A Data-Embedding Pen

Seiichi Uchida[*], *Kazuhiro Tanaka*[*], *Masakazu Iwamura*[**],
Shinichiro Omachi[***], *Koichi Kise*[**]
[*]Kyushu University, Fukuoka, Japan. [**]Osaka Prefecture University,
Osaka, Japan. [***]Tohoku University, Miyagi, Japan.
uchida@is.kyushu-u.ac.jp

Abstract

In order to use handwriting as a universal man-machine interface, we assume a special pen device, called data-embedding pen, which can embed binary data into handwriting as a sequence of invisible ink drops along the handwriting in a real-time manner. This paper describes the assumed hardware, applications, and required technologies of the data-embedding pen. Especially, an accurate stroke recovery algorithm is proposed for retrieving the data embedded in drawing order. In the algorithm, the embedded data itself is fully utilized to improve the accuracy of the recovery. A simulation experiment showed that the algorithm can attain high accuracy on the stroke recovery and the data retrieval.

Keywords: data embedding, handwritings, stroke recovery, invisible ink

1. Introduction

The goal of this research is to enhance the value of handwriting on ordinary paper. For this goal, we assume a special pen device, called *data-embedding pen*. The data-embedding pen can embed arbitrary binary data along black ink stroke while the stroke is drawn on a paper. Specifically, data embedment is done by dropping “invisible” ink from nozzles equipped around the pen tip. The embedded data is represented by a sequence of invisible ink drops.

Using the embedded data, handwriting turns into new media of man-machine interface. Since handwriting can be created without any special skill, they may become a *universal* man-machine interface. In a latter section, we will see their promising applications of the handwriting patterns, such as handwritten bar-code.

Since the data is embedded as a sequence of invisible ink drops in the drawing order of the black ink stroke, it can be retrieved by reading the sequence in the drawing order. Thus, *stroke recovery,* which is the technique to recover the drawing order, is necessary for the retrieval. Stroke recovery is an inverse problem and thus a difficult problem. We will tackle with the problem by fully utilizing the regularity of invisible ink drops.

Today, there are several technologies which try to process handwritings. They, however, do not fully utilize the great flexibility of the handwriting by a pen and paper. A graphic tablet is a typical device to capture handwritings electrically. The drawback of the tablet is that it cannot be separated from a computer; we neither draw any handwriting on the tablet nor observe the handwriting without a computer. Anoto pen [1] is a sophisticated device that can measure and memorize the coordinate of the pen tip on a paper. Unlike the tablet, Anoto can provide human-visible handwriting as black ink stroke on a paper. However, it only accepts special paper where very small dots are printed. The data-embedding pen proposed in this paper is more flexible than those technologies, because it can use any paper; for example, it can makes a handwriting on a package, a post card, or a book. In addition, another and more important feature is that the handwriting provided by the data-embedding pen is not just a pen trajectory; it can convey various data, such as writer’s ID.

In Section 2, the assumed hardware of the data-embedding pen, several applications, and the strategy of the data embedding and retrieval are described briefly. In Section 3, a new algorithm for the stroke recovery is provided. The performance of the algorithm is experimentally shown in Section 4. Finally, Section 5 deals with conclusions and future work.

2. Data-embedding pen

2.1. Applications

As noted in Section 1, the handwriting in which some binary data is embedded can be utilized in various ways. In the following, three different applications of the data-embedded handwriting are shown to emphasize the usefulness of the proposed data-embedding pen.

In the application of Fig. 1, the handwriting is *authenticated* by the embedded data which represents writer’s ID and/or time. This handwriting authentication framework is far more reliable than the conventional matching-based and feature-based signature verification frameworks.

Fig. 2 illustrates the second application where the data-embedded handwriting is utilized as *a link to cyber-space*. The message ID of an e-mail about a meeting is embedded into the handwriting “Meeting”. The user can access cyber-space to recall the detail of the meeting by retrieving the message ID from the handwriting and then viewing the corresponding e-mail in cyber-space.

In the application of Fig. 3, the data-embedded handwriting is used as a *handwritten bar-code* where various information, such as price and producer’s name, is embedded. It is noteworthy that this bar-code is not only human-

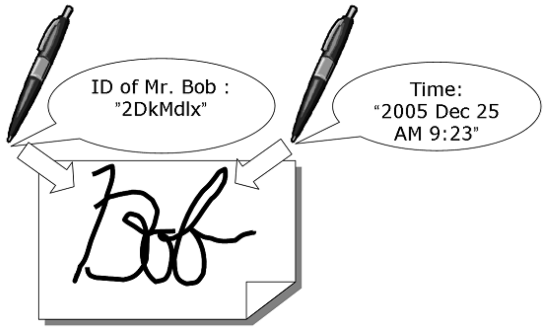

Figure 1. Application 1: Highly accurate signature verification by embedded writer's ID and/or time.

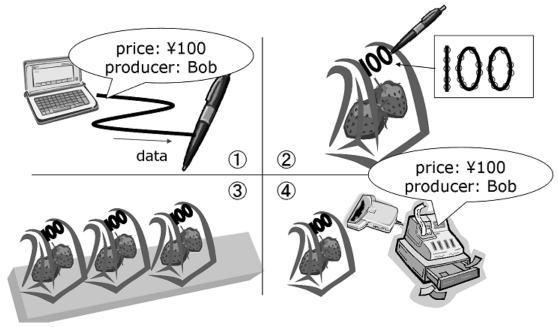

Figure 3. Application 3: Handwritten bar-codes.

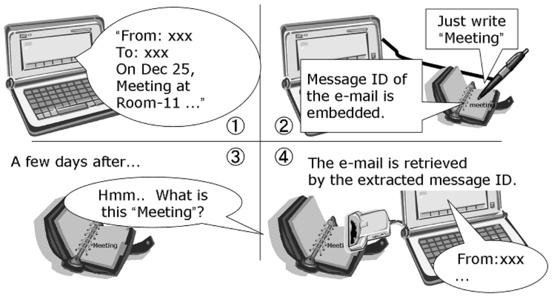

Figure 2. Application 2: Handwritings as a link to cyber-space.

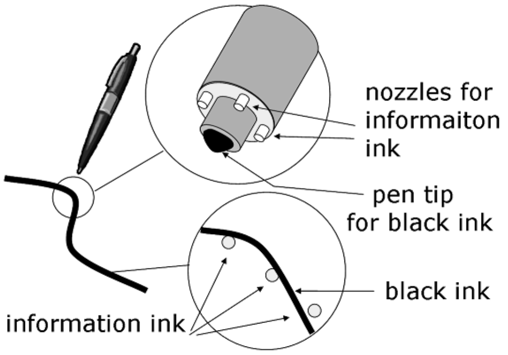

Figure 4. Assumed hardware of data-embedding pen.

readable but also machine-readable. Also note that the handwriting can be an alternative to RFID tag because it can be contact-free machine-readable information attachable to any object (a pack of strawberries in Fig. 3).

2.2. Assumed hardware

The desired goal of this research is the development of the data-embedding pen of Fig. 4. The data-embedding pen deposits two kinds of ink; ordinary black ink and information ink (Fig. 5). The black ink creates a (visible) handwriting pattern and comes out from the pen tip just like a ball-point pen. In contrast, the information ink is periodically dropped from the nozzles equipped around the pen tip. The sequence of information ink drops represents embedded data. The sequence will be detailed in 2.4.2. The retrieval of the embedded data is performed by detecting the information ink in the camera/scanner image of the handwriting.

2.3. Invisible ink

The information ink should be distinguishable easily from the black ink and should not contaminate the black ink stroke. One promising choice for the information ink is invisible ink, which is a special ink and becomes visible only under ultraviolet rays. Color (cyan, magenta, yellow) invisible inks are also available to the public.

2.4. Data embedding

2.4.1. Data ink and guide ink

As shown in Fig. 5, the information ink is further divided into two kinds of ink; data ink and guide ink. The former is used to represent binary data to be embedded. The latter is used not only to separate the data ink but also to represent the drawing direction. The data ink and the guide ink should be distinguishable. As discussed in the next section, they are painted by invisible inks having different colors.

2.4.2. Sequence of information ink drops

Fig. 6 shows an example of the sequence of the information ink drops. Cyan and magenta are used as the data ink and yellow is used as the guide ink. One drop of the data ink represents one bit of data (cyan→0, magenta→1). After continuous M data ink drops ($M = 4$ in Fig. 6), $1\sim 3$ guide ink drops are inserted to separate the data ink drops. In addition, the guide inks represent drawing direction by changing their number as $1\rightarrow 2\rightarrow 3\rightarrow 1\rightarrow 2\ldots$. The

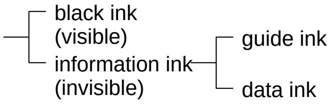

Figure 5. Inks used in data-embedding pen.

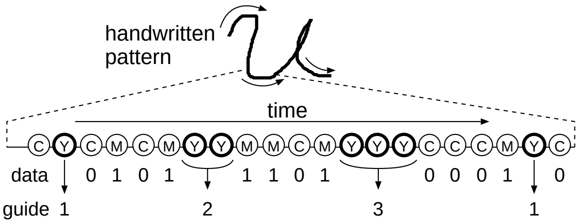

Figure 6. An example of drop sequence of information ink. The letters “C”, “M”, and “Y” represent the color of invisible ink under ultraviolet rays.

direction that the number of the guide inks changes like 1→2→3 (3→2→1) is the correct (wrong) direction.

2.5. Data retrieval

2.5.1. Detection of information ink

For the retrieval of the embedded data, the color and the location of each invisible ink drop should be detected in the image of the handwriting. (If invisible ink is used as information ink, this image should be captured under ultraviolet rays.) The location of the black ink stroke will be helpful for the detection because the drops will locate around the stroke.

2.5.2. Stroke recovery

The detected invisible ink drops are then sorted in the drawing order for data retrieval. The drawing order is not obvious because it disappears in the handwriting image. Thus, the drawing order should be estimated by applying some stroke recovery algorithm to the black ink stroke. The detail of the stroke recovery algorithm is discussed in Section 3.

2.5.3. Removal and correction of erroneous data

The embedded data around stroke intersections and double-traced lines will be intermingled with each other and cannot be extracted correctly. Thus, those intermingled data should be removed by detecting the stroke intersections and the double-traced lines during stroke recovery. For the compensation of the removed data, we can employ (i) repetitive data embedding and/or (ii) error-correcting coding on formatting the sequence of the information ink drops.

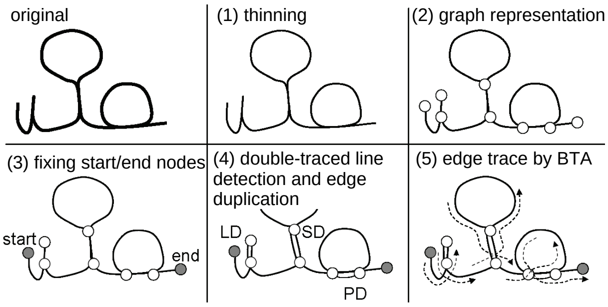

Figure 7. Outline of stroke recovery.

3. Accurate stroke recovery with information ink

Stroke recovery is the technique to recover the drawing order of the black ink stroke and has been investigated in the area of handwritten character recognition. Generally speaking, stroke recovery is an inverse problem where the input (i.e., drawing order) should be estimated from the result (i.e., a handwriting image) and difficult due to the ambiguity in its solution. Thus, most of conventional stroke recovery algorithms employ some heuristics to regularize their solution.

In 3.1, we introduce a promising stroke recovery algorithm by Kato and Yasuhara [2] (hereafter called *basic algorithm*) [1]. Although the basic algorithm mostly works well, it has limitations as shown in 3.2. These limitations come from the above ambiguity. In this paper, a new accurate stroke recovery algorithm is proposed, where the information ink is fully utilized to obtain correct drawing order under the ambiguity. The proposed stroke recovery algorithm will be detailed in 3.3.

3.1. Basic stroke recovery algorithm [2]

The basic algorithm is comprised of several procedures as shown in Fig. 7. The detail of each procedure is described in the following.

3.1.1. Graph representation of handwriting

In the basic algorithm, a single-stroke handwriting image is firstly thinned and then represented as a graph. Each node of the graph corresponds to a stroke intersection or an end-point of the stroke. Each edge corresponds to a stroke between intersections/end-points. An edge may correspond to a double-traced line. In general, the degree of a node is 1, 3, or 4. The node of degree 1 is an end-point or a turn-around point of a double-traced line. The node of degree 4 is a intersection of two strokes crossing like “X”. The node of degree 3 is the beginning or end of a double-traced line.

---

[1] We can employ any stroke recovery algorithm as a basic algorithm. In fact, the authors have tried to use another stroke recovery algorithm based on an optimal path finding approach like the literature [3].

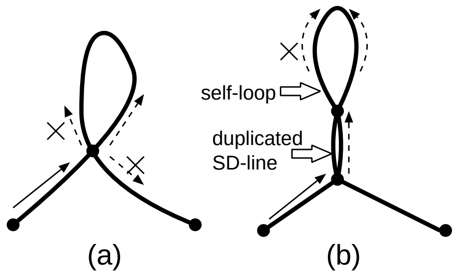

Figure 8. Two main strategies of basic algorithm.

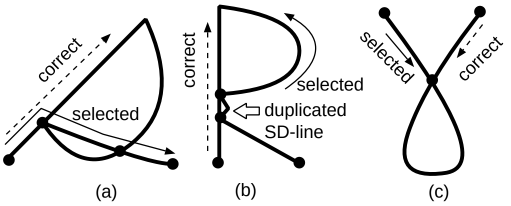

Figure 9. Limitation of basic algorithm.

The thinning operation often make adjacent spurious nodes around intersections. In order to eliminate those spurious nodes, a clustering operation (i.e., the unification of adjacent nodes) is recommended in [2].

3.1.2. Start and end points

Two nodes of degree 1 are selected as the start and the end points of the stroke. Unfortunately, those points are often ambiguous. In [2], a try-and-error selection technique is proposed, where all possible $_MP_2$ selection candidates are tried (M is the number of nodes of degree 1) and the candidate which gives the most "smooth" recovery result is finally selected as the start and the end points.

3.1.3. Detection of double-traced lines

Double-traced lines should be detected for complete stroke recovery. They are classified into three types (Fig. 7):

LD-line : The edge linking a node of degree 3 and a node of degree 1 except the start and the end points.

SD-line : The edge linking a node of degree 3 with a self-loop and another node of degree 3.

PD-line : The shorter between two edges linking the same two nodes of degree 3.

These definitions indicate that the double-traced lines can be detected by examining nodes with odd degrees.

3.1.4. Recovery

After duplicating the edges of the detected double-traced lines, all the nodes except the start and the end nodes have even degrees. Thus, graph theory guarantees that such a graph has a path passing all edges of the graph. In the basic algorithm, this path is searched for by starting from the start node and then choosing the next node according to several strategies at each node until reaching the end node. Fig. 8 (a) shows the most important and general strategy that at the node of degree D the middle edge is selected among $D-1$ subsequent edges. (In most cases, $D=4$.) Fig. 8 (b) shows the strategy specially applied to the node of SD-line. This strategy fixes the ambiguous drawing direction of the self-loop linking a SD-line. (For other strategies, see [2].)

3.2. Limitation of basic algorithm

The basic algorithm mostly works well; however, it has the following limitations:

Problem 1 : The strategy to choose the middle edge at an intersection may fail, as shown in Fig. 9(a).

Problem 2 : The strategy for the self-loop linking to a SD-line may fail, as shown in Fig. 9(b).

Problem 3 : The reversed drawing direction may be obtained as shown in Fig. 9(c); that is, we can always consider another recovery result by exchanging the start node and the end node.

The Problem 1 comes from the ambiguity on choosing the subsequent edge. The Problems 2 and 3 come from the ambiguities of local and global drawing directions, respectively.

Recently, several improvements to the basic algorithm [2] have been proposed [4, 5]. Specifically, an improvement to accept triple-traced lines has been proposed in [5]. Similarly, improvements to accept nodes of degree 5 or more are proposed in [4, 5]. The above problems, however, have not been solved since they come from inevitable ambiguity in the stroke recovery problem.

3.3. Stroke recovery with information ink

In this paper, we try solve the problems in the previous section by fully utilizing the information ink. The Problem 1 is solved by choosing the subsequent edge whose information ink drops can connect smoothly to the information ink drops of the current edge. The smoothness is evaluated by checking not only the number of the guide ink drops but also the number of data ink drops. Thus, for example, if the number of the guide inks changes irregularly like 1→3→2 by choosing a subsequent edge, this edge is discarded and another subsequent edge is examined.

The Problems 2 and 3, which come from the ambiguities of local and global drawing directions, can be easily solved by checking the change of the number of the guide ink drops. That is, the direction that provides the change like 3→2→1 is the wrong direction. In this way, the information ink can improve the accuracy of the stroke recovery and therefore improve the accuracy of the information retrieval.

Figure 10. Results of stroke recovery.

4. Simulation experiment

A simulation experiment was conducted for the evaluation of how the performance of stroke recovery is improved by the help of the information ink.

4.1. Samples

Since the data-embedding pen has not been developed yet, data-embedded handwriting images were prepared in an artificial manner. Specifically, single-stroke on-line handwritings were captured by a tablet and plotted as black ink strokes on binary bitmap images. The pen trajectory on the tablet was used as the ground-truth of stroke recovery.

On-line handwritings of 52 English capital/small letters were drawn by 6 writers on the tablet. Thus, the number of the test images were 312. The average size of the images was about 250×250 pixels.

In the following experiment, it was assumed that the data and guide inks were dropped along the black ink stroke according to the format described in 2.4.2. Thus, the experiment was a simulation because the positions of all information ink drops were known.

Let n be the interval of the information ink drops along the black ink stroke. If a black ink stroke is N pixels in length, $(12N)/(18n)$-bit data can be embedded as the data ink. For example, if $n = 10$, 50-bit data can be embedded into a handwriting whose length $N = 750$.

4.2. Results

4.2.1. Accuracy of stroke recovery

As discussed in 3.1.1, each of the 312 test images was thinned and then represented as a graph. Although the test images were created from tablet data, the thinning operation was necessary because the width of black ink stroke was not 1 around intersections and double-traced lines. The thinning operation and/or the succeeding clustering operation were failed in 9 test images. (For example, an unexpected node of degree 5 was produced.) Those 9 images were excluded from the following experiment; thus, the remaining 303 test samples were used for the evaluation of the proposed algorithm.

The accuracy of stroke recovery was measured by observing the recovery results provided under the condition that the start and the end points were given manually. The basic algorithm (i.e., stroke recovery without information ink) could succeed on 293 (96.7%) images. In contrast, the proposed algorithm (i.e., stroke recovery with information ink) could succeed on 296 (97.7%), 296 (97.7%), 299 (98.7%), and 298 (98.3%) images when the ink interval n was fixed at 3, 5, 7, and 9, respectively. These results reveal the usefulness of the information ink for stroke recovery.

Fig. 10 shows seven examples of stroke recovery results. The both algorithms could succeed in the left four results, whereas only the proposed algorithm could succeed in the right three results. Among those three improved results, the first result overcame Problem 2 and the remaining two results overcame Problem 1 of 3.2 by utilizing the information ink.

It is noteworthy that the conditions in the above experiment were favorable for the basic algorithm. This is because; (i) the condition that the start and the end points were given correctly can avoid Problem 3 of 3.2; (ii) the condition that the thin black ink stroke is created on the tablet can reduce the number of double-traced lines; and (iii) the condition that only single-stroke patterns were subjected is a necessary condition for the basic algorithm. (In other words, the basic algorithm has not been designed to deal with multi-stroke patterns.) In contrast, the utilization of the information ink will remove the necessity of those conditions. The proposed algorithm, therefore, will show further superiority over the the basic algorithm in practice.

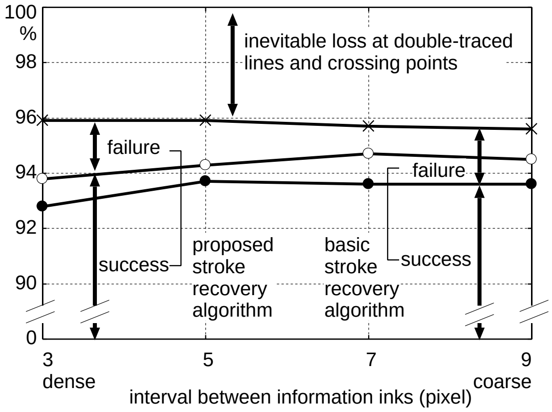

Figure 11. Accuracy of data retrieval.

4.2.2. Performance of data retrieval

The embedded data can be retrieved by reading the information ink drops along the recovered drawing order. This retrieval failures that the information ink drops cannot be recovered in their original order are due to the following two reasons.

- *Errors in stroke recovery*: The embedded data around the strokes whose drawing directions are wrongly estimated will not be retrieved correctly. The amount of this failure depend on the performance of the stroke recovery algorithm.

- *Double-traced lines and stroke intersections*: The information ink drops on these parts are intermingled with each other and cannot be extracted correctly even when stroke recovery is successful. Note that this is inevitable failure but will not be fatal. This is because we can notice those parts from the stroke recovery results and thus remove and compensate the failure by the methods of 2.5.3.

Fig. 11 shows the accuracy of data retrieval when the interval between information ink drops is n pixels. The accuracy is defined as the ratio of the number of the information ink drops which do not suffer from the above failures to the number of all information ink drops. From this result, it is shown that the proposed recovery algorithm could attain 94$\sim$95% data retrieval accuracy. It is also shown that the proposed algorithm outperformed the basic algorithm by about 1% improvement. This improvement indicates the usefulness of the information ink in the data retrieval.

The 6% retrieval failures of the proposed algorithm were comprised of 4% inevitable failures due to double-traced lines and stroke intersections and 2% failures due to errors in stroke recovery. The 6% failures will be not trivial for practical use and therefore emphasizes the necessity of the automatic error removal and compensation discussed in 2.5.3.

5. Conclusion and future work

The assumed hardware, applications, and required technologies of a special pen device, called data-embedding pen, were described. The data-embedding pen can embed binary data into a handwriting as a sequence of invisible ink drops in a real-time manner. A novel stroke recovery algorithm has been proposed as a required technology for the retrieval of the embedded data. The proposed algorithm fully utilizes the drop sequence and could provide higher retrieval accuracy than a conventional algorithm. par In future, we will tackle the following tasks as well as the hardware development of the data-embedding pen:

- *Detection of information ink drops in camera-captured handwriting image*: An image processing method should be developed for the detection of the colors and the locations of invisible ink drops in a handwriting image. Note that without the hardware of the data-embedding pen, the images of data-embedded handwritings can be prepared by an ink-jet printer with cartridges filled by color invisible inks.

- *Multi-stroke handwritings*: The basic stroke recovery algorithm by Kato and Yasuhara [2] assumes only single-stroke handwritings. Its extension for multi-stroke handwritings is necessary. The information ink may be useful for this extension.

- *Correction of erroneous data*: As discussed in 2.5.3, the format the sequence of information ink drops should be reconsidered for realizing automatic removal and compensation of erroneous data around double-traced lines and intersection parts.

Acknowledgment: This work was partially supported by Microsoft Public Trust for Intellectual Property Research Support Fund.

References

[1] `http://www.anoto.com/`

[2] Y. Kato and M. Yasuhara, “Recovery of drawing order from single-stroke handwriting images,” IEEE Trans. Pat. Anal. Mach. Intell., vol. 22, no. 9, pp.938–949, 2000.

[3] E. -M. Nel, J. A. du Preez and B. M. Herbst, “Estimating the pen trajectories of static signatures using hidden Markov models”, IEEE Trans. Pat. Anal. Mach. Intell., vol. 27, no. 11, pp. 1733–1746,

[4] Y. Qiao, M. Nishihara, and M. Yasuhara, “A novel approach to recover writing order from single stroke offline handwritten images,” *Proc. ICDAR*, vol. 1 of 2, pp. 227-231, 2005.

[5] L. Rousseau, É. Anquetil, and J. Camillerapp, “Recovery of a drawing order from off-line isolated letters dedicated to on-line recognition,” *Proc. ICDAR*, vol. 2 of 2, pp. 1121-1125, 2005.