Generation of Synthetic Training Data for an HMM-based Handwriting Recognition System

Tamás Varga and Horst Bunke
Institut für Informatik und angewandte Mathematik
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
email: {varga,bunke}@iam.unibe.ch

Abstract

A perturbation model for generating synthetic textlines from existing cursively handwritten lines of text produced by human writers is presented. Our purpose is to improve the performance of an HMM-based off-line cursive handwriting recognition system by providing it with additional synthetic training data. Two kinds of perturbations are applied, geometrical transformations and thinning/thickening operations. The proposed perturbation model is evaluated under different experimental conditions.

1. Introduction

In the past decade, researchers in the field of pattern recognition realized that the performance of a recognition system does not only depend on the underlying features and classification algorithms, but is also strongly affected by the size and quality of the training data [1]. In the area of handwriting recognition, some experiments have been conducted to examine under what circumstances a larger training set improves the accuracy of handwriting recognition systems. These experiments were inspired by the well known rule of thumb saying that the classifier wins that is trained on the most data. In [8], the effect of increasing the training set size beyond those generally available is analyzed for online character and word recognition problems. It is shown that an increasing amount of training data improves the accuracy, even when the recognizer's representational power is (not too severely) limited. In those experiments only natural, i.e. human written training data was used. However, most researchers don't have the possibility to arbitrarily enlarge their training sets. One promising way to overcome this dilemma is to use synthetic data. The synthetic generation of new samples can be achieved in many different ways, e.g. through perturbation of, or interpolation between, the original samples. Examples where additional synthetic training data was successfully used for the recognition of isolated characters have been reported in [2, 7]. Apart from augmenting a training set, perturbation approaches can be applied in the recognition phase, making the recognizer insensitive to small transformations or distortions of the image to be recognized. Examples from the field of isolated character recognition are [3] and [9].

However, to the knowledge of the authors, for the problem of general, off-line cursive handwritten word and text recognition, no similar procedures involving synthetically generated text images have been reported. In this paper we present a perturbation model to generate synthetic textlines from existing cursive handwritten text. The motivation is to add synthetic data to the natural training data so as to enlarge the training set. The basic idea of the approach proposed in this paper is to use continuous nonlinear functions that control a class of geometrical transformations. The functions ensure that the distortions performed cannot be reversed by standard preprocessing operations of the handwriting recognition system. For the experiments, the HMM-based sentence recognizer described in [5], and subsets of the IAM Database introduced in [6] were used. The model proposed in this paper was evaluated under several experimental conditions.

The paper is organized as follows. Section 2 describes the perturbation model. In Section 3 experimental results are given. Finally, Section 4 presents some conclusions and a plan for future work.

2. Perturbation model

In this section, a perturbation model for the distortion of cursive handwritten textlines is presented. The model incorporates some parameters over a range of possible values, from which a random value is picked each time before distorting a textline. There is a constraint on the textlines to be distorted: they have to be skew and slant corrected, because of the nature of the applied geometrical transforma-

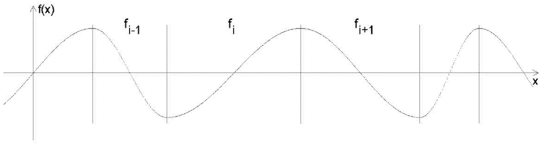

Figure 1. An example of a CosineWave function.

tions. This constraint is not severe, because skew and slant correction are very common preprocessing steps found in almost any handwriting recognition system. In the following subsections we describe the model in greater detail.

2.1. Underlying functions and their properties

Each geometrical transformation in the model is controlled by a continuous nonlinear function, which determines the strength of the considered transformation at each horizontal or vertical coordinate of the textline or connected component. These functions will be called underlying functions.

The underlying functions are built from a simple function called *CosineWave*. A CosineWave is the concatenation of n functions, $f_1, f_2, \ldots, f_n$, where $f_i : [0, l_i] \rightarrow \Re$, $f_i(x) = (-1)^i \cdot a \cdot \cos(\frac{\pi}{l_i} \cdot x),\ l_i > 0$.

An example is shown in Fig. 1. The functions f_i (separated by vertical line segments in Fig. 1) are called *components*. The *length* of component f_i is l_i and its *amplitude* is $|a|$. The amplitude doesn't depend on i, i.e. it is the same for all the components.

To randomly generate a CosineWave instance, three ranges of parameters need to be defined:

- $[a_{min}, a_{max}]$ for the amplitude $|a|$,
- $[l_{min}, l_{max}]$ for the component length,
- $[x_{min}, x_{max}]$ for the interval to be covered by the components.

The generation of a CosineWave is based on the following steps. First the amplitude is selected by picking a value $\alpha \in [a_{min}, a_{max}]$ randomly and letting $a = \alpha$ or $a = -\alpha$ with a 50% probability each. Then l_1 is decided by randomly picking a possible value from $[l_{min}, l_{max}]$. Finally the beginning of the first component (i.e. f_1) is chosen randomly from the $[x_{min} - l_1, x_{min}]$ interval. From this point on we only have to add additional components, one after the other, with randomly chosen lengths, until x_{max} is reached.

An underlying function is obtained by summing up a number, m, of such CosineWave functions.

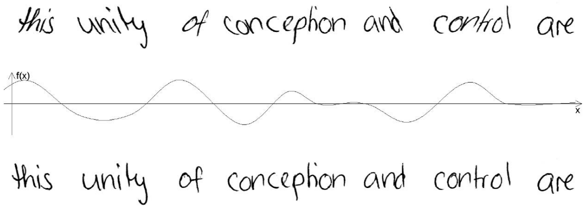
this unity of conception and control are
f(x)
x
this unity of conception and control are

Figure 2. Illustration of shearing

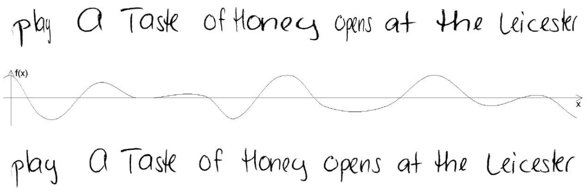
play A Taste of Honey opens at the Leicester
f(x)
x
play A Taste of Honey opens at the Leicester

Figure 3. Illustration of horizontal scaling

2.2. Geometrical transformations

The underlying functions introduced in Section 2.1 control two kinds of geometrical transformations described in the following. The first transformation is applied to a whole line of text, and the second to the connected components of the considered line of text. The underlying function of each transformation is randomly generated, as described in 2.1. The parameters x_{min} and x_{max} are always determined by the actual size of the image to be distorted.

In the following the geometrical transformations will be defined and illustrated by figures. Note that the figures are only for illustration purposes, and weaker instances of the distortions are used in the experiments described in Section 3.

2.2.1. Line level transformations

There are four classes of such transformations. Their purpose is to change properties such as slant, horizontal and vertical size, and the position of characters with respect to the baseline.

Shearing: Its underlying function defines the tangent of the shearing angle for each x coordinate. Shearing is performed with respect to the lower baseline. An example is shown in Fig. 2. In this example and the following ones, the original textline is shown at the bottom, the underlying function in the middle, and the result of the distortion on top.

Horizontal scaling: Here the underlying function defines the horizontal scaling factor for each x coordinate. This transformation is performed through horizontal shifting of the pixel columns. An example of this operation is

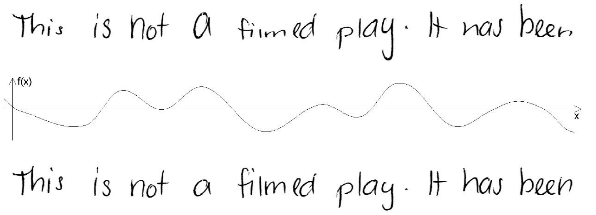
This is not a filmed play. It has been
f(x)
x
This is not a filmed play. It has been

Figure 4. Illustration of vertical scaling

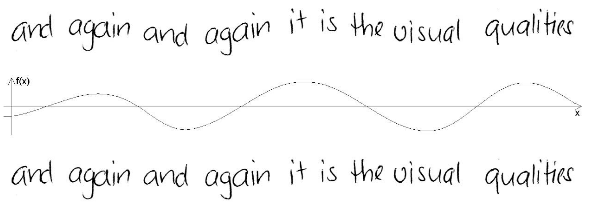
and again and again it is the visual qualities
f(x)
x
and again and again it is the visual qualities

Figure 5. Illustration of baseline bending

shown in Fig. 3.

Vertical scaling: The underlying function determines the vertical scaling factor for each x coordinate. Scaling is performed with respect to the lower baseline. An example is shown in Fig. 4.

Baseline bending: This operation shifts the pixel columns in vertical direction according to the value of the underlying function for each x coordinate. An example is shown in Fig. 5.

2.2.2. Connected component level

Contrary to line level transformations, which aim at distortions at a rather global level, the perturbations discussed in this sub-section intend to change the structure of the writing within individual characters. After having applied all the desired connected component level transformations, the connected component is scaled in both horizontal and vertical direction, so that its bounding box regains its original size. These transformations are quite similar to those at the line level, but here the underlying functions relate to the actual connected component. For each connected component individual underlying functions are generated.

Horizontal scaling: This transformation is identical with horizontal scaling as described in sub-section 2.2.1, but it is applied to individual connected components rather than whole lines of text.

Vertical scaling 1: This is the counterpart of horizontal scaling in the vertical direction.

Vertical scaling 2: This transformation is identical with vertical scaling as described in sub-section 2.2.1, except that scaling is performed with respect to the horizontal middle-line of the bounding box.

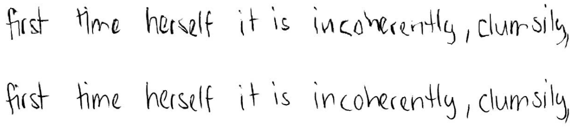
first time herself it is incoherently, clumsily,
first time herself it is incoherently, clumsily,

Figure 6. An illustration of connected component level distortions

The effect of all three transformations applied one after the other is shown in Fig. 6. In this figure, the lower textline is the original one, and above its distorted version is displayed. One must observe that in spite of the distortions the connected components underwent, their bounding boxes have remained the same.

2.3 Thinning/thickening operations

The appearance of a textline can also be changed by varying the thickness of its strokes. In the present perturbation model this is done by applying thinning or thickening steps iteratively. The method is based on a grey scale variant of the MB2 thinning algorithm [4]. (A general way to get the grey scale version of a specific type of thinning algorithm operating on binary images can be found in [10]). Thinning and thickening could also be performed using the morphological erosion and dilation operators, respectively, but this wouldn't be safe when applied iteratively, because part of the original writing might be lost after too many steps of erosions.

2.4 Distorted textline generation

Now that the main constituents of the perturbation model have been introduced, a simple scheme is provided so that whole textlines can be distorted. The steps of distorting a given skew and slant corrected textline are the following:

1. Apply each of the line level transformations to the textline, one after the other, in the order given in sub-section 2.2.1.

2. For each individual connected component, apply the transformations given in sub-section 2.2.2. After the transformations have been applied, scale the resulting connected components so that their bounding boxes regain their original sizes, and place them in the image exactly at their original locations.

3. Apply thinning/thickening operations.

Of course, these steps are not required to be always rigorously followed. In particular, one can omit one or several of these transformations.

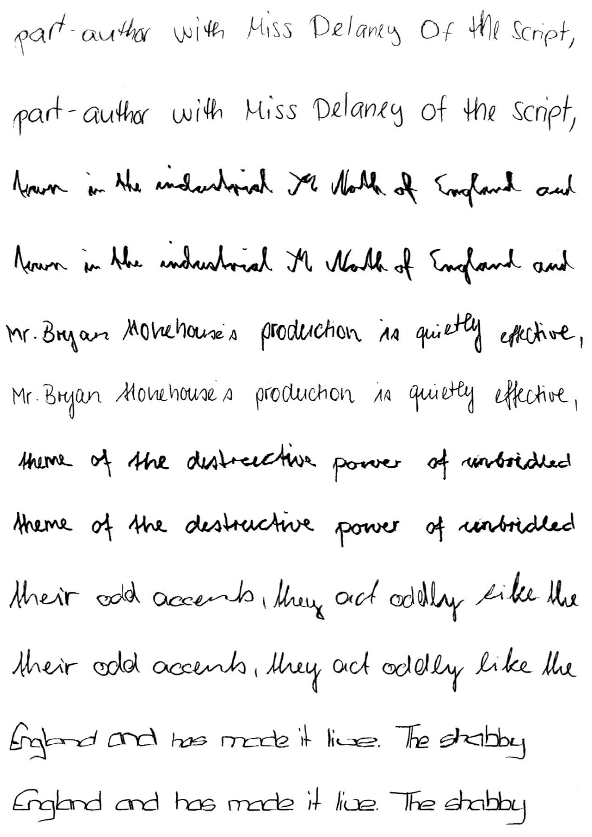
part-author with Miss Delaney Of the Script,
part-author with Miss Delaney of the script,
born in the industrial N North of England and
born in the industrial N North of England and
Mr. Bryan Morehouse's production is quietly effective,
Mr. Bryan Morehouse's production is quietly effective,
theme of the destructive power of unbridled
theme of the destructive power of unbridled
their odd accents, they act oddly like the
their odd accents, they act oddly like the
England and has made it live. The shabby
England and has made it live. The shabby

Figure 7. Natural and synthetic textlines for writers a-f

3 Experimental results

The purpose of the experiments described in the following is to investigate whether the performance of a handwritten text recognizer can be improved by adding synthetically generated texts to the training data.

The recognizer used in this work is the HMM-based cursive handwritten sentence recognizer described in [5]. The recognizer takes, as basic input unit, a complete line of text. That is, no segmentation of the text into isolated words is required. The output is a sequence of words. In the experiments described in the following the recognition rate will always be measured on the word level.

For the experiments, a subset of the IAM Database [6], containing 541 textlines from 6 different writers, was used. These textlines include a total number of 3899 word instances from a vocabulary of 412 words. The six writers who produced the data used in the experiments will be denoted by *a*, *b*, *c*, *d*, *e* and*f* in the following. Subsets of writers will be represented by sequences of these letters. For example, *abc* stands for writers *a*, *b*, and *c*.

All the experiments were writer independent, where the writers who contributed to the training set were disjoint from the writers who produced the test set. This makes the task of the recognizer extremely hard, because the writing styles found in the training set can be totally different from those of the test set.

Three experiments were conducted, in which the textlines of the training sets were distorted by applying three different subsets of the distortions described in Section 2. The three subsets were the set of all distortions, the set of line level geometrical transformations, and the set of connected component level geometrical transformations. In each case, five distorted textlines per given training textline were generated and added to the training set. So the extended training set was six times larger than the original one. The recognition results of the three experiments are shown in Table 1, where the rows correspond to the different training modalities. The test set is always the complement of the training set, and consists of natural text only. For example, the test set corresponding to the first row consists of all natural textlines written by writers *bcdef*, while the training set is given by all natural textlines produced by writer *a* plus five distorted instances of each natural textline. In the first column, the results achieved by the original system that uses only natural training data are given for the purpose of reference. The other columns contain the results of the three experiments using expanded training sets. In those three columns each number corresponds to the median recognition rate of three independent experimental runs. In each run a different recognition rate is usually obtained because of the random nature of the distortion procedure. Fig. 7 shows examples of natural and synthetically generated textline pairs used in the experiment where all the distortions were applied. For each pair of textlines the natural one is shown below, while the synthetic one is above it. The first pair belongs to writer *a*, the second to writer *b*, and so on.

In Table 1 it can be observed that adding synthetic training data leads to an improvement of the recognition rate in 29 out of 33 cases. Some of the improvements are quite substantial, for example, the improvement from 33.1% to 49% in row *a*.

Augmenting the training set of a handwriting recognition system by synthetic data as proposed in this paper may have two adversarial effects on the recognition rate. First, adding synthetic data increases the variability of the training set, which may be beneficial when the original training set has a low variability, i.e. when it was produced by only one or a few writers. On the other hand, the distortions may produce unnaturally looking words and characters, which may bias the recognizer in an undesired way, because the test set includes only natural handwriting.

The greatest increase in recognition performance can be

Table 1. Results of the experiments (in %)

	original	all dist.	line level	cc. level
a	33.1	49.0	47.1	38.7
b	38.7	43.1	40.4	42.6
c	39.2	49.31	46.8	44.4
d	30.6	53.1	48.6	43.0
e	54.4	59.6	58.9	54.2
f	18.8	32.0	26.9	27.8
ab	60.7	73.5	75.8	54.9
cd	56.8	61.3	62.4	59.6
ef	63.8	68.5	67.5	67.5
abc	75.2	74.1	75.8	74.8
def	65.3	68.9	67.0	68.7

observed in Table 1 for those cases where there is only one writer in the training set. Then the variability of the training set is low and the addition of synthetic data leads to a better modeling of the test set. In this case, the application of all distortions outperforms the use of only line level or connected component level distortions. Where more than one writer are used for training, the variability of the training set is larger and the increase in recognition performance becomes smaller when synthetic training data is added.

To compare the performance of the different subsets of distortions (all, line level, cc level) with each other, one can simply count the number of cases in Table 1 where an improvement was observed. The resulting numbers are 10, 11 and 8 for all, line level, and cc level, respectively (corresponding to columns 2-4). So, based on this kind of comparison, line level distortions perform best in the experiments reported in this paper.

4 Conclusions and future work

In this paper a perturbation model for generating synthetic textlines from existing cursive handwritten textlines was presented. The purpose was to investigate whether the performance of a handwritten text recognizer can be improved by adding synthetically generated texts to the training data. Experiments were conducted in which different subsets of distortions were applied to generate synthetic data. In the majority of all experimental runs, an improvement of the recognition rate was observed. In some cases the increase was quite substantial. Hence the use of synthetic training data can potentially lead to improved handwriting recognition systems.

In the future, experiments will be conducted on a larger scale, i.e. using larger subsets of the IAM Database. Furthermore, the effects of all individual distortions (shearing, scaling, baseline bending a.s.o) will be investigated in greater detail. The perturbation model described in this paper may also have other possible applications. For example, using it in the testing phase can perhaps help to identify potential weaknesses of handwriting recognition systems.

Acknowledgements

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 U.-V. Marti and Matthias Zimmermann for providing the recognition system and the IAM Database, and Simon Günter for his valuable comments and advices.

References

[1] H. Baird. State of the art of document image degradation modeling. In *Proc. 4th IAPR Workshop on Document Analysis Systems (DAS 2000)*, Invited plenary talk, Rio de Janeiro, Brasil, December 2000.

[2] J. Cano, J. Pérez-Cortes, J. Arlandis, and R. Llobet. Training Set Expansion in Handwritten Character Recognition. In *Proc. 9th SSPR / 4th SPR*, pages 548–556, Windsor, Ontario, Canada, 2002.

[3] T. Ha and H. Bunke. Off-line handwritten numeral recognition by perturbation method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):535–539, May 1997.

[4] A. Manzanera and T. Bernard. Improved Low Complexity Fully Parallel Thinning Algorithm. In *Proc. 10th Int. Conf. on Image Analysis and Processing*, pages 215–220, Venice, Italy, 1999.

[5] U.-V. Marti and H. Bunke. 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):65–90, 2001.

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

[7] M. Mori, A. Suzuki, A. Shio, and S. Ohtsuka. Generating new samples from handwritten numerals based on point correspondence. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 281–290, Amsterdam, The Netherlands, 2000.

[8] H. Rowley, M. Goyal, and J. Bennett. The effect of large training set sizes on online Japanese kanji and English cursive recognizers. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 36–40, Niagara-on-the-Lake, Ontario, Canada, 2002.

[9] P. Simard, Y. Cun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. Hanson et. al., editor, *Advances in Neural Information Processing Systems 5*, pages 50–58. Morgan Kaufmann, San Mateo CA, 1993.

[10] P. Soille. *Morphological Image Analysis*. Springer-Verlag, Berlin Heidelberg, 1999.