Journal of Communication and Computer 13 (2016) 164-170
doi:10.17265/1548-7709/2016.04.003

Handwriting Command Recognition and Digital Operation Using Digitalized Pen

Naoya Toyozumi, Junji Takahashi and Guillaume Lopez

Department of Integrated Information Technology, Aoyama Gakuin University, 1-10-5, Fuchinobe, Chuo-ku, Saganihara, Kanagawa, 252-5258, Japan

Abstract: We aim to realize a novel pen based interface system that provides various digital operations for both low-tech and high-tech people. In this paper, we developed an algorithm that associates drawn shapes, such as a circle and a rectangle, with operating commands, such as message input, address input. Using a commercially available digital pen, we confirmed that our proposed command recognition and operation algorithm worked well and outperformed usuale-mail sending operation in time. As a result, the accuracy of the command recognition is 93.8% and the response time of 90% strokes are within 100 ms.

Key words: Digital operation, shape recognition, human computer interface, handwriting.

1. Introduction

ICT (information and communication technology) has provided more convenient life to us with their development. Recently, SNS (social networking service) such as LINE, Twitter and Facebook, has become part of our life. However, some people cannot use digital services and systems efficiently. One capital reason is that they feel difficulty to use conventional interface systems, such as a keyboard, a mouse, a touch-pad, a flick-input device, and so on because learning them takes long time. So, we focus a spotlight again on hand-writing motion, which was invented as soon after the birth of human race. Many people in the world have basic literacy skills, because anyone is able to master literacy skills as long as he/she receives enough education. As a user interface writing with a pen has many advantages for outputting ideas. First, a pen is good in portability. Second, it excels in instancy. Third, it excels in intuitiveness. So, if digital operations such as talking-memo, file manipulation, mail-sending, tweeting, and so on, could get executed using a pen based interface, not only low-tech people but also high-tech people can take advantage of it. Therefore, we aim to realize a novel pen based interface that provides various digital operations for both low-tech and high-tech people.

The rest of this paper is as follow: Section 2 presents related works of this paper; Section 3 provides a quick overview of our concept of pen based interface system that enables us to enforce digital operation such as, talking a digital memo, sending an e-mail or a short message, and tweeting by just writing a correspond command; Section 4 explains details of each command and an algorithm that recognizes commands and messages; In Section 5, we evaluate accuracies of a recognition and a response time as an application.

2. Related Works

Yeo et al. [1] proposed a handwriting message system. Their system needs smart phone operations to activate the system. Saund and Lank [2] proposed sketch-based interfaces. Their system can switch between two modes: a drawing mode and an editing mode by handwriting trajectories. However, in their system, the detected operation is within the drawing

Corresponding author: Junji Takahashi, Ph.D., assistant professor, research fields: mechatronics, distributed autonomous robotic systems, factory automation, and bio-signal processing.

software. Kolberg and Magill [3] proposed networked appliances control system using a digital pen and a paper. Their system allows users to control appliances only a digital pen, however the control method requires an exclusive paper and takes costs more than a regular one. Matulic and Norrie [4] presented a document-authoring environment for interactive tabletops, in which several common document-editing operations are supported by gestures instead of, or in addition to widgets. However, the gesture by pen and touch may complicate user interface due to many operations. In handwriting gesture study, Wang and Chuang [5] presented digit and gesture recognition for accelerometer-based digital pen, the recognition rate is very high. However, their method is needed high performance computer to train the neural network. Aggarwalet al. [6] proposed a feature representation for the handwritten data captured using depth sensors such as Leap Motion and Kinect. They showed alphabet characters recognition algorithm. A large movement is needed to draw the characters. Amma et al. [7] developed an input system which allows us to write characters in the air using a data grove. Their system is not required any handheld devices, however it is difficult to use it in daily life. Tian et al. [8] presented pen tail gestures, a technique using the pen tail while the pen tip is occupied by other tasks. It is not intuitive because users are not familiar to move pen tail. We have developed pen based interface system for digital operation [9].

3. Writing Based Digital Operation

We provide a quick overview of our new concept of pen based interface system. This system enables us to enforce digital operations such as, taking a digital memo, sending an e-mail or a short message, and tweeting by just writing a correspond command. Actually this system needs a smartphone with Bluetooth connection to make a digital operation.

However, a user does not need to touch any buttons, the screen or the smartphone itself. The necessary operations are inputted by digital pen via written trajectory recognition algorithm. So, it works even if the smartphone is in his/her bag. We have already developed an e-mail transmission operation that consists of command series as an example (Fig. 1). In order to send an e-mail message, the system has to recognize what is the message, who is the destination, and when the system executes the transmission. Many additional applications of writing based digital operation beside the (1) message transmission, (2) note sharing between students and teachers in a school, (3) web search service with handwriting, (4) authentication for net shopping. Therefore, it is considered

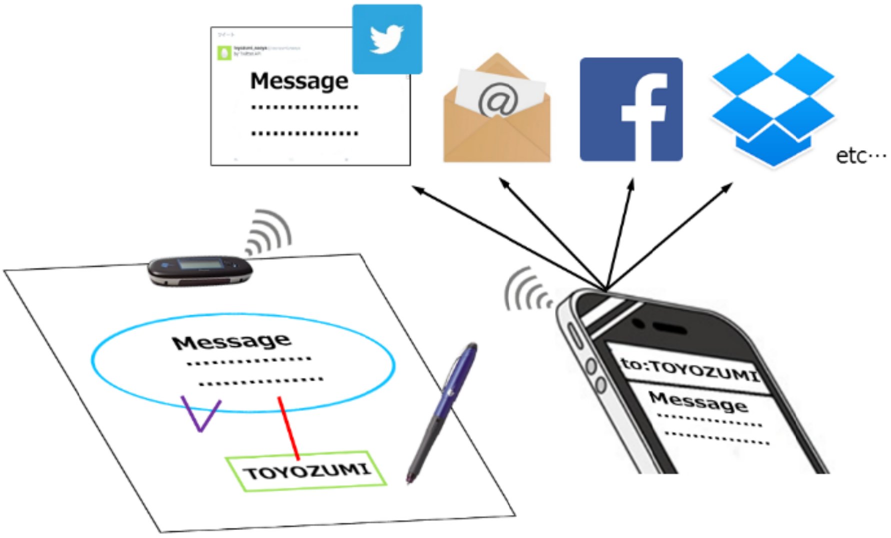

Fig. 1 The concept of digital operation through pen based interface system.

that a pen based interface has many possibilities.

4. Command Recognition Algorithm

4.1 Command Definition

This section describes commands for transferring a message. Each command is drawn with a single stroke. We define a circular trajectory as a MC (message command). When the MC is drawn on a paper, the inside area of the circle is extracted and saved as an image file on the device storage. We define a rectangular trajectory as an AC (address command). When the AC is drawn on the paper, the inside area of the rectangle is extracted and analyzed by OCR algorithm (Tesseract-OCR [10]). Then, the obtained characters are used for matching of registered addresses in a mobile device. We define a line segment, which links the MC and the AC, as a SC (send command). The SC is a trigger for sending the message to the address. This command is recognized when the MC and the AC are connected with a line segment. The forth command is a TC (tweet command), which is recognized when a check sign is drawn near MC. A tweet message attached to the image using a twitter account that has been registered in the mobile device.

4.2 Algorithm Flow

Fig. 2 shows a flow chart of the command recognition algorithm. This algorithm receives a trajectory and classifies it into one of the 5 categories. This process is executed when the pen gets away from a paper.

4.3 Stroke Division

A digitalized pen can detect statuses such as a Pen Down (contact with a paper), a Pen Move (drawing on a paper) and a Pen Up (getting away from a paper). Handwriting trajectories are represented series of drawing points. The drawing points are divided to strokes based on this status cycle and stored as a list format (Fig. 3).

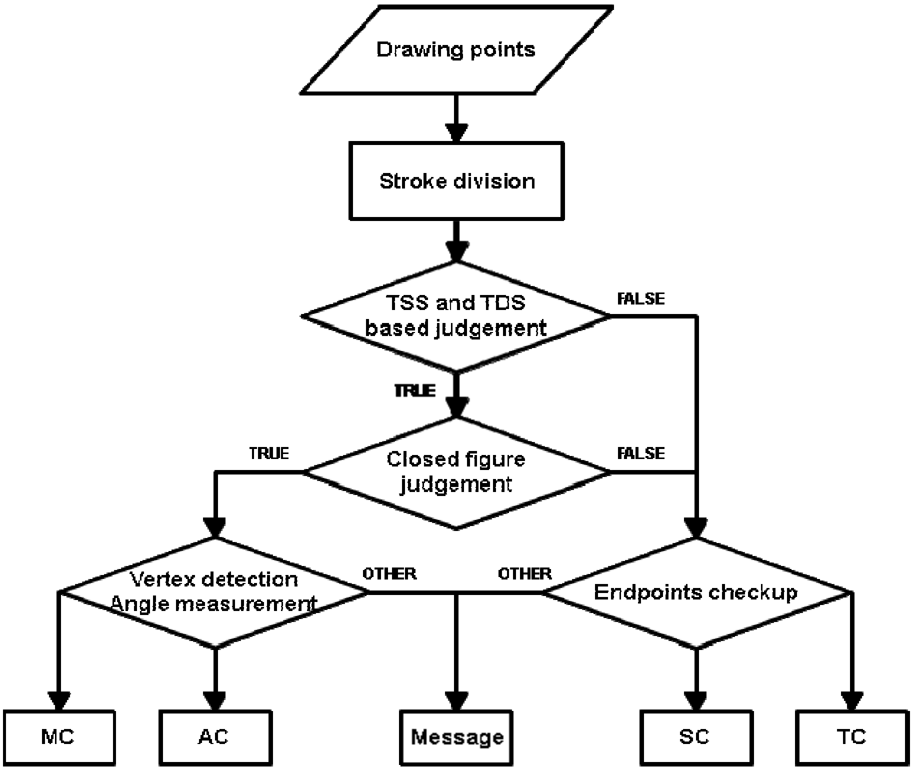

Fig. 2 Command recognition flow.

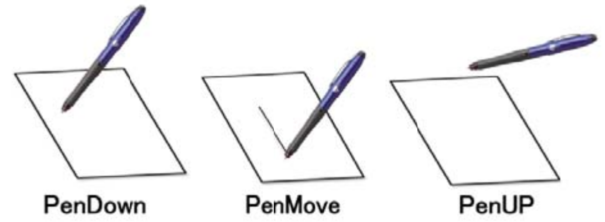

Fig. 3 Pen status cycle.

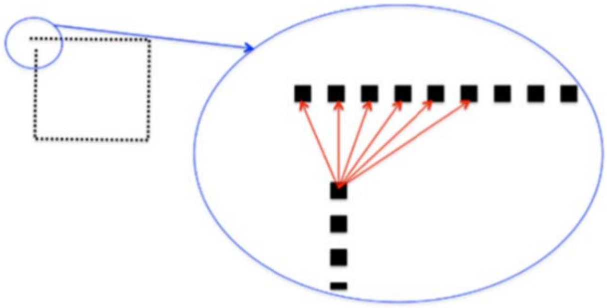

Fig. 4 The minimum distance search.

4.4 Calculating TSS and TDS

In this process, the *TSS* (time span of the stroke) and the *TDS* (travel distance of the stroke) are calculated. We define the starting time and the end time of the stroke as t_0 and t_{N-1}, respectively. Using them and the sampled positions (x_i, y_{i}) (i = 1, 2…N), *TSS* and *TDS* are calculated by following formula,

$$TSS = t_{N-1} - t_0 \tag{1}$$

$$TDS = \sum_{i=1}^{N} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{2}$$

When a user draws a command, *TSS* and *TDS* are expected a large value. Therefore, the stroke satisfied $TSS > T_{tss}$ and $TDS > T_{tds}$ is extracted.

4.5 Judgement of Figure Closing

In this process, a minimum distance between a starting and end point of the stroke is calculated in order to judge whether the stroke is closing or not. A closed figure is considered that starting point group and end point group are near. Let D_{min} and T_{cf} denote the distance and the threshold, respectively. The stroke whose distance satisfied $D_{min} < T_{cf}$ is proceeded next process (Fig. 4).

4.6 Vertex Detection and Angle Measurement

The stroke recognized as closed figure is calculated interior angles and judged as MC, AC or Message. The drawing points tend to be denser in around the vertices because a pen speed becomes slow down there. The dense points may give bad effect to the following angle measurement. Therefore, the points are extracted so that each distance is longer than T_d. In addition, for easy to extract the vertices, the points are searched in order of the distance between the point and the gravity point. We define that the distance between starting and gravity point is $D_{s,g}$ and extracting parameter is P_{ex}. Then T_d is calculated by following formula,

$$T_d = P_{ex} D_{s,g} \tag{3}$$

Fig. 5 shows examples of vertices detection with various-shaped figure such as ellipse, rectangle, and star. The dense points are removed by this process.

Next, a figure composed of the extracted points is considered n-sided polygon, the angles between neighbor points are calculated. The points are defined p_i(i = 1,2…N), we define the vector composed of p_i and p_{i-1} is A, the vector composed of p_i and p_{i+1} is B, interior angle θ_i is described as follows,

$$\theta_i = \cos^{-1} \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{4}$$

Finally, we focus on distribution of θ_i. In this process, an angle range $(0 \leq \theta_i \leq \pi)$ is divided to three parts and numbers of the angle included each range are counted. The number and range is defined as follows,

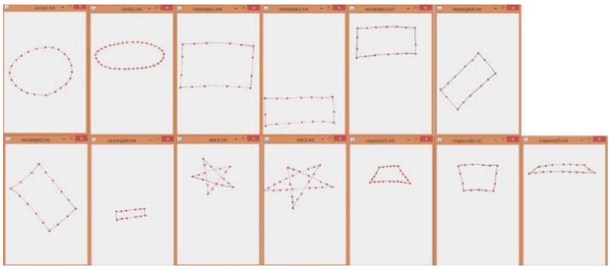

Fig. 5 Examples of detected vertices.

$$\begin{cases} N_s(0 \le \theta \le \frac{1}{3}\pi), \\ N_r(\frac{1}{3}\pi < \theta < \frac{2}{3}\pi), \\ N_s(\frac{2}{3}\pi \le \theta \le \pi). \end{cases} \tag{5}$$

We define a sum of N_s, N_r and N_o as $N_{all.}$ The commands are determined based on the following condition,

$$\begin{cases} AC\, 3 \le N_r \le 5\, and\; N_o = N_{all} - N_r \\ MC\; N_o = N_{all} \\ Message\, otherwise \end{cases} \tag{6}$$

4.7 Endpoints Checkup

In case that the stroke is not a closed figure, the stroke is recognized as SC, TC or Message. In this section we focus on the relation between the position of last stroke points and a drawn command. For this, it is necessary to identify that a point is whether in a command or not. Let x-maximum, x-minimum, y-maximum and y-minimum points of the command be *max* (x_i), *min*(x_i), *max*(y_i) and *min*(y_i) respectively (Fig. 6). We define an endpoint point as (x', y'), if the following condition is satisfied, the point is judged that is included in the command,

$$\begin{cases} \min(x_i) < x' < \max(x_i) \\ \min(y_i) < y' < \max(y_i) \end{cases} \tag{7}$$

In case of SC recognition, if the point is in MC and the other is in AC, the stroke is recognized as SC. In case of TC recognition, if the starting and end point is in MC and some point in out of MC, the stroke is recognized as TC.

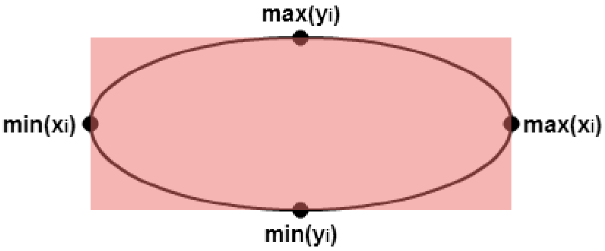

Fig. 6 Included point detection.

5. Experimental Evaluation

We evaluate the algorithm from two points of view. One is the recognition accuracy. The other is the response time between writing finish a stroke and showing a result of the recognition on a display.

5.1 Experimental Setup

We use Airpen (PENTEL Co.) as a digital pen. It is connected to Nexus 7 (Google Inc.) through Bluetooth connection in order to obtain data of the process time, pen position and pen status. The data is send to PC through UDP connection and analyzed off-line. The subject writes in the following order: (1) "Hello world"; (2) Circle around "Hello world"; (3) Own name; (4) Rectangle around your name; (5) Connect to the circle and rectangle; (6) Check sign near the circle. This is conducted 10 times per man. The subject number is 14 people.

5.2 Accuracy Evaluation

Table 1 shows the accuracy before a parameter tuning. The precisions are high rate. However, the recalls of AC, SC and TC are not because it is necessary that MC and AC are already recognized in order to recognize SC. The algorithm needs some parameters (T_{tss}, T_{tds}, T_{cf} and P_{ex}). In this result, the parameters are determined empirically. So, we apply the parameter tuning to the algorithm with a grid search method. Let the search ranges be $0 \le T_{tss} \le 10000$, $0 \le T_{tds} \le 1000$, $0 \le T_{cf} \le 1000$, $0 \le P_{ex} \le 1$. Let a step sizes be one-tenth of the maximum range value. Table 2 shows results of command recognition after the grid search and Table 3 shows the parameters then. The accuracies are improved as a whole, especially AC one. It is considered that a small rectangle is easy to be recognized. However, the TC accuracy is not improved well. Causatively, it is possibility that the starting or end point of some check-sign strokes is not included in MC. So, it is necessary TC is recognized exactly. As the reason that T_{tss} becomes 0 after the tuning,

Table 1 Results of command recognition (before tuning).

Accuracy (%) 88.7		True Command					Precision (%)
		MC	AC	SC	TC	MSG	
Recognized Command	MC	129	2	0	0	0	98.5
	AC	4	101	0	0	0	96.2
	SC	0	0	84	0	1	98.8
	TC	0	0	0	101	1	99
	MSG	10	32	50	40	2785	95.5
Recall (%)		90.2	74.8	62.7	71.6	99.9	

Table 2 Results of command recognition (after tuning).

Accuracy (%) 93.8		True Command					Precision (%)
		MC	AC	SC	TC	MSG	
Recognized Command	MC	136	4	0	0	1	96.5
	AC	2	128	0	0	0	98.5
	SC	0	0	113	0	1	99.1
	TC	0	0	0	103	1	99.0
	MSG	5	3	21	38	2784	97.6
Recall (%)		95.1	94.8	84.3	73.0	99.9	

Table 3 Parameter set.

	T_{tss}	T_{tds}	T_{cf}	P_{ex}
Empirically	700	3000	200	0.5
Grid search	0	1000	200	0.4

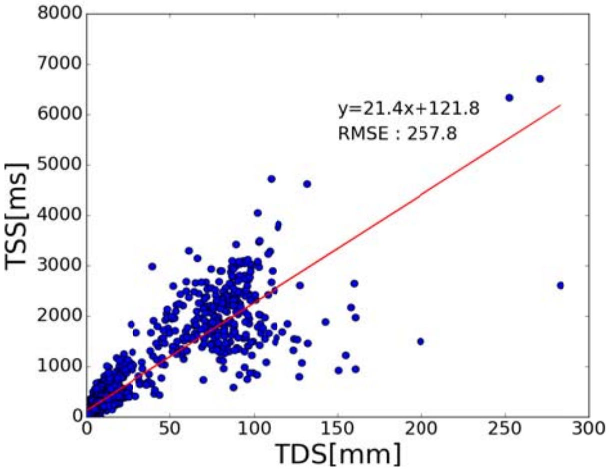

Fig. 7 Relation between TDS and TDS.

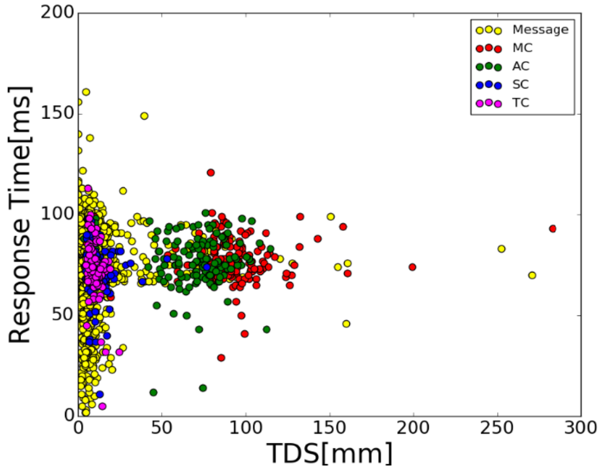

Fig. 8 Relation between TDS and response time.

it can be considered that TSS and TDS have a strong relation (Fig. 7).

5.3 Response Evaluation

Fig. 8 shows the relation between the response time and TDS. We assumed that the response time increases proportionately to TDS. However, in fact, 90% plots are located within 100 ms.

6. Conclusion

In this paper, we proposed the digital operation using digitalized pen and developed the algorithm which recognizes the commands and the message. The result of the experimental evaluation showed that the accuracy is 88.7%. Furthermore, the accuracy improved to 93.8% after the grid search based parameter tuning. The result of response evaluation showed it within about 100 ms and it found that user interface is good. This application is limited to transmitting message such as e-mail and tweeting. Various-type digital operations such as transmitting other SNS are needed.

References

[1] Yeo, J. Y., Lee, Y. D., Sang, H. J., and Gu, M. J.. 2011. "Design and Implementation of a Hand-Writing Message System for Android Smart Phone Using Digital Pen." In *Proceedings of Int. Conf. MulGraB*, 133-8.

[2] Saund, E., and Lank, E. 2011. "Minimizing Modes for Smart Selection in Sketching/Drawing Interfaces.". Book: *Sketch-based Interfaces and Modeling*, Chap. 3, 55-80.

[3] Mario, K., and Evan, H. M. 2006. "Using Pen and Paper to Control Networked Appliances." *IEEE Communications Magazine* 44 (11): 148-54.

[4] Fabrice, M., and Moira, C. N. 2013. "Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops." In *ITS '13 Proc. of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, 41-50.

[5] Jeen, S. W., and Fang, C. C. 2012. "An Accelerometer-Based Digital Pen with a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition." *IEEE Transactions on Industrial Electronics* 59 (7).

[6] Rajat, A., Sirnam, S., Anoop, M. N., Jayanthi, S., and

Jawahar, C. V. 2015. “Online Handwriting Recognition Using Depth Sensors.” Presented at the 2015 13th International Conference on Document Analysis and Recognition (ICDAR).

[7] Christoph, A., Marcus, G., and Tanja, S. 2014. “Airwriting: A Wearable Handwriting Recognition System.” *Personal and Ubiquitous Computing* 18 (1).

[8] Feng, T., Fui, L., Ying, Y. J., Zhang, X. L., Cao, X., Dai, G. Z., and Wang, H. A. 2013. “An Exploration of Pen Tail Gestures for Interactions.” *International Journal of Human-Computer Studies* 77 (5): 551-69.

[9] Naoya, T., Takahashi, J. J., and Guillaume, L. 2015. “Development of Pen based Interface System for Digital Operation.” In *Proc. of IEEE/SICE Symposium on System Integration,* Nagoya.

[10] Smith, R. 2007. “An Overview of the Tesseract OCR Engine.” Presented at the ICDAR’2007, International Conference on Document Analysis and Recognition.