Tracing Handwriting on Paper Document under Video Camera

Jae-Hyun Seok[+] *Simon Levasseur*[++] *Kee-Eung Kim*[+] *Jin Hyung Kim*[+]
Department of Computer Science, KAIST
373-1, Guseong-dong, Yuseong-gu, Daejeon, South Korea[+]
University of Technologies of Belfort - Montbéliard, France[++]
jhseok@ai.kaist.ac.kr, simon.levasseur@gmail.com, {kekim, jkim}@cs.kaist.ac.kr

Abstract

This paper describes a system that traces handwriting on paper document under overlooking video camera. This work is motivated to capture annotations on paper documents written by ordinary pen as an input to computer. As the trajectory of the pen tip is extracted from the video, each part of the trajectory is classified as 'pen-down' or 'pen-up', according to whether the part makes a dark line. Detecting written inks is not simple when handwriting is made over printed documents. Because written inks may fall on dark regions of the document and often overlap previously written inks, simple background checking may not work on dark regions. So, we interpolated the decisions at the entering and the exiting of the dark region. The system makes two-level decisions to achieve both speed and accuracy. The classifier makes quick decisions based on local information in order not to lose pen trace. The local pen up-down decisions are corrected in the global point of view when the whole information of the writing process is available, such as when the hand is out of the view. Experimental result shows that the system detects handwritings accurately even on printed documents.

Keywords: online handwriting trace, annotation detection, pen-based interface

1. Introduction

Handwriting is the most natural ways of input to computers. However, needed are special input devices such as tablets, styluses, and Anoto-like pen and papers [5]. Capturing handwriting by video camera is an attractive alternative because the system would allow the user to write on a normal paper with a normal pen.

By such a technology, we envisioned that handwritten annotations and markers are recorded as a new layer associated with the document. It opens new interesting applications which are needed to transfer handwriting trace to others through internet. For example, a student's solution on an exercise sheet can be transferred to a remote teacher through network. Then it may be overlaid on the same exercise sheet on the teacher's monitor. We may also imagine that handwriting is transferred to a teller in a bank during a customer filling out a slip. Then the information of the customer is known to the teller before he or she shows up in front of the teller with the written slip. So, the processing time would be reduced.

A few studies have been reported for obtaining handwriting trace (i.e. the writing order of the handwriting) from a video image sequence [2,6,7,9,10]. The previous works have dealt with handwritings on blank paper. So, they may not be able to detect annotations or marks on preprinted papers. We challenge to online recover of handwriting traces not only on blank papers but also on papers of printed text and diagrams.

There are two major challenges to meet our goal. The first is that written ink should be detected even when they are written on dark region. Tracing written inks should not be interrupted by printed ink.

The second is to meet the real-time online processing constraint. Each frame should be processed before next frame comes (within 20ms). Otherwise, the next frame would be lost and the pen trace would be inaccurate.

For the online processing requirement, the previous works utilized only locally available information and, therefore, inevitably showed a bit inaccurate tracing result [6,7,9]. We overcame such inaccuracy by using a two-level decision making. Quick local decisions by the online processing are confirmed or corrected in the global point of view. The global checking is performed with enough time when the hand is completely out of the camera view. By the two-level approach, we achieved both speed and accuracy.

The rest of this paper is organized as follows. In section 2, the overall architecture of our system is presented. The local online processing is described in section 3 and 4, where section 3 is for the pen tip tracing process and section 4 is for the pen up/down classification process. Section 5 describes the global confirmation step. Section 6 presents experimental results in an evaluation set up. Finally, conclusion is followed in section 7.

2. System Overview

Our system is composed of a writing space and a video camera overlooking the space. The camera captures image sequences while the user writes by a normal pen on the paper in the writing space. Then the captured image sequences are processed by a computer to produce the pen's trace in real-time. Not only pen trace but also whether the pen is up or down are reported.

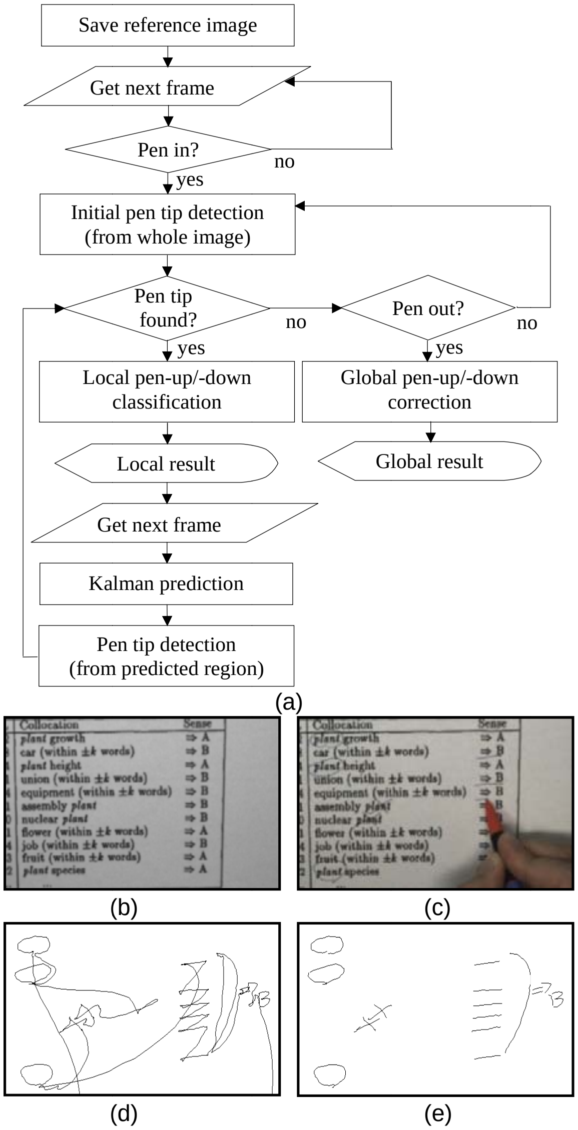

Figure 1. (a) Block diagram of the system. (b) Reference image. (c) Image captured by the camera. (d) Pen trajectory. (e) Pen-down strokes.

Based on operational scenarios, several requirements and assumptions are made for developing our system. The first is that the camera should be properly located so that it can see the pen head during writing. The second is that lighting is good enough not to make strong shadow on written ink area. The third is that the paper on which handwriting made is a normal document with bright background and dark text/diagrams, and it is not moving during writing. The last is that the pen head should have a conical shape with a distinctive color from the paper and printed contents.

Figure 1(a) shows how the system processes an image sequence captured by the camera. At first, the very first image of no hands in is saved as a reference image as shown in figure 1(b). If the pen is detected, the system locates the pen tip, gets pen trace fragment, and classifies it as pen-up or pen-down movement. The position of the pen tip in the next frame is predicted by a Kalman filter [1] and the pen tip search is performed in the vicinity of the predicted region. If the pen tip is not found in the predicted area, the system tries to find the pen tip from the entire image. These processes are repeated until the pen exits from the writing space. Although local pen-up/-down classification can recover handwriting trace promptly during writing, the result is somewhat inaccurate because it is based on only the information obtained at the current image frame. As we mentioned earlier, the global pen-up/-down correction module is called when the pen exits from the writing space. Figure 1(d) shows the trajectory before the classification and figure 1(e) shows the result of the global correction.

3. Pen Tracking

After the initial pen tip detection, a Kalman prediction module is called to track the pen tip. By connecting sequentially the position of the pen tip in each video frame, we can get a complete trajectory of the pen tip as shown in figure 1(d). The pen tracking result counted correct if it is within 2 pixel range.

3.1. Pen Tip Detection

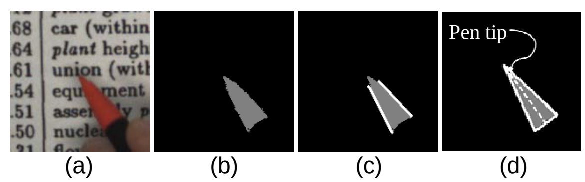

Figure 2. (a) Image with a pen. (b) Detected pen head. (c) Two longest side lines. (d) Triangular model of the pen head.

The first step of the pen tip detection is finding the pen head area. Our process is basically a color matching. The color of the pen head is modeled before using it. In order to be robust on the variation of the light intensity, a normalized RGB value is used in the place of raw RGB. That is, we work in the (r, g) space instead of the (R, G, B) space. The color distribution of the pen head is modeled as Gaussian. If the pixel's probability of being pen head based on the color is higher than a threshold T_p, the pixel is considered as a foreground pixel as shown in figure 2(b).

The pen head is assumed to have a conical shape. Thus, after selecting the two side lines of the pen head, the triangular model of the pen head can be derived as in figure 2(c). Then, the pen tip location is selected on the center line (dotted line in figure 2(d)) by considering the relative pen tip location from the pen head.

3.2. Kalman Prediction

The most likely position of the pen tip on the following frame can be predicted based on its current position, velocity and acceleration. Since the prediction may not accurately indicate the exact position, search process is needed to find the exact position from the predicted region. Good prediction yields reduction on computing time, while maintaining high detection accuracy. We use the Kalman filter algorithm [1], especially the model used by Munich & Perona [7].

4. Pen-up/-down Classification

The complete pen trajectory cannot show what is written because it also contains the traces of pen-up movement. Thus, next process is classifying each part of the pen trace as pen-up or pen-down. As we mentioned earlier, the classification is preformed in two steps: local decision making and global confirmation.

4.1. Local Pen-up/-down Classification

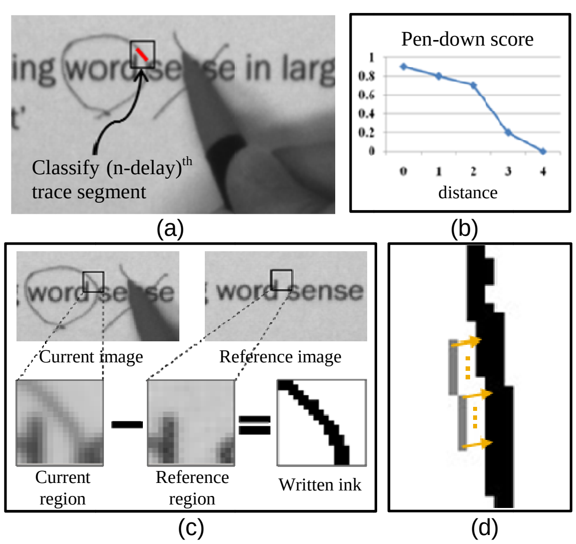

Figure 3. (a) Image of the n next frame which avoids occlusions. (b) Pixel pen-down scores by distance to the nearest written ink. Empirically decided. (c) Written ink Extracting process. (d) Distance to nearest written ink pixel. The gray line is digitized trace segment. The arrows are pointing the nearest written ink pixels.

The unit of local pen-up/-down classification is a *trace segment* which is defined as a line segment between pen tips of two consecutive frames. In order to avoid frequent occlusions by the pen head or the writer's hand, we perform the classification process after a certain time delay[1], with a hope that the writing hand is sufficiently far away from the part of the image under analysis [9].

Because 3D position of pen tip cannot be detected by one camera, pen-up/-down of a trace segment has to be inferred indirectly from whether the trace segment generates ink or not. However, the dark region, called *printed ink*, already exists in the document before the writing, so the current image frame alone cannot give enough information to distinguish *written ink* from printed ink. To extract the written ink, the reference image is needed which has been saved before the start of the writing.

Around the trace segment, region of interest is defined on the current image and the reference image, respectively. Then the pixel value difference between two regions is computed to search written ink. The threshold of pixel value difference is computed by Otsu's method [8]. The pixels of higher difference than a threshold value are regard as written ink pixel. This approach compensates well for variations in the lighting conditions.

The likelihood as being pen-down is based on the distance between the trace segment and written inks. If a trace segment is a pen-down movement, written ink pixels would be found on or near the trace segment. The pen-down score of the pixel is determined by the distance to the nearest written ink pixel computed along the direction perpendicular to the trace segment as shown in figure 3(d). The score by distance has been set empirically as the table in figure 3(b). Relatively high scores are given to the pixels in distance 1 and 2, so that the one or two pixel errors of pen tracking result can be absorbed.

The pen-down likelihood of a trace segment is the score averaged over its pixel scores. Then, we apply thresholding to classify the trace segment as pen-down. If the score is greater than the threshold T_{score} [2], we classify it as pen-down.

4.2. Hidden Ink Restoration

Not all the written ink pixels can be detected by the method described in section 4.1. As shown in the leftmost image of figure 4(a), the written ink overwritten on the printed character is not detected. We call it *hidden ink problem*. Hidden ink refers to the written ink overwritten on a dark region.

Hidden ink may cause errors in pen-up/-down classification of a trace segment. As shown in figure 4(a), the trace segment (denoted by the solid line) is truly a pen-down segment but classified as pen-up, because there are not enough written ink pixels around it. This kind of misclassification can be corrected by detecting hidden ink pixels and restoring them into written ink.

It is impossible to tell whether a stroke is overwritten on the dark region by dark pencil. However, it is natural to think the written line has gone through the character 's' in

[1] We set the delay to approximately 2 second; 100 frames
[2] we set T_{score} to 0.7

the example of figure 4(a). We use such human knowledge to detect hidden ink. We interpolated the decisions at the entering and the exiting of the dark region into the ones on the dark region.

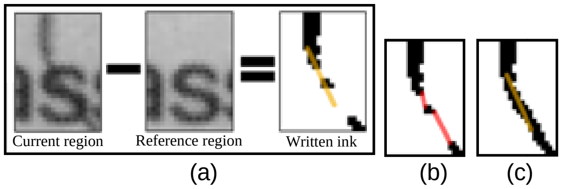

Figure 4. (a) Example of hidden ink. The solid line is a trace segment. (b) Example of connecting lines. (c) Written inks restored.

The rule of hidden ink restoration is as follows. Lines are formed connecting the nearest written inks together as in figure 4(b). Then, we restore written ink pixels along the connecting lines if they satisfy the next two conditions:

- **Condition 1:** The direction of the connecting line is similar to that of the trace segment.
- **Condition 2:** The connecting line is covered with printed ink and/or written ink.

By the process, hidden ink pixels are restored to written ink pixel as shown in figure 4(c). The trace segment which was classified as pen-up is now classified correctly as pen-down.

5. Global Pen-up/-down Correction

The result of local pen-up/-down classification may have errors because it is based on a single frame. Figure 5(c) shows the local classification result of the trace segments in figure 5(b). The small arrows point at the local classification errors. The main source of the errors is that a written ink may overlap previously written traces. If a pen-up trace goes through a written ink area, it may be misclassified as pen-down because many written inks are around, as shown in the examples at the lower left corner of ‘a’ and left part of ‘b’ of figure 5(b). Such a misclassification cannot be avoided in the local classification stage in which trace segment is the unit of the classification decision and decisions are made without a global view.

For the global correction step, a *stroke* is defined as a sequence of trace segments of the same type. A pen-up trace segment is combined with adjacent pen-up trace segments to form a pen-up stroke, while a pen-down trace segment is combined with adjacent pen-down trace segments to form a pen-down stroke. Small gaps are filled preferring long stroke. However, a stroke is separated at high curvature point believing that two strokes may meet at such a point [3]. As the stroke is used as the unit of classification, complete pen trajectory is segmented into strokes and each stroke is classified either pen up or down. Figure 5(d) shows strokes segmented from the trajectory in figure 5(b).

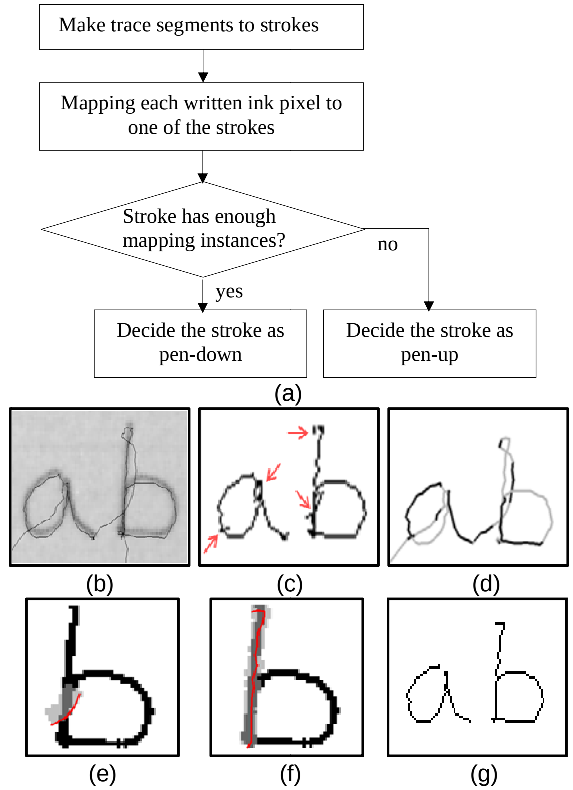

Figure 5. (a) Block diagram of global correction phase. (b) Pen tracking result (c) Local classification result. (d) Strokes. Successive strokes are indicated alternating colors. (e) Example of stroke region denoted by gray area. The stroke is denoted by the thin line (pen-up stroke). (f) Example of stroke region (pen-down stroke). (g) Global correction result.

To correct local classification errors, each written ink pixel finds the stroke which has generated it, based on the assumption that a written ink pixel has been generated from only one stroke. Figure 5(a) shows the processes. At first, a *stroke region* is defined as the area within certain width around the stroke. Examples of the stroke regions of pen-up stroke and pen down-stroke are shown in figure 5(e) and (f), respectively. Each of the written ink pixels in that stroke region should map into a pen-down stroke. If a written ink pixel maps into more than two pen-down strokes, the winning stroke is determined by the confidence measure of being pen-down.

The confidence measure P of a stroke d to be pen-down is defined combining three independent features as,

$$P(L,R,Q\,|\,d) = P(L\,|\,d)\cdot P(R\,|\,d)\cdot P(Q\,|\,d) \qquad (1)$$

where L is the length of the stroke, R is the written ink pixel ratio of the stroke region, and Q is a score reflecting desired continuality of the same type

(a) (b) (c) (d)

Figure 6. Examples of classification result. (a) Images after writing. (b) Complete pen trajectories. (c) Local classification results. (d) Global correction results.

Eq. 1 is designed to satisfy the following conditions:

- **Condition 1:** The longer pen-down stroke gets higher confidence. (related to L)
- **Condition 2:** A stroke which has the higher written ink pixel ratio in the stroke region gets higher confidence. (related to R)
- **Condition 3:** A stroke gets higher confidence if it is between pen-down strokes (related to Q)

An instance of the above confidence measure comparison is as follows. All of the written ink pixels mapped into the stroke in figure 5(e) also map into the stroke in figure 5(f). The pen-down confidence of the stroke in figure 5(e) is 0.073 and that of the stroke in figure 5(f) is 0.645, so the written ink pixels maps into the stroke in figure 5(f). This is what we expected. Also a pen-up stroke is hardly to be pen-down, as no written ink pixel is around it.

After mapping each written ink pixel to one of the strokes, each stroke is classified according to the mapping result. If a stroke obtains enough mapping instances, it is classified as a pen-down stroke. In the above example, the stroke in figure 5(e) is classified as a pen-up stroke as it does not obtain any mapping instance. The one in figure 5(f) is classified as a pen-down stroke as it obtains enough mapping instances. That is, the false pen-down stroke is corrected to a pen-up stroke. As a result of the global correction, more accurate classification is resulted. The errors in figure 5(c) are corrected in the result shown in figure 5(g).

6. Experimental Result

The methods described in section 3, 4 and 5 have been implemented in C++ considering a real-time operation on a Pentium D 3.00GHz PC with a HVR-2300C video camera. The camera has spatial resolution of 640x480 pixels at a frequency of 50Hz. The writing space visible by the video camera covers an area of approximately 14x17 cm^2 so that the writing of multiple word instances or annotations on a part of a document can be observed.

In order to evaluate the performance of the pen-up/-down classification, we collected 11 video clips containing various types of handwriting in English, Korean, diagrams, numbers, and mathematical formulas on several types of printed documents. A trace segment was used as a unit of evaluation and we obtained the ground truth by manually classifying each of the trace segments in the video clips. Table 1 shows the result of the experiment. Figure 6 displays the result of classifications on two test data.

Table 1. Pen-up/-down classification results

Result / Truth	Local classification (# of trace segments)		Global correction (# of trace segments)	
	Pen-up	Pen-down	Pen-up	Pen-down
Pen-up	2044	706	2723	27
Pen-down	118	2690	46	2755

(a) Confusion matrix

	Local classification (%)	Global correction (%)
Accuracy	85.16	98.68

(b) Accuracy

Although the result of local classification is inaccurate, it is not hard to read what is written through the result (see the figure 6(c)) as most of the errors are false pen-down and they occur near the written ink areas.

However, most local classification errors are corrected by the global correction. Almost all false pen-down strokes are corrected and two third of the false pen-up strokes are corrected. The second row of figure 6(c) and (d) shows the case when written ink pixels map into a pen-up stroke to correct its classification because no other pen-down stroke is near the stroke.

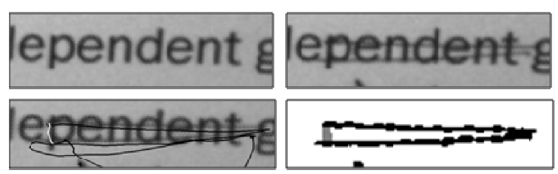

Figure 7. Example of hidden ink restoration error. Each image shows the reference image, the image after writing, the complete pen trajectory and the written ink image, respectively.

We found some undesirable cases. Some pen-up movements coincidentally satisfy the hidden ink conditions in section 4.2 and falsely restore written ink. As in the example in figure 7, the pen-up movement going over ‘p’ (denoted by white line in the lower left image) restores written ink pixels although there is no actual writing on it. This causes misclassification of the pen-up stroke to pen-down. We found that a large portion of errors remaining after global correction falls in this case.

7. Conclusion

Motivated to capture annotations on paper documents written by ordinary pen as an input to computer, we designed and implemented a video-based handwriting tracing system which can extract in real-time handwritings written on printed document. From an overlooking video camera image sequences, pen trajectories are extracted as a sequence of trace segments in the local decision process. When the hand exits from the view, pen trajectories are confirmed or corrected by the global decision process, converting the trace segment sequence into a sequence of strokes with a pen-up/-down class label.

In an evaluation set up, we confirmed our system works reasonably well. It correctly classifies trace segment in 85.16% accuracy in the local classification and in 98.68% in the global classification. The two-level decision is found very effective to achieve both speed and accuracy.

To be more acceptable, the system should handle the movement of background paper during writing and be more robust for severe lighting variations.

Acknowledgement

This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-S2-30S.

References

[1] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*, John Wiley & Sons, 1997

[2] H. Bunke, T. Von Siebenhal, T. Yamasaki and M. Schenkel, “Online handwriting data acquisition using a video camera”, *Proceedings of the International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp. 573-576

[3] D. Chetverikov and Z. Szabo, “A simple and efficient algorithm for detection of high curvature points in planar curves”, *Proceedings of the 23rd Workshop of the Austrian Pattern Recognition Group*, Steyr, Austria, 1999, pp. 175-184

[4] G. A. Fink, M. Wienecke and G. Sagerer, “Video-based on-line handwriting recognition”, *Proceedings of the International Confer-ence on Document Analysis and Recognition*, 2001, pp. 226-230

[5] P. Kauranen, “The ANOTO pen - why light scattering matters”, 2000

[6] M. E. Munich, “Visual input for pen-based computers”, *PhD thesis, California Institute of Technology*, Pasadena, California, January 2000

[7] M. E. Munich and P. Perona, “Visual input for pen-based computers”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.3, 2003, pp. 313-328

[8] N. Otsu, “A threshold selection method from gray-level histograms”, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.9, 1976, pp. 62-66

[9] M. Wienecke , G. A. Fink and G. Sagerer, “A handwriting recognition system based on visual input”, *Proceedings of the 2nd International Workshop on Computer Vision Systems*, Vancouver, Canada, 2001, pp. 63–72

[10] T. Yamasaki and T. Hattori, “A new data tablet system for handwriting characters and drawing based on image processing”, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Beijing, China, 1996, pp. 428-431