Online Handwriting Data Acquisition Using a Video Camera

H. Bunke (1), T. von Siebenthal (1), T. Yamasaki (2), M. Schenkel (3)

(1) Institut für Informatik und angewandte Mathematik
Universität Bern
Neubrückstr. 12
CH – 3012 Bern
Switzerland
bunke@iam.unibe.ch

(2) Faculty of Engineering
Kagawa University
Takamatsu
Kagawa 760 - 8526
Japan

(3) Swiss Federal Institute of Technology
Zurich
Switzerland

Abstract

A novel method for online data acquisition of cursive handwriting is described. A video camera is used to record the handwriting of a user. From the acquired sequence of images, the movement of the tip of the pen is reconstructed. A prototype of the system has been implemented and tested. In one series of tests, the performance of the system was visually assessed. In another series of experiments, the system was combined with an existing online handwriting recognizer. Good results have been obtained in both sets of experiments.

1. Introduction

The field of handwriting recognition can be divided into online and offline recognition. In online recognition, the movement of the pen is recorded as a function of time, while in offline recognition the input data is a two-dimensional spatial array of pixels, acquired by an image scanner. Because of the availability of temporal information, online recognition is often considered the easier of the two problems. However, the applicability of online recognition is severely limited by the fact that the writing device must be physically connected to a computer. Popular writing devices for online recognition are electronic tablet with pen, and mouse.

In this paper we describe a prototypical system for the online acquisition of cursive handwriting data that is based on a video camera. Using this system, the user writes with a normal pen on normal paper. The writing is recorded by a video camera. From the acquired sequence of images the trajectory of the tip of the pen can be reconstructed.

In [1] a system with a similar goal has been described. It is based on the idea of tracking the tip of the pen by means of Kalman filtering. The distinction between pen-up and pen-down strokes is accomplished using a nearest neighbor classifier. In contrast with [1] our system directly uses the trace of ink produced on paper to reconstruct the movement of the pen. Thus the distinction between pen-up and pen-down strokes is made more directly. A system similar to the one described in this paper, but for isolated Japanese characters, has been described in [2].

2. System Description

Greylevel images of a user writing with a pen on a sheet of paper are captured at a rate of 19 images/s. Image resolution is 384 x 288 pixels. The distance between the camera and the pen is about 35 cm, and the angle between the optical axis of the camera and the table about 45 degree. The main steps in the system are “Differential image processing” and “Reconstruction of the pen trajectory”, which are described below. For a more detailed description of the whole system see [3].

2.1. Differential Image Processing

The basic idea behind the system is illustrated in Fig. 1. If we take the difference of two images recorded at time t and t+1, respectively, we obtain, in the ideal case, the trace of ink that has been produced between t and t+1. Assuming a sufficiently high time resolution, this trace of

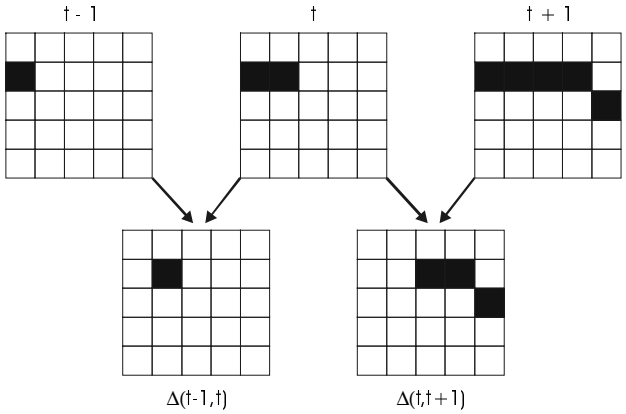

Fig. 1: An example of differential images

ink will be a *small* blob. Its location approximately corresponds to the location of the tip of the pen between t and t+1. Hence, if we compute all differential images in the recorded sequence, we get the complete trace of the tip of the pen. As we are interested in extracting the trace of ink over time but nothing else, only pixels with a transition from white to black, but not from black to white, are included in a differential image.

Under real circumstances, there are the following two problems caused by artefacts:

- **Problem 1**: Differences between two consecutive images, taken at time t and t+1, respectively, are not only due to traces of ink that have been produced between t and t+1, but also to movements of the pen and the hand, and shadows caused by the pen and the hand.
- **Problem 2**: Differences may be also due to the fact that a trace of ink visible at time t may be occluded by the pen or the hand at time t+1, and become visible again at a later time.

In order to solve the first problem, we observe that the last image of a sequence shows the complete text written by the user. This text is extracted from the last image by thresholding using Otus's algorithm [4]. (It is assumed that neither the pen nor the writers hand is visible in the last image of a sequence.) Once the complete text has been extracted from the last image, the problem of reconstructing the trajectory of the tip of the pen is reduced to the problem of finding the temporal order in which the text pixels have been produced. Therefore, it makes sense to restrict all further processing steps to just the text pixels. In order to be able tolerate artefacts caused, for example, by small displacements of the paper during writing, a dilation operation is applied on the text.

For the experiments described in Section 3, a dilation of width 3 has given the best results. Thus we get a region

a)

b)

Fig. 2: a) last image of a sequence; b) region of interest obtained by dilating the image in Fig. 2a

of interest (the dilated text) and restrict all further image processing operations just to this region of interest (It means that our 'online' data reconstruction procedure is started only after the last image frame has been recorded.). A graphical illustration is shown in Fig. 2. Fig. 2a shows the last image of a sequence with the complete text. The region of interest is shown in Fig. 2b. As most of the pen and the hand will be outside the region of interest, Problem 1 will be greatly alleviated. (In addition, a significant speed-up is achieved by restricting all image processing operations to the region of interest.)

To further eliminate artefacts of the 'Problem 1' category, we combine the region of interest method with a lookahead technique along the times axis. If a pixel that changes its value from white to black between time t and t+1 really belongs to the text written by the user, it has to stay black for the rest of the image sequence. Otherwise, if it changes its value from black to white at a later time t+k, it is very likely to be an artefact and is not included in the differential image.

To furthermore combat 'Problem 1', we used a white pen in all our experiments and carefully controlled the lighting conditions in order to reduce hard shadows as much as possible.

To solve 'Problem 2', we first observe that pixels changing their value from black to white because of occlusion by the pen are not included in a differential image, because only white-to-black are recorded there. However, a pixel that has been recorded as part of the written text in an earlier differential image and that is occluded by the pen at time t, may reappear at time t+1 because the pen moved on. In this situation it should not be recorded a second time, i.e., it should not be included in the differential image between t and t+1. Yet this problem can be easily solved by maintaining an *aggregated image* that includes all pixels that have been included in any of the differential images computed at an earlier time.

Initially, the aggregated image includes no pixels. Later, if a pixel is considered for inclusion in an differential image, it is checked whether the same pixel is present in the aggregated image. If it is already present, it won't be included in the considered differential image

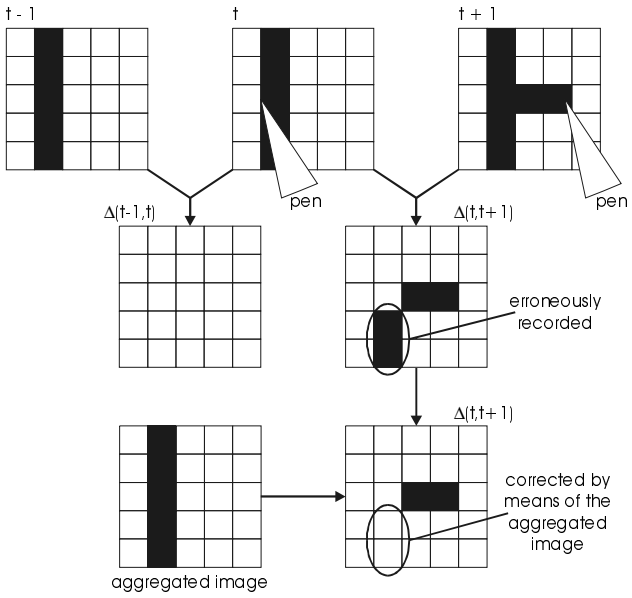

Fig. 3: A simple example of the use of the aggregated image

(i.e., it won't be recorded a second time). Only if the considered pixel is not yet present in the aggregated image, it will be included in the differential image (and then become part of the aggregated image as well). A graphical illustration of this procedure is shown in Fig. 3. Three consecutive images of a sequence are shown in the top row. The corresponding differential images $\Delta(t-1,t)$ and $\Delta(t,t+1)$ are given in the second row. If the differential images are computed without using the aggregated image, two pixels will be erroneously recorded in $\Delta(t,t+1)$. However, if we compare $\Delta(t,t+1)$ with the aggregated image, the pixels will not become part of $\Delta(t,t+1)$.

2.2 Reconstruction of the Pen Trajectory

In the ideal case, each of the differential images will consist of exactly one pixel that exactly corresponds to the location of the tip of the pen between time t and t+1. Under real circumstances, however, the differential image between time t and t+1 contains between zero and some small number of pixels. If the number of pixels is zero, a *null movement* is recorded between t and t+1. In case the considered differential image includes $n \geq 1$ pixels, the position of the pen can be approximated by several methods, for example, by the center of gravity or the median of all pixels in the differential image. We implemented and empirically evaluated some of these methods. The best results were obtained by selecting the pixel as position of the tip of the pen that has the average

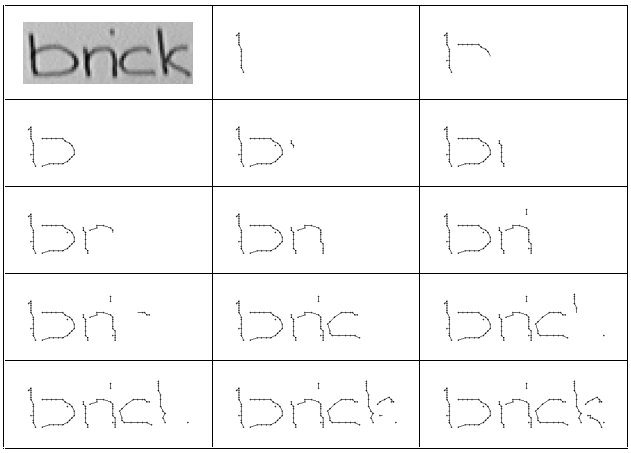

Fig. 4: Reconstruction of the word 'brick'

minimum squared distance to all other pixels in the differential image.

From a sequence of n differential images we get a sequence $p = (p_1, ..., p_n)$ of n "positions" p_i, where each pi is a real position, i.e. a pair of coordinates $p_i = (x_i,y_i)$, or the *null movement* label. In the final step, some postprocessing operations are applied to sequence p. Postprocessing is guided by a few heuristic rules. First, if the Euclidean distance $((x_i-x_{i+1})^2 + (y_i-y_{i+1})^2)^{½}$ of two consecutive coordinate points p_i and p_{i+1} is above a predefined threshold T_1, we add the information 'pen-up' to the movement from p_i to p_{i+1}. Similarly, if a subsequence $p_i, ..., p_{i+k}$ consists of no coordinate points, but only of *null movement* labels, and k exceeds another predefined threshold T_2 (equal to 2 in the experiments reported in Section 3), we add the information 'pen-up' to all movements belonging to this subsequence. By contrast, if $p_i, ..., p_{i+k}$ consists of only *null movements* but k is smaller than T_2, we linearly interpolate the coordinates of $p_i, ..., p_{i+k}$ using p_{i-1} and p_{i+k+1} and replace the *null movement* labels by these coordinates.

3. Experimental Results

The methods described in Section 2 have been fully implemented in C++ and run on SUN workstations and PC's with LINUX. A small set of video clips were acquired to test the performance of the system. In the database of video clips, the input of a user is restricted to a single line of text. A total of 14 lines of text, comprising 52 words, written by two different writers each have been collected.

The performance of the system on the word 'brick' is shown in Fig. 4. In the left upper corner, the (greylevel) subimage containing the complete word is displayed. The rest of the figure shows, from top-to-bottom and left-to-right, the reconstructed trajectory.

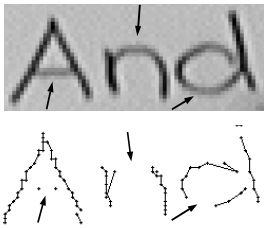

Fig. 5: Reconstruction of the word 'And' (the gaps are due to low contrast in the video images)

There are cases, where the system's performance is not as satisfactory as in Fig. 4. An example is given in Fig. 5, where gaps in the trajectory occur, due to low contrast.

As the ultimate goal of online handwriting data acquisition is usually recognition, we decided not only to assess the performance of our method by visual inspection, but also to do a performance evaluation using an online handwriting recognition system. A number of experiments were run using the recognizer described in [5].

Our database contains only 104 word instances. In order to get more data for training and testing the recognizer, we first extracted the trajectories of the tip of the pen from the available word image sequences using optimal parameter values. (These values were empirically found during system development.) The 104 trajectories obtained this way were used as test set. In order to get training data, additional trajectories were extracted from the given 104 word images, using parameter values slightly different from the optimal ones. Thus another 2177 trajectories were produced. They were used as training set. The trajectories in the training set are similar to those in the test set, but training set and test set are strictly disjoint.

Table 1 shows the results that were obtained in the experiments with the system described in [5]. A total of 12 experiments were conducted along two different dimensions. The first dimension is the amount of retraining that was applied to the recognizer. The other dimension is the size of the lexicon that defines the set of legal words. The *Unix Spell Checker* includes 25144 English words, while the *Task Specific lexicon* consists of just those words from the *Unix Spell Checker* that are included in the test set.

It can be observed from Table 1 that the recognizer behaves quite reasonably on the input data produced by our system. Retraining the recognizer with the data actually provided by our acquisition system greatly improves the recognition rate. On the other hand, the intensity of retraining, i.e. the learning factor, is not a critical parameter. Including dictionary information also significantly improves the recognition rate. The more specific the lexicon is, the greater is the improvement.

	No dictionary	UNIX Spell Checker	Task specific lexicon
No retraining	4.8 %	11.5 %	56.7 %
Learning factor 0.2	35.6 %	**53.8 %**	89.4 %
Learning factor 0.1	**36.5 %**	52.9 %	**90.4 %**
Learning factor 0.05	36.5 %	53.8 %	89.4 %

Table 1: Experimental results with an online recognizer

4. Summary and Conclusions

A new device for online data acquisition of handwriting has been proposed in this paper. It is based on a video camera that records the handwriting of a user. Thus using this system no special writing devices that are physically connected to a computer are needed. Instead just a normal pen and normal paper are required. The system extracts the trajectory of the pen from differential images that are obtained from consecutive images of a recorded video clip.

References

[1] Munich, M., Perona, P.: Visual input for pen-based computers. Proceedings International Conference on Pattern Recognition, Vienna, 1996, 173-176.

[2] Yamasaki, T., Hattori, T.: A new data tablet system for handwriting characters and drawing based on image processing, Proceedings IEEE Int. Conf. on Systems, Man and Cybernetics, Beijing, 1996, 428-431.

[3] von Siebenthal, T.: Online acquisition of hand writing using a video camera, Master thesis, University of Bern, 1998 (in German).

[4] Otsu, N.: A threshold selection method from grey-level histogram, IEEE Trans., SMC-9:62-66, Jan. 1979.

[5] Schenkel, M., Guyon, I. and Henderson, D.: On-Line Cursive Script Recognition using Time Delay Neural Networks and Hidden Markov Models, Machine Vision and Applications, Volume 8, 1995, 215-223.

Acknowledgment

The 2nd author wants to thank Dr. T. M. Ha for his assistance.