\Sigma}$ is covariance matrix. The mean vector is more important in the generation because it determines the position of corresponding point. The MP points of a distribution, therefore, would be the mean vector itself in the notion of maximum likelihood parameter estimation. So we try to adapt WI mean vectors to the given points of the instance and use the covariance matrix of WI model.

Before we explain the detail of proposed method, let us define some notations. $\boldsymbol{O}$ is an instance which has a set of points $\boldsymbol{O} = \{\boldsymbol{ep}_i, \boldsymbol{ip}_{i,j} | 1 \le i \le n, 1 \le j \le 2^d - 1\}$. Let L parents of point $\boldsymbol{o}_k \in \boldsymbol{O}$ be $\boldsymbol{z}_k = \left[pa_1(k)^T, \dots, pa_L(k)^T\right]$, where $pa_l(k)$ is l-th parent of point k. Then our adaptation problem is formalized as follows; given $\boldsymbol{O}$, find $\hat{\boldsymbol{W}}_k$ which minimizes $\boldsymbol{o}_k = \hat{\boldsymbol{W}}_k \boldsymbol{z}_k, \forall \boldsymbol{o}_k \in \boldsymbol{O}$.

Note that $\boldsymbol{W}_k$ in WI model doesn't guarantee equality of that equation. Assuming there are some error portions in the regression matrix of WI in predicting $\boldsymbol{o}_k$, we introduce error matrix $\boldsymbol{E}_k$ into the regression matrix in the form of $(\boldsymbol{W}_k + \boldsymbol{E}_k)\boldsymbol{z}_k$[1]. Then we find $\hat{\boldsymbol{E}}$ which satisfies $(\boldsymbol{W} + \boldsymbol{E})\boldsymbol{z} - \boldsymbol{o} = \boldsymbol{0}$.
The linear system can be solved by traditional gradient search method. We define squared error function as follows.

$$J(\boldsymbol{E}; \boldsymbol{o}) = ((\boldsymbol{W} + \boldsymbol{E})\boldsymbol{z} - \boldsymbol{o})^T((\boldsymbol{W} + \boldsymbol{E})\boldsymbol{z} - \boldsymbol{o}) \qquad (2)$$

and we find $\hat{\boldsymbol{E}}$ in which $\nabla J(\boldsymbol{E}; \boldsymbol{o}) = \boldsymbol{0}$. By solving simple calculus, we obtain $\hat{\boldsymbol{E}} = \boldsymbol{o}\boldsymbol{z}^T(\boldsymbol{z}\boldsymbol{z}^T)^{-1} - \boldsymbol{W}$. To leave a room for balancing WI and WD, we introduce interpolation coefficient λ, $0 \le \lambda \le 1$ to the error matrix: $\hat{\mu} = (\boldsymbol{W} + \lambda\hat{\boldsymbol{E}})\boldsymbol{z}$. As λ approaches to 1, a generated handwriting will have more similar shape to the instance. If $\lambda \to 0$, as opposite case, we neglect the effect of adaptation.

3.2 Writer Adaptation Algorithm

The proposed adaptation algorithm on a point model is hierarchically performed for all point models in a letter model. The EPs that correspond to global shape are adapted first, then IPs to local shapes are adapted until predefined recursion depth D' is reached. Alg. (2) describes the hierarchical adaptation of a letter.

WD handwriting generation algorithm is basically the same as Alg. (1) except that we use the adapted distribution of each point model. However, the recursion depth D' doesn't need to be the same as D of WI model.

4 Extention to Hangul

A Hangul syllable character consists of graphemes (a first consonant, a vowel and an optional last consonant) and ligatures between them. These components have geometrical relationships one another. Therefore, a Hangul syllable model is constructed with grapheme and ligature models which have relationships one another, called inter-grapheme relationships (IGRs).

The shape of Hangul syllable is catagorized as one of six types by 2 dimensional position of graphemes. IGRs are grouped into six types in Hangul syllable characters. They are modeled by the relationships between bounding boxes. Fig. (3) shows Bayesian network models for IGRs. Each bounding box depends on the type variable T and the previous graphemes. Because a vowel has one bounding box BJ for Type 1, 2, 4 and 5, and two (BJ_H and BJ_v) for Type 3 and 6, it is represented by a rectangular notation as shown in (c). According to the type T, BJ becomes one or two random variables.

Algorithm 2 Hierarchical adaptation of a letter model

```
{Global Shape Adaptation}
Match given intance with corresponding model to get $\boldsymbol{O}$
for $i = 0$ to $N$ do
    Find $\hat{\boldsymbol{E}}_i$ which minimizes $J(\boldsymbol{E}_i; \boldsymbol{ep}_i)$
    Adjust the mean $\boldsymbol{\mu}_i \leftarrow (\boldsymbol{W}_i + \lambda_i \hat{\boldsymbol{E}}_i)\boldsymbol{z}_i$
end for
{Local Shape Adaptation}
for $i = 0$ to $N$ do
    for $d = 1$ to $D'$ do
        for $j = 2^{d-1}$ to $2^d - 1$ do
            Find $\hat{\boldsymbol{E}}_{i,j}$ which minimizes $J(\boldsymbol{E}_{i,j}; \boldsymbol{ip}_{i,j})$
            Adjust the mean $\boldsymbol{\mu}_{i,j} \leftarrow (\boldsymbol{W}_{i,j} + \lambda_{i,j} \hat{\boldsymbol{E}}_{i,j})\boldsymbol{z}_{i,j}$
        end for
    end for
end for
```

The adatation algorithm for Hangul syllable is basically the same in the sense of minimization of error matrices to the given bounding boxes and graphemes using Eq (2). The bounding box models are adapted first, then the grapheme models are adapted. Once the WD model for a given syllable is built, bounding boxes, graphemes in each bounding box are successively generated in topological order.

5 Experimental Results

5.1 Writer Independent Handwriting Generation

We performed WI handwriting genreation of numeral characters. The training set consisted of 4,046 numeral charaacters from KAIST DB. It has well written characters by high school students without any writing restriction. Fig. (4) shows the MP instances which were generated by proposed method. We can see that linear traces as well as curves seem to be natural. This implies the Bayesian network based modelling successfully learned not only common writing styles, but also dependencies between character components from lots of examples. For the generation of Hangul, readers can refer the paper [2].

[1] We eleminate the point index k in successive equations for notational simplicity.

0-7695-2187-8/04 $20.00 © 2004 IEEE

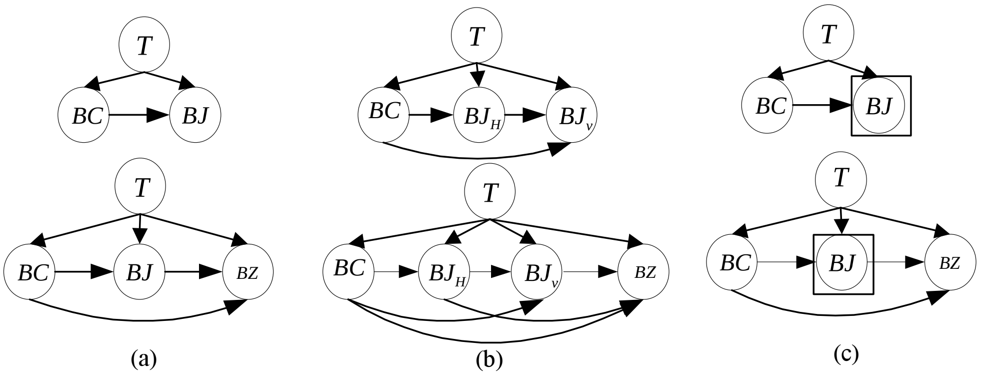

Figure 3. Six types of grapheme relationship models. (a) Type 1, 2 (above figure) and Type 4,5 (below). (b) Type 3 (above) and Type 6 (below). (c) Unified notation of Type 1, 2, 3 (above) and Type 4, 5, 6 (below).

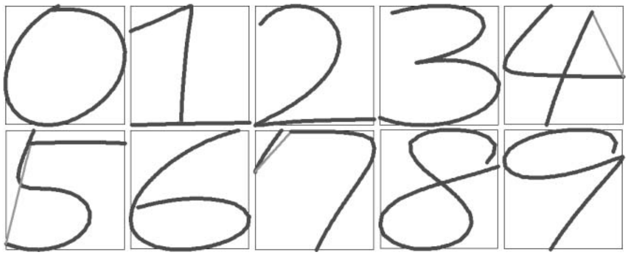

Figure 4. Generated numeral characters(KAISTDB).

5.2 Writer Dependent Handwriting Generation

The WI model from Sec. (5.1) was used for the baseline system of the proposed adaptation method. To see the effect of adaptation in terms of shape similarity, we introduced the sum of squared error(SSE) between points of given instance and MP points of WI/WD model(Fig. (5)). The error of WI model is a constant becase the parameters of the model are fixed. However WD model reduces the error significantly as λ approaches to 1. We were able to verify that other examples have the same tendency of the figure.

Since most writers produce slightly different handwritings of same class according to writing condition, we use the distributions of point models to simulate writing variation by Alg. (1) with WD models. As interpolation coefficient λ approaches to zero, the generated handwriting tends to be more similar to the training example. However, the proposed algorithm doesn't consider adaptation of ligature as shown in '1'. This can be easily done by copying the status of corresponding ligature in the training example to the

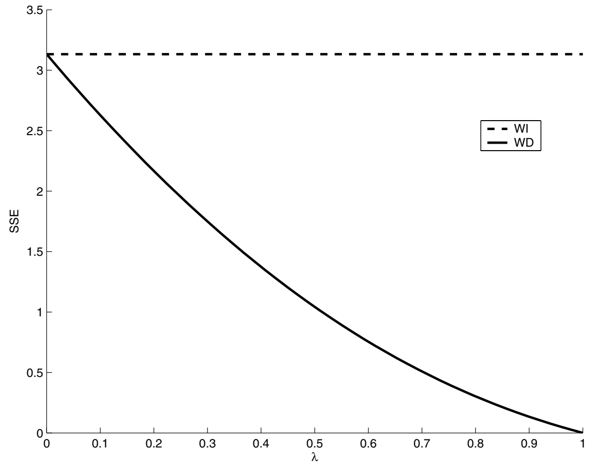

Figure 5. Sum of squared error between instance and MP points of an adapted model. Solid line is SSE between instance and MP points of WD model, dashed line is SSE between instance and MP points of WI model.

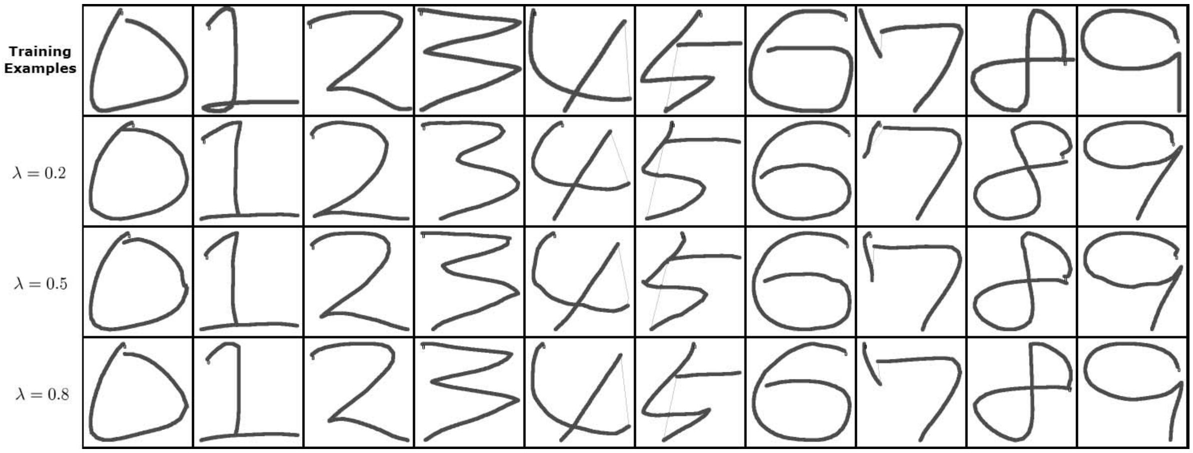

Figure 6. Generated handwritings with the adapted models. Training examples and randomly chosen handwritings from the adapted models were shown in shaded boxes and normal boxes, respectively.

that of generated handwriting.

6 Conclusions & Future Works

We proposed a method to generate writer dependent handwritings which was based on writer adpatation. The shape of character was hierarchically modelled with probabilistic relationships between character components. After WI models were trained with lots of data, error minimization was used to adjust WI model parameters to get WD models. We demonstrated that WI models learned common writing styles and proposed WD handwriting generation algorithm simulated specific writer's handwritings successfully.

Our future works are about finding a way to decide proper λ for each point and word-level generation. For the proper λ, it is hard to find objective measure in choosing λ of each point. Since, if certain storke in one's handwriting shows erratic shape, the stroke should be adapted more than other strokes by making λ higher than that of other strokes. One possible solution is use of the amount of stroke error as a function of λ.

The word-level generation is undergoing by connecting proposed letter models with corresponding ligature models.

References

[1] S. Cho and J. Kim. Bayesian network modeling of hangul characters for on-line handwriting recognition. *International Conferene on Document Analysis and Recognition*, pages 173–179, 2003.

[2] H. I. Choi, S. Cho, and J. Kim. Generation of handwritten characters with bayesian network based on-line handwriting recognizers. *International Conferene on Document Analysis and Recognition*, pages 995–1001, 2003.

[3] I. Guyon. Handwriting synthesis from handwritten glyphs. *International Workshop on Frontiers of Handwriting Recognition*, 1996.

[4] A. Herztmann, N. Oliver, S. Seitz, and B. Curless. Curve analogies. *Eurographics Workshop on Rendering*, pages 233–246, 2002.

[5] F. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[6] K. Murphy. Inference and learning in hybrid bayesian networks. *Technical Report 990, U.C.Berkeley, Dept. Comp. Sci*, 1998.

[7] R. Plamondon and W. Guerfali. The generation of handwriting with delta-lognormal synergies. *Bilogical Cybernetics*, 78:119–132, 1998.

[8] R. Plamondon and F. J. Maarse. An evaluation of motor models of handwriting. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1060–1072, 1989.

[9] Y. Singer and N. Tishby. Dynamical encoding of cursive handwriting. *Biological Cybernetics*, 71(3):227–237, 1994.

[10] J. Wang, C. Wu, Y.-Q. Xu, H.-Y. Shum, and L. Ji. Learning-based cursive handwriting synthesis. *International Workshop on Frontiers in Handwriting Recognition*, pages 157–162, 2002.