Template-based Synthetic Handwriting Generation for the Training of Recognition Systems

Tamás VARGA, Daniel KILCHHOFER and Horst BUNKE

Department of Computer Science, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{*varga, kilchhof, bunke*}*@iam.unibe.ch*

Abstract. In this paper we present a method for synthesizing English handwritten textlines from ASCII transcriptions. The method is based on templates of characters and the Delta LogNormal model of handwriting generation. To generate a textline, first a static image of the textline is built by concatenating perturbed versions of the character templates. Then strokes and corresponding virtual targets are extracted and randomly perturbed, and finally the textline is drawn using overlapping strokes and delta-lognormal velocity profiles in accordance with the Delta LogNormal theory. The generated textlines are used as training data for a hidden Markov model based off-line handwritten textline recognizer. First results show that adding such generated textlines to the natural training set may be beneficial.

Keywords: off-line handwriting recognition, handwriting generation, Delta LogNormal model, hidden Markov model (HMM)

1. Introduction

Handwriting recognition has been an intensive research field for more than 40 years (Bunke, 2003). The most successful applications include isolated character recognition, automatic address processing, and bank check reading. However, for the problem of general word and sentence recognition, where no task specific constraints exist and a large lexicon is considered, the recognition rates are still rather low. Besides the recognition algorithm and the types of features used, the amount and quality of training data also has a great impact on the recognition performance of handwriting recognition systems. As a rule of thumb says, the classifier trained on the most data wins. This has been experimentally justified in many works before (Cano & al., 2002; Rowley & al., 2002; Velek & al., 2002). The most straightforward way to expand the training set would be to collect additional natural, i.e. human written, samples. But collecting human written texts is a rather expensive and time consuming process. Alternatively, the training set can be expanded by synthetic, i.e. computer generated, texts.

Several methods have been reported in the literature for synthetic text generation. Most of them generate the synthetic texts from existing natural ones. In (Guyon, 1996) and (Helmers & al., 2003), human written character tuples are used to build up synthetic texts. For each writer, the tuples of characters can either be collected separately (Guyon, 1996), or extracted from textlines written by that specific writer (Helmers & al., 2003). In other approaches, synthetic texts are generated by applying random perturbations on human written characters (Cano & al., 2002; Drucker & al., 1993; Mori & al., 2000), words (Setlur & al., 1994), or whole lines of text (Varga & al., 2004). This has proved to be a very efficient way to increase the variability of the generated set of synthetic texts, yielding improvements of the recognition rate when used as additional training data to the handwriting recognition system (Cano & al., 2002; Drucker & al., 1993; Mori & al., 2000; Varga & al., 2004).

However, generating synthetic texts does not necessarily require to use human written texts as a basis. In (Lee & al., 1998), Korean characters are synthesized using templates of characters, and a handwriting generation model. The templates consist of strokes of predefined writing order. After the geometrical perturbation of a template, beta curvilinear velocity and pen-lifting profiles are generated for the (overlapping) strokes. Finally, the character is drawn using the generated velocity and pen-lifting profiles (see (Lee & al., 1998) for details). Although the generated characters look natural, they were not used to train a recognizer.

In the present paper, we follow a similar approach for the generation of English handwritten textlines. That is, we manually build templates (or prototypes) of handwritten characters using Bézier splines (Hearn & al., 1986). As a first step to generate a textline image corresponding to a given ASCII transcription, the appropriate series of character templates are perturbed and concatenated. The resulting static image is then decomposed into strokes of straight segments and circular arcs (Li & al., 1998). These strokes are then randomly expanded and overlapped in time. For each stroke, a delta-lognormal curvilinear velocity profile is generated, and the skeleton image of the textline is drawn, followed by grayscale thickening operations to make the generated script look more natural. The delta-lognormal velocity profiles as well as the decomposition into straight and circular strokes are taken from the Delta

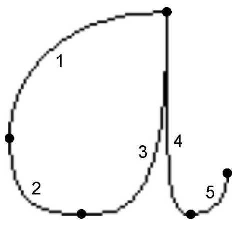

Figure 1. Template of letter 'a', consisting of 5 consecutive Bézier arcs.

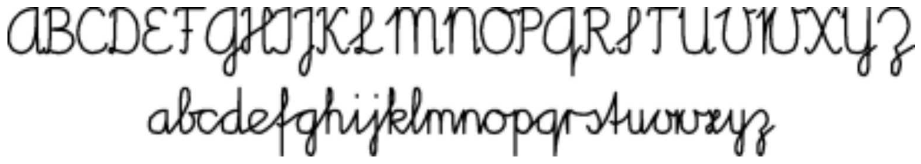

Figure 2. Prototypes for the cursive style English alphabet.

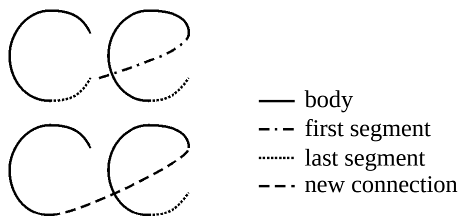

Figure 3. Linking letter 'c' to letter 'e'.

LogNormal theory of handwriting generation (Plamondon & al., 1998). The advantage of using a handwriting generation model is that the variations resulting from perturbing its parameters may better reflect psychophysical phenomena of human handwriting than applying geometric ad-hoc distortions on a static image. Furthermore, this way we can get online data, not only static images of texts.

The textlines generated by the proposed method are used to train a hidden Markov model (HMM) based off-line handwritten textline recognizer. The aim of our work is to examine how the synthesized training sets compare to the natural ones, in terms of the recognition rate.

The paper is organized as follows. In Section 2, the method to generate static images from an ASCII transcription is briefly presented. Section 3 describes how to apply concepts of the Delta LogNormal theory to synthesize more natural looking scripts. First experimental results are given in Section 4. Finally, discussion of the results and plans for future work are provided in Section 5.

2. Character Templates and their Concatenation

We have built templates for the upper and lower case characters of the English alphabet, for the ten digits, and for many special symbols including punctuations. Each prototype is a Bézier spline (Hearn & al., 1986), which was put together manually from a series of Bézier arcs, one after the other, in a predefined writing order. An example is shown in Fig. 1, where a template for the lower case letter 'a' can be seen. The template consists of 5 Bézier arcs. In general, the start and end point of the letter, the points where the tangent is horizontal or vertical, and the corner points where the tangent is ambiguous, separate the Bézier arc segments of the template from each other. That is, those points correspond to the start and end points of the individual Bézier arcs.

In this work, we have used two versions of the English alphabet prototypes: block and cursive style. They set of cursive prototypes are shown in Fig. 2. Cursive writing style means that a character is linked to the preceding one. The linking of two consecutive characters is usually done by removing the last and the first segment of the first and the second character, respectively, and connecting the new end point of the first character to the new start point of the second one, using a Bézier arc. An example of linking letter 'c' to 'e' is shown in Fig. 3.

Now suppose that we are given an ASCII transcription of a textline. The first step towards the generation of a corresponding image of that textline is that we transform the ASCII character sequence (including spaces) into a sequence of template characters using the available prototypes. If a character class has more than one prototype available (e.g. block and cursive style), one of them is chosen according to a random parameter.

Next, each template in the series undergoes some geometrical perturbations, including scaling, shifting, and changing the slant. Since the character templates are defined by the control points of the

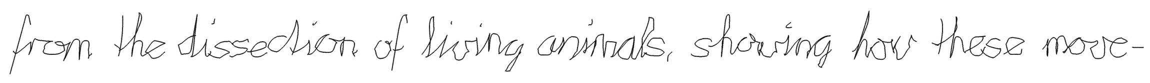
from the dissection of living animals, showing how these move-

Figure 4. Static image after the perturbation and concatenation of character templates.

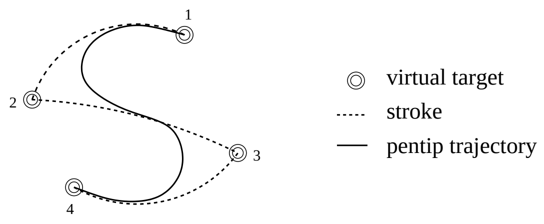

Figure 5. Action plan for letter 's' – virtual targets are linked by circular strokes.

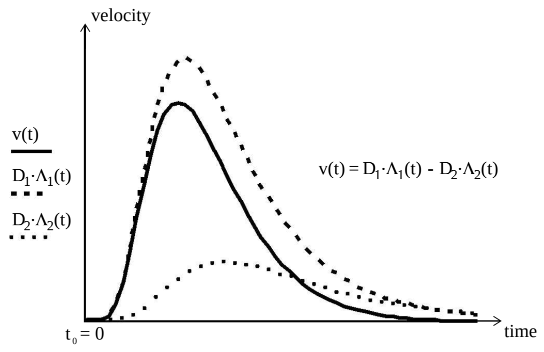

Figure 6. Calculation of curvilinear velocity for a single stroke.

Bézier spline, it is sufficient to perform the perturbations only on the control points. The perturbation method also includes shifting all control points using randomly chosen displacement vectors.

We get the static image of the textline by concatenating the perturbed templates along a horizontal baseline. This means that the perturbed templates are put one after the other and cursive style letters are linked. An example can be seen in Fig. 4. Many of the curves in this figure seem to contain too much noise. However, reducing the level of noise seriously reduces the diversity of character shapes.

To alleviate this problem, in the next section we incorporate elements of a handwriting generation model into the synthesizing process. This enables us to generate naturally looking handwriting with high character shape variability while keeping the noise level low.

3. Handwriting Generation Model

The Delta LogNormal model of handwriting generation describes fluent handwriting as the vectorial superimposition of consecutive strokes in time (Plamondon & al., 1998). The strokes of a letter or a word can either be straight line segments or circular arcs, and they are part of a so-called action plan connecting a sequence of virtual targets. Such an action plan regarding letter 's', for example, is depicted in Fig. 5.

Each stroke results from an action of rapid writing, produced by the synergy of two parallel neuromuscular systems, one being the agonist and the other the antagonist to the movement, resulting in a delta-lognormal velocity profile of the pentip (Plamondon, 1995):

$$v(t) = D_1 \cdot \Lambda_1(t, t_0, \mu_1, \sigma_1^2) - D_2 \cdot \Lambda_2(t, t_0, \mu_2, \sigma_2^2)$$

where $\Lambda_i(t, t_0, \mu_i, \sigma_i^2)$ $(i = 1, 2)$ is a lognormal function:

$$\Lambda_i(t, t_0, \mu_i, \sigma_i^2) = \frac{1}{\sigma_i \sqrt{2\pi} \cdot (t - t_0)} \cdot e^{-\frac{(ln(t-t_0)-\mu_i)^2}{2\sigma_i^2}}$$

D_i denotes the amplitude of the input command to the neuromuscular system, occurring at time t_0, while μ_i and σ_i are called the logtime delay and the logresponse time, respectively (see (Plamondon & al., 2003) for a detailed discussion on the meaning of these parameters).

An illustration of the curvilinear velocity calculation is shown in Fig. 6. Since the total integral of the lognormal function is equal to 1, the stroke length is given by $D = D_1 - D_2$. However, since the velocity curve is unbounded with respect to time ($t \to \infty$), in practice the stroke is truncated at some time, t', where $v(t')$ is close to zero.

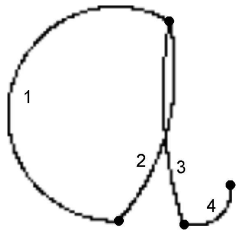

Figure 7. Approximation of letter 'a' by circular arc segments.

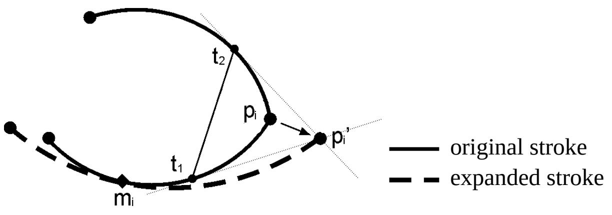

Figure 8. Illustration of the stroke expansion method.

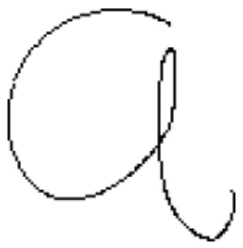

Figure 9. Letter 'a' generated using delta-lognormal velocity profiles and stroke overlapping.

The execution of two or more consecutive strokes can partially overlap in time, which means that a stroke can be initiated before the preceding one reaches its target (Plamondon, 1995). If, say n, strokes overlap at time t, the resulting velocity vector of the pentip is obtained as the vectorial summation of the n individual velocity vectors. This temporal superimposition of discontinuous strokes results in a continuous trajectory of the pentip, as for the example depicted in Fig. 5. It also can be seen that the virtual targets are usually never reached (except for the very first and the very last one), so the actual trace of the pentip differs from that of the action plan.

As a first step towards the incorporation of the Delta Lognormal model in our method, the static image described in Section 2 is approximated by segments of straight lines and circular arcs, without overlapping (Li & al., 1998). We consider the following points as segmentation points: start or end point of a component, local maxima of curvature being greater than a predefined threshold, inflection points, and corner points. These points can be easily extracted from the Bézier spline. In Fig. 7, such an approximation of letter 'a' of Fig. 1 can be seen.

This kind of approximation can already be considered as a simple action plan, where the segmentation points correspond to the virtual targets, and the segments to the strokes of the action plan. However, in real handwriting the action plan is usually of greater extent than the actual trace of the writing, due to the stroke overlapping effect (see the example in Fig. 5). That's why we apply a stroke expansion method on the simple action plan. The idea of the method is illustrated in Fig. 8. Here the i-th and the $(i+1)$-th strokes join together at virtual target p_i. We choose randomly two points, t_1 and t_2, on either side of p_i. The new virtual target, p_i', is the intersection of the tangents at t_1 and t_2. Finally, the expanded i-th stroke (dashed line in Fig. 8) is a circular arc intersecting the original i-th stroke at some point, m_i (this condition ensures that the expansion will not be too large).

After we have expanded the action plan, delta-lognormal velocity profiles are generated for each stroke, and the degree of overlapping is decided for each consecutive pair of strokes, using randomly perturbed parameters of the Delta LogNormal model (see (Guerfali & al., 1995) for an analysis on the effect of various parameter values). Based on this online information, the textline can be generated. In Fig. 9, a generated version of letter 'a' of Fig. 1 can be seen. Here we note that a complete textline is drawn component by component, i.e. pen lifting is not modeled in this work.

To make the textline look more realistic, thickening operations are also applied, using gradually decreasing gray level values. Some examples of the final result of our textline generation method are shown in Fig. 10. As it can be seen, the curves are smoother compared to Fig. 4, due to the overlapping effect.

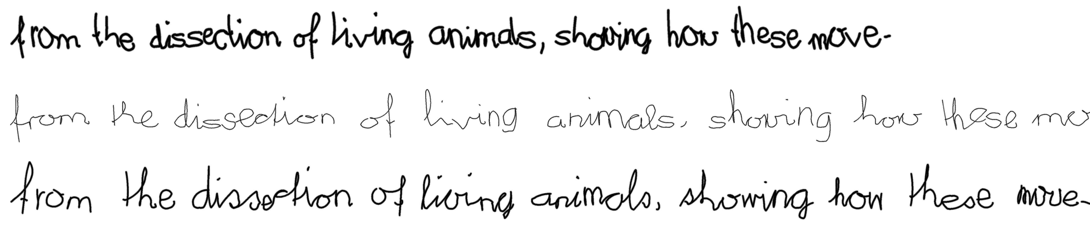
from the dissection of living animals, showing how these move-
from the dissection of living animals, showing how these move-
from the dissection of living animals, showing how these move-

Figure 10. Examples of complete textlines generated using delta-lognormal velocity profiles and stroke overlapping.

Table 1. Recognition rates on the test set, using different types of training sets.

Training set	Recognition rate (in %)	Capacity (Ga)
natural only (Tr_1)	70.81	6
synthetic only (Tr_2)	61.78	21
natural + 1 synthetic (Tr_3)	72.30	9
natural + 5 synthetic (Tr_4)	70.06	27

4. Experimental Results

In this section, first results of our on-going experiments using synthesized handwriting for training a recognizer are provided.

The application considered in this paper is the off-line recognition of handwritten textlines. The recognizer used is the hidden Markov model (HMM) based handwritten textline recognizer described in (Marti & al., 2001). The recognizer takes, as a basic input unit, a complete line of text. The output is a sequence of words. For each character, a continuous HMM with linear topology is provided, and the character HMMs are concatenated to represent words and lines. In the emitting states of all HMMs, the observation probability distributions are estimated by using a number, Ga, of Gaussian mixture components.

For the experiments, subsets of the IAM-Database are used (Marti & al., 2002). We defined a training set of 320 textlines from 64 writers, and a test set of 160 textlines from 32 writers. The writers of the training and the test set are disjoint, i.e. the experiments are writer-independent. The underlying vocabulary consists of $6,012$ words. For each of the 320 natural textlines in the training set, 5 synthetic ones were generated, using the ASCII transcription of the corresponding natural textline. This means that altogether 320 natural and $320 \cdot 5 = 1,600$ synthetic textlines are available. The recognition rate is measured on the word level, using the natural textlines of the test set for testing.

In the experiments, we investigate the effects of two factors. The first is the proportion of natural and synthetic textlines in the training data. For this purpose, four different training sets were produced from the pool of available training textlines: $Tr_1 = \{$all 320 natural textlines$\}$, $Tr_2 = \{$all $1,600$ synthetic textlines$\}$, $Tr_3 = \{$320 natural + 320 synthetic textlines$\}$, $Tr_4 = \{$320 natural + $1,600$ synthetic textlines$\}$. The second factor is the capacity of the recognition system, i.e. the number of free parameters to be estimated, which has turned out to be important when using synthetic training data (Varga & al., 2004). We control the capacity by varying the value of Ga.

In Table 1, the best results using Tr_1, Tr_2, Tr_3, and Tr_4 for training are shown, together with the corresponding capacity values. Although these are our first results on limited data sets, some conclusions can already be drawn. First of all, the natural training set performs much better than the training set consisting exclusively of synthesized textlines. Nevertheless, the experiments also suggest that augmenting the natural training set by synthetic textlines can improve the recognition rate, but the proportion of the natural and synthetic data is an important factor. Only with set Tr_3, where the amount of natural and synthetic training data are the same, an improvement was observed. In case of Tr_2 and Tr_4, a recognition rate lower than that for Tr_1 is achieved. The experiments reported in Table 1 confirm our earlier results (Varga & al., 2004), which show that increasing the capacity of the system is beneficial when the training set is expanded by synthetic textlines. This is because in case of too low capacity the system is biased towards the unnatural variability introduced by the synthetic training data.

5. Discussion and Future Work

A method for synthesizing handwritten textlines from ASCII transcriptions was presented. The method is based on character templates and the Delta LogNormal handwriting generation model. Our aim is to use synthetic textlines for the training of a handwritten textline recognizer. The experimental results show that the natural training set performs much better than the one that contains exclusively synthesized textlines. On the other hand, the addition of synthetic textlines to the natural training set may improve the recognition rate, but the proportion of natural and synthetic textlines plays an important role.

These results suggest that additional work is conducted in the future to make the synthetic textlines represent more diverse writing styles. For example, increasing the number of templates per character would be a straightforward step towards this direction. Furthermore, since the recognizer uses grayscale information of the images, improving the grayscale thickening operations to better reflect the characteristics of natural textline images can also be beneficial.

Regarding the generation of online data, we plan to refine the stroke expansion scheme, as well as its relation with the degree of overlapping. Additionally, to generate online data not only on the component level but also on the textline level, modeling of the pen-lifting movements is needed.

Acknowledgments
This work was supported by the Swiss National Science Foundation program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Scene Analysis", as part of NCCR. The authors also thank Urs-Viktor Marti and Matthias Zimmermann for providing the recognition system and the IAM-Database.

References

Bunke, H. (2003). Recognition of Cursive Roman Handwriting – Past, Present and Future. Proc. 7th Int. Conf. on Document Analysis and Recognition, pp. 448–461.

Cano, J., Pérez-Cortes, J.C., Arlandis, J. & Llobet, R. (2002). Training Set Expansion in Handwritten Character Recognition. Proc. 9th SSPR / 4th SPR, pp. 548–556.

Drucker, H., Schapire, R. & Simard, P. (1993). Improving Performance in Neural Networks Using a Boosting Algorithm. In Advances in Neural Information Processing Systems 5, Hanson S. & al. eds, Morgan Kaufmann, San Mateo CA, pp. 42–49.

Guerfali, W. & Plamondon, R. (1995). The Delta LogNormal Theory for the Generation and Modeling of Cursive Characters. Proc. 3rd Int. Conf. on Document Analysis and Recognition, pp. 495–498.

Guyon, I. (1996). Handwriting Synthesis from Handwritten Glyphs. Proc. 5th Int. Workshop on Frontiers in Handwriting Recognition, pp. 309–312.

Hearn, D. & Baker, M.P. (1986). Computer Graphics. Prentice-Hall, New Jersey.

Helmers, M. & Bunke, H. (2003). Generation and Use of Synthetic Training Data in Cursive Handwriting Recognition. Proc. 1st Iberian Conf. on Pattern Recognition and Image Analysis, pp. 336–345.

Lee, D.-H. & Cho, H.-G. (1998). A New Synthesizing Method for Handwriting Korean Scripts. Int. Journal of Pattern Recognition and Artifical Intelligence, 12(1), pp. 46–61.

Li, X., Parizeau, M. & Plamondon, R. (1998). Segmentation and Reconstruction of On-line Handwritten Scripts. Pattern Recognition, 31(6), pp. 675–684.

Marti, U.-V. & Bunke, H. (2001). Using a Statistical Language Model to Improve the Performance of an HMM-based Cursive Handwriting Recognition System. Int. Journal of Pattern Recognition and Artifical Intelligence, 15(1), pp. 65–90.

Marti, U.-V. & Bunke, H. (2002). The IAM-Database: an English Sentence Database for Off-line Handwriting Recognition. Int. Journal on Document Analysis and Recognition, 5(1), pp. 39–46.

Mori, M., Suzuki, A., Shio, A. & Ohtsuka, S. (2000). Generating New Samples from Handwritten Numerals based on Point Correspondence. Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition, pp. 281–290.

Plamondon, R. (1995). A Kinematic Theory of Rapid Human Movements. Part I: Movement Representation and Generation. Biological Cybernetics, 72, pp. 295–307.

Plamondon, R., Feng, C. & Woch, A. (2003). A Kinematic Theory of Rapid Human Movement. Part IV: a Formal Mathematical Proof and New Insights. Biological Cybernetics, 89, pp. 126–138.

Plamondon, R. & Guerfali, W. (1998). The Generation of Handwriting with Delta-lognormal Synergies. Biological Cybernetics, 78, pp. 119–132.

Rowley, H. A., Goyal, M. & Bennett, J. (2002). The Effect of Large Training Set Sizes on Online Japanese Kanji and English Cursive Recognizers. Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition, pp. 36–40.

Setlur, S. & Govindaraju, V. (1994). Generating Manifold Samples from a Handwritten Word. Pattern Recognition Letters, 15, pp. 901–905.

Varga, T. & Bunke, H. (2004). Off-line Handwriting Recognition Using Synthetic Training Data Produced by means of a Geometrical Distortion Model. Int. Journal of Pattern Recognition and Artifical Intelligence, 18(7), pp. 1285–1302.

Velek, O. & Nakagawa, M. (2002). The Impact of Large Training Sets on the Recognition Rate of Off-line Japanese Kanji Character Classifiers. Proc. 5th Int. Workshop on Document Analysis Systems, pp. 106–109.