Learning-based Cursive Handwriting Synthesis

Jue Wang[1,2 *], Chenyu Wu[1,2] , Ying-Qing Xu[1] , Heung-Yeung Shum[1] and Liang Ji[2]
1. Microsoft Research Asia, Beijing 100080, China
2. Dept. of Automation, Tsinghua Univ., Beijing 100084, China
E-mail: yqxu@microsoft.com

Abstract

In this paper, an integrated approach for modeling, learning and synthesizing personal cursive handwriting is proposed. Cursive handwriting is modeled by a tri-unit handwriting model, which focuses on both the handwritten letters and the interconnection strokes of adjacent letters. Handwriting strokes are formed from generative models that are based on control points and B-spline curves. In the two-step learning process, a template-based matching algorithm and a data congealing algorithm are proposed to extract training vectors from handwriting samples, and then letter style models and concatenation style models are trained separately. In the synthesis process, isolated letters and ligature strokes are generated from learned models and concatenated with each other to produce the whole word trajectory, with guidance from a deformable model. Experimental results show that the proposed system can effectively learn the individual style of cursive handwriting and has the ability to generate novel handwriting of the same style.

1. Introduction

Pen-based interfaces are now a hotspot in Human-Machine Interface (HCI) research because in numerous situations, a pen together with a notepad is more convenient than a keyboard or a mouse. The flourish of pen-based devices such as Tablet PCs brings a great demand for various cursive handwriting computing techniques. When writing a note on a Tablet PC, if the computer can automatically correct some written errors and generate some predefined handwriting strokes, communication through a pen-based interface would be more effective and intelligent. Furthermore, handwriting is preferable to typed text in some cases because it adds a personal touch. All these applications bring an urgent requirement for handwriting synthesis techniques.

The problem of handwriting synthesis has been addressed for a substantial amount of time and many studies can be found in the literature. Generally speaking, these approaches can be divided into two categories based on their properties: movement simulation techniques and shape simulation methods. Most movement simulation approaches are based on motor models [1] and try to model the process of handwriting production [2, 3]. Shape simulation techniques, on the contrary, only consider the trajectory of handwriting. They are more practical than movement simulation techniques when the dynamic information of handwriting is not available and the trajectory of handwriting has been re-sampled by other processors such as recognizers, as addressed in [4]. Our approach is a shape simulation technique, as the one proposed in [5]. In that study, a straightforward approach was proposed in which handwriting is synthesized from collected handwritten glyphs. However, this work lacks mathematical models and explanations for individual styles of cursive handwriting, which are essential for achieving satisfying synthesis results.

In this paper, a mathematical analysis of cursive handwriting style is given, and a series of models are proposed to capture the characteristics of different writing styles. Based on the fact that in individual cursive handwriting, not only the styles of letters but the styles of ligature strokes between letters are unique, a tri-unit cursive handwriting model is proposed, which enables us to extract letter strokes and ligature strokes from cursive handwriting. Each stroke is modeled by a cubic B-spline with some control points extracted by 1-D Gabor filters. A parametric generative model is proposed based on the distributions of control points to model writing styles of these strokes. After the training process, generative models can synthesize new letters and ligature strokes of the same style. A deformable model is then applied to incorporate these strokes into integrated, seemingly natural handwritten words.

2. Models

2.1 Tri-Unit Handwriting Model

The model of cursive handwriting plays the most important role in determining how natural our synthetic results appear. Because letters are connected with their

*Jue and Chenyu participated in this work when they were working as interns at Microsoft Research Asia.

neighbors in cursive handwriting, each letter is "context related", which means that each letter is connected to and affected by its two neighbors in specific ways. Based on this fact, we borrow the ideal of "triphthong" [12] in speech processing and propose a tri-unit handwriting model. The trajectory of each letter can be divided into three parts in sequence: the head, the body and the tail units. The head unit of each letter is connected with and influenced by the tail unit of the previous letter and the tail unit is connected with and influences the head unit of the next letter. The three units combine together to give the whole trajectory of the letter, and the tail and head unit of adjacent letters together can be called the concatenation part.

2.2 Generative Models

Because our system is recognition- and segmentation-based, it is assumed that the contents of the handwriting samples are known and that adjacent letters are roughly segmented. In our proposed generative models, each letter is modeled by a B-spline curve controlled by some control points. Let $X \equiv \{x_1, x_2, ..., x_{n-1}, x_n\}$ denote a written stroke in terms of its n control points. The i^{th} control point is located at (p_i^x, p_i^y). The location of any point $s(b)$ on the stroke can be expressed by the control points using the following linear function:

$$s(b) = \sum_{i=1}^{n} c_i(b) x_i . \quad (1)$$

To extract control points automatically from handwritten strokes, we use a series of 1-D Gabor filters and take the strongest response points as control points. The 1-D Gabor filter is defined as:

$$\begin{cases} G\sin(x) = const \cdot \exp\left\{-\frac{2x^2}{T^2}\sin(\frac{2\pi}{T}x)\right\} \\ G\cos(x) = const \cdot \exp\left\{-\frac{2x^2}{T^2}\cos(\frac{2\pi}{T}x)\right\} \end{cases} \quad (2)$$

where T is the scale of the filter. Since handwriting data contains an x component and a y component, each component is viewed as an individual 1-D signal. Using this method, a series of multi-scale control points can be extracted from handwritten strokes, as shown in Fig. 1. The reconstructed strokes using B-splines are also shown. The trivial difference between the original and reconstructed trajectories shows that the extracted control points can represent handwriting data quite well.

It is obvious that the control points at larger scales contain more global and structural information of the stroke, while those at smaller scales typically reflect local variations. For a handwritten letter, the control points at the largest scale roughly represent the structural characteristics, as shown in Fig. 2. This fact makes the implementation of the proposed tri-unit handwriting model quite easy. For a letter with n (n>=3) control points

Figure 1. (a) Control points extracted from the original strait. (b) Reconstructed strait using control points and B-splines.

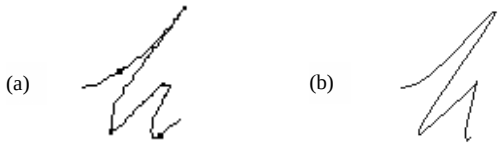


Figure 2. (a) Control points in the largest scale of the handwritten letter "h". (b) Reconstructed "h" using only the largest- scale control points.

at the largest scale, the second control point is viewed as the segment point between the head and the body units, while the $(n-1)th$ control point is viewed as the segment point between the body and the tail units. In the case of n<3, the control points at the next smaller scale are used for substitution. Control points at the largest scale are called main control points. Since each head part and tail part contain two main control points, there are always four main control points in the stroke of a concatenation part.

2.3 Writing Style Models

According to the generative model, the characteristics of a letter are determined by the locations of its control points. So the problem of learning the writing style of a letter is converted to learning the specific distributions of these control points. For convenience, the following analysis is focused on one style of a certain letter, which is denoted as S. A written style model is adopted as an instantiation of the general framework of latent variable models. In this study, the manifest variables are the control point vectors extracted from handwritten strokes, and the latent variable is the style intrinsically involved in these straits.

When a stroke X_i is being written, the writer is guided intrinsically by the inherent style S and this process is denoted as F_i. Furthermore, each strait has its unique global style-independent parameters such as scale, location and slant. So we have

$$X_i = F_i(S) = F(S, N_i, A_i) \quad (3)$$

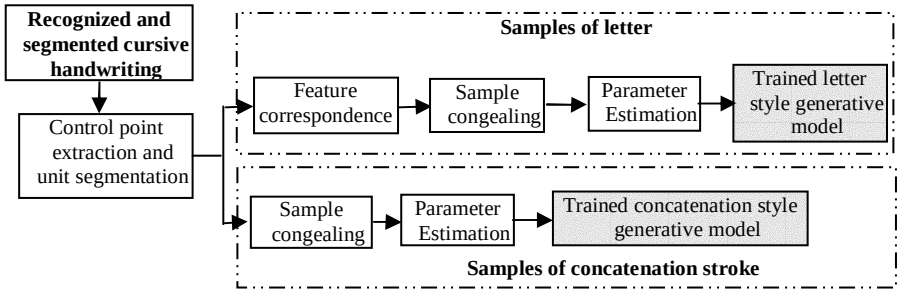


Figure 3. Flow chart of the model learning framework.

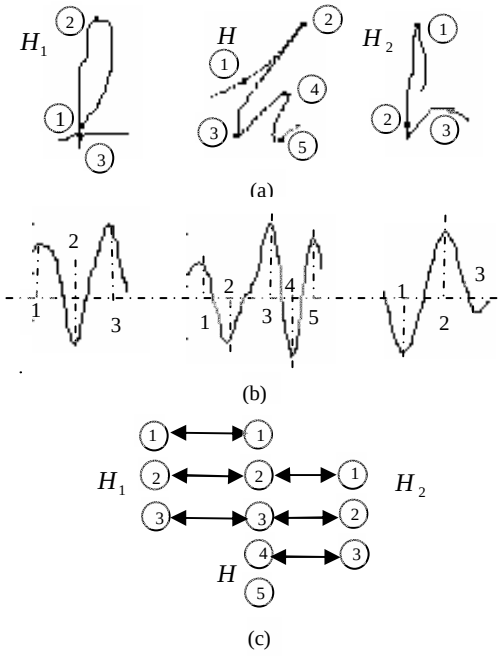


Figure 4. Illustration of the template based control point matching algorithm. (a) Two samples of "h" and the template of "h" (centre). (b) Responses of Y component data to Gabor filters at the largest scale. (c) Indirectly matching results using the matching algorithm.

where N_i is structural noise, and A_i stands for affine transform parameters. Introducing these parameters enables us to unify the different F_i's into a general F.

Typically, affine transforms are uncorrelated with the intrinsic style of a written letter, but largely affect letter appearance. Miller et al. proposed a congealing algorithm to learn these affine parameters of each sample [11], and some EM-based algorithms are proposed in [9] and [10] for the same purpose. In this study, a similar algorithm is applied to diminish the effects of affine transforms in the training process, which will be discussed in detail in the next section. In this way, (3) can be divided into two steps as follows:

$$X_i^* = F(S, N_i)$$
$$X_i = T_A(X_i^*, A_i) \quad (4)$$

where T_A stands for the affine transform. The congealing algorithm is used to get X_i^* from X_i by eliminating the effects of A_i. We denote $H = F(S)$ as the "standard" instantiation of the writing style. Following [8] and assuming a Gaussian distribution for these samples, the probability of the control points lying within a small hypervolume δV is approximately:

$$p(X) = \delta V \frac{1}{(2\pi)^n |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(X-H)^T \Sigma^{-1}(X-H)\right] \quad (5)$$

where Σ is the covariance matrix of the distribution.

3. Learning Strategies

3.1 Learning Framework

The learning framework of the proposed system is shown in Fig. 3. Letters and ligature strokes are extracted from recognized and segmented cursive handwriting samples. Due to the variability of different writing styles, a template-based feature correspondence algorithm is proposed to match control points of different samples correctly. Before the parameter learning process, a sample congealing algorithm is used to separate global deformations from sample strokes. These algorithms will be discussed below.

3.2 Template-based Feature Correspondence

In [8], the number of control points of each digit is fixed and there is no matching problem. However, due to the complexity of cursive handwriting, the numbers of main control points extracted from different samples of different writing styles are typically different. Even in the situation that the numbers of the main control points are equal for two samples, these control points do not necessarily correspond to each other naturally. To solve this problem, an automatic matching algorithm is proposed. The basic idea of the algorithm is that rather than matching control points in different samples directly,

control points of a sample are matched to a "self-contained" template, which acts as a bridge between different samples. In this way, control points of different samples are matched correctly, although indirectly.

The vector of main (largest-scale) control points extracted from the template is denoted as $T(x_1, x_2, \ldots, x_m)$. For a sample X_i, suppose it has *n* main control points. The template is carefully designed to make $m \geq n$. For correct matching of control points, we try to match their responses to Gabor filters. Each control point corresponds to a local maxima of the Gabor filtering response, and the value of the response at this point is denoted as $g(k)$, where k is the index number of a control point in the vector. A matching function is calculated as:

$$M(t) = \sum_{k=1}^{n} g_{X_i}(k+t) \cdot g_T(k), \qquad t = 0, \cdots, m-n \tag{6}$$

An optimal t_m is calculated as

$$t_m = \arg\max_t M(t) \tag{7}$$

That means the kth control point of the sample corresponds to the $(k+t_m)th$ control point of the template.

Another advantage of the matching algorithm is that it can do sample classification automatically. Since the matching modes of different samples may be different, samples with the same matching mode are clustered and samples with different matching modes are viewed as instantiations of different writing styles. Samples of different styles are trained separately.

The template-based matching algorithm is used for feature correspondence of the segmented letters. For the concatenation part, no special matching algorithm is needed because the four main control points of different concatenation samples can be matched naturally.

3.3 Data Congealing Algorithm

As discussed above, the locations of corresponding control points are not clustered very well due to the different affine transforms. Since the structure of a handwritten letter is largely determined by the largest-scale control points, only these points are considered in the congealing algorithm, which is similar to the one proposed in [11]. A joint entropy is defined as the criterion in that study and the proposed algorithm works well in image-based applications, but it is not suitable in our stroke-based situation. In our study, a deformable energy-based criterion, which is similar to the data mismatch criterion proposed in [9], is defined as:

$$E = -\log\left[\frac{1}{N_s}\sum_{i=1}^{N_s}\exp\left(-\frac{\left\|X_i - \overline{X}\right\|^2}{2 \cdot V_X}\right)\right] \tag{8}$$

where N_s is the number of samples, $\overline{X}$ is the mean vector calculated as

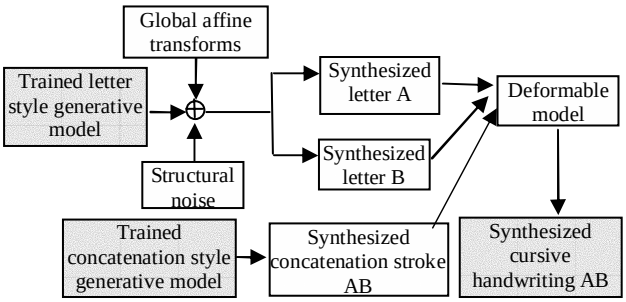


Figure 5. Flow chart of the cursive handwriting synthesis framework.

$$\overline{X} = \frac{1}{N_s}\sum_{i=1}^{N_s} X_i \tag{9}$$

and V_X is the variance of the Gaussian calculated as

$$V_X = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|X_i - \overline{X}\right\|^2 \tag{10}$$

The algorithm is formally described as follows:

1. Maintain an affine transform matrix U_i for each sample, which is set to identity initially.
2. Compute the deformable energy-based criterion E.
3. Repeat until convergence:
   (a) For each sample X_i,
      i. For each one of the six unit affine matrixes [11] $A_j, j = 1, \cdots, 6$,
         A. Let $U_i^{new} = A_j U_i$
         B. Apply U_i^{new} to the sample and recalculate the criterion E.
         C. If E has been reduced, accept U_i^{new}, otherwise:
         D. Let $U_i^{new} = A_j^{-1} U_i$ and apply again. If E has been reduced, accept U_i^{new}, otherwise revert to U_i.
4. End.

After the congealing process, the distributions of the main control points are more likely to satisfy the Gaussian distribution assumption in (5).

4. Synthesis Strategies

4.1 Synthesis Framework

The handwriting synthesis framework is illustrated in Fig. 5. Instantiations of letters A and B are generated from the trained generative models, and some structural noise and transforms are added to increase the variability of the synthetic results. An instantiation of the ligature stroke is also generated from the trained concatenation style model.

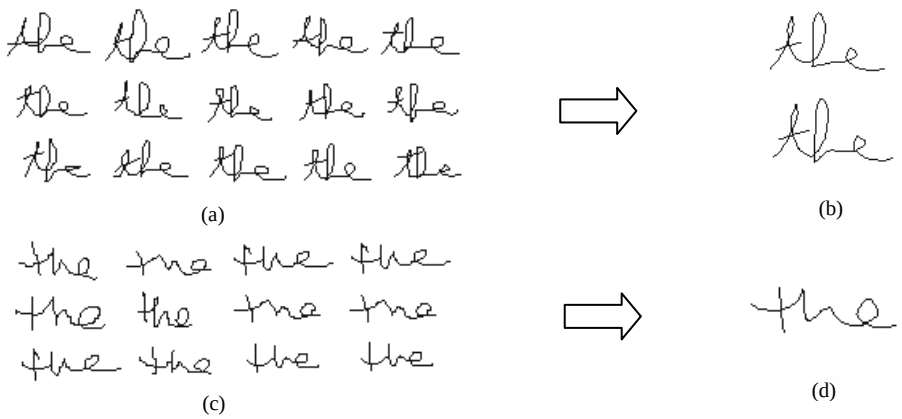


Figure 6. Synthesizing words from samples. (a) Samples of the word "the" written by writer A. (b) Two synthetic words of different styles learned by the system. (c) Samples of the same word written by writer B. (d) One synthetic word of the specific style learned by the system.

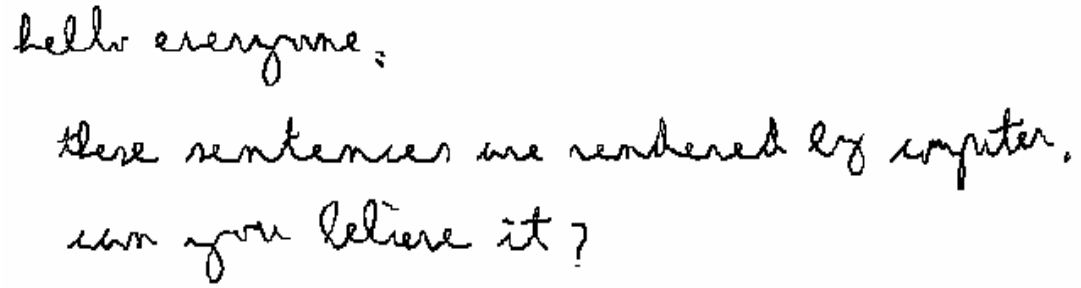


Figure 7. A handwriting note synthesized by the system. Note that our approach can synthesize fluent cursive handwriting words while the glyph-based approach [5] can not.

These generated strokes are integrated together by a deformable model to yield a cursive handwritten AB that is natural and smooth.

4.2 Deformable Models

In the proposed tri-unit handwriting model, there is an overlap between the letter and the concatenation part. The trajectory in the overlap is determined jointly by the letter style model and the concatenation style model. This kind of design guarantees that the letter style and the concatenation style are preserved simultaneously.

A deformable energy of the trained generative model is defined from (5) as

$$E(X) = \frac{1}{2}(X-H)^T \Sigma^{-1}(X-H) + \frac{1}{2}\log|\Sigma| \qquad (11)$$

For a synthetic letter, its main body is generated from the letter style model, but its head part and tail part deserve some deformations to make a smooth and natural connection with neighbors. A concatenating energy and an energy minimization criterion are defined to guide the deformation process.

Three adjacent letters in a cursive handwriting are denoted as X_{i-1}, X_i, X_{i+1}, and their ligature strokes are denoted as $Xc_{i-1,i}$ and $Xc_{i,i+1}$, respectively. Suppose that after concatenation, the head part of the sample X_i is deformed, so the ligature stroke between the letter X_{i-1} and X_i is changed from $Xc_{i-1,i}$ to $Xc^{new}_{i-1,i}$. Similarly, the ligature stroke between the letter X_i and X_{i+1} is changed from $Xc_{i,i+1}$ to $Xc^{new}_{i,i+1}$. The letter X_i is changed to X_i^{new}. The concatenation energy is defined as

$$E_c(X_{i-1}, X_i, X_{i+1}) = \alpha\left[E(Xc^{new}_{i-1,i}) + E(Xc^{new}_{i,i+1})\right] + E(X_i^{new}) \qquad (12)$$

where α is a constant. To create a natural and smooth strait, the head and tail parts of X_i are deformed to minimize the concatenation energy:

$$X_i^{new} = \arg\min E_c(X_{i-1}, X_i, X_{i+1}) \qquad (13)$$

Each letter in a cursive handwriting is deformed in this way and a synthetic, style-preserving handwriting word finally formed.

5. Experimental Results

To demonstrate the effectiveness of the proposed system, some experimental results are shown in this section. Firstly, we focus on a single simple handwritten word. The sample is "the". Because the proposed system is unit-based, a word containing three letters is enough to demonstrate the effectiveness of the system, and there is no difficulty for the system to deal with more complex words containing more letters. We collected several

samples of the word written by two writers and tried to synthesize the same word of the same styles. The results are shown in Fig. 6. It is interesting to note that the first writer has two writing styles of the character "h", that are summarized and learned by the system. The results show that after given some samples, the proposed system can really learn the writing styles of letters and ligature strokes of adjacent letters, and has the ability to generate novel strokes and to concatenate them naturally.

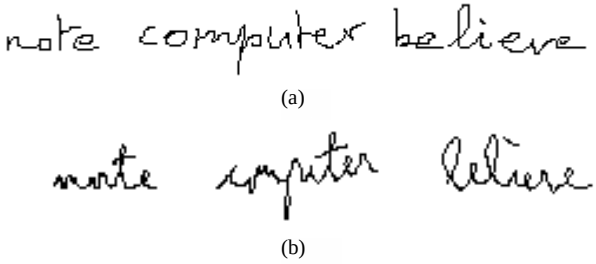
Furthermore, the full collection of models were trained for a specific writer. About 80 different words written by the writer were collected and used for model training. Each lower-case alphabetic letter is included in the training set, which enables the system to synthesize any sentence. Fig. 7 shows a handwritten note synthesized by the system. Compared with the glyph-based technique in [5], our proposed system is more flexible and has the ability of generating the cursive handwriting trajectory of a word fluently even if it does not appear in the training samples. On the contrary, the glyph-based technique cannot synthesize a whole word fluently and can only do it by simply juxtaposing several glyphs in sequence. Some synthetic words using different techniques are compared in Fig. 8.

6. Conclusions and Future Work

Handwriting computing on the post-recognition level is becoming an urgent requirement due to the rapid developments of pen-based devices and systems. In this study the problems of cursive handwriting modeling, learning and synthesis are integrated together and a model-based learning approach has been proposed to synthesize individual cursive handwriting. The ability to learn personal writing style and synthesize novel handwriting of the same style brings handwriting computing to a more intelligent level, and enables computers to help and facilitate a user's work on pen-based devices.

Some experimental results are shown in the paper to illustrate the effectiveness of the system on learning and synthesizing cursive handwriting. However, how to give an objective evaluation on synthetic handwriting is still a problem. Another area to be addressed is that the spline-based system can only handle fluid cursive handwriting and must be coordinated with other techniques to deal with mixed-style handwriting, which may compose of abundant straight lines.

Another issue is that the success of the system heavily depends on the accuracy of the segmentation of handwriting samples used for training. Currently we are improving our segmentation techniques to give more reliable segmentation results. On the other hand, given the fact that segmentation is never perfect, when the system is relegated to end users of pen-based systems, some interactive scenarios must be employed to enable users to manually correct some severe segmentation mistakes in the training data.



(a)

(b)

Figure 8. (a) Some words synthesized in [5]. (b) The same words synthesized by our approach. Note that there are always some intervals in the synthetic words in (a) while the synthetic words in (b) are more fluent.

7. References

[1] R. Plamondon and F.J. Maarse, "An evaluation of motor models of handwriting", *IEEE Trans. PAMI*, 19, pp. 1060-1072, 1989.

[2] W. Guerfali and P. Plamondon, "The Delta LogNormal theory for the generation and modeling of handwriting recognition", *Proc. ICDAR* , pp. 495-498, 1995.

[3] Y. Singer and N. Tishby, "Dynamical encoding of cursive handwriting", *Proc. IEEE Conf. CVPR*,1993.

[4] H. Beigi, "Pre-Processing the Dynamics of On-Line Handwriting Data, Feature Extraction and Recognition", *Proc. The Fifth International Workshop on Frontiers of Handwriting Recognition*, Colchester, England, 1996.

[5] I. Guyon, "Handwriting synthesis from handwritten glyphs", *Porc. the Fifth International Workshop on Frontiers of Handwriting Recognition*, Colchester, England, 1996.

[6] G.L. Cash and M. Hatamian, "Optical character recognition by the method of moments", *Comp. Vis., Graph. and Image Proc.*, 39, pp. 291–310, 1987.

[7] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the neocognition", *IEEE Trans. Neural Networks*, 2, pp. 355–365, 1991.

[8] M. Revow, G.K.I. Williamst and G.E. Hinton, "Using generative models for handwritten digit recognition", *IEEE Trans. PAMI*, 18, pp. 592–606, 1996.

[9] K.W. Cheung, D.Y. Yeung and R.T. Chin, "A Bayesian framework for deformable pattern recognition with application to handwritten character recognition", *IEEE Trans. PAMI*, 20, pp. 1382–1388, 1998.

[10] A.K. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates", *IEEE Trans. PAMI*, 19, pp. 1386–1391, 1997.

[11] E.G. Miller, N.E. Matsakis and P.A. Viola, "Learning from one example through shared densities on transforms", *Proc. IEEE Int'l Conf. CVPR*, Hilton Head, South Carolina, 2000.

[12] A.W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis", *Proc. Eurospeech*, Rhodes, Greece, 1997.