A Digital Ink Recogntion Server for Handwritten Japanese Text

Daqing WANG, Bilan ZHU, Masaki NAKAGAWA
Department of Computer and Information Sciences
Tokyo University of Agriculture and Technology
Tokyo, Japan

***Abstract*— This paper describes the design and implementation of a digital ink recognition server for handwritten Japanese text. Currently, fast and accurate recognition of online handwritten characters requires a high-performance CPU and a large memory space. However, PDA and other small portable devices lack the qualification. On the other hand, data transmission speed is getting remarkably higher. To solve the CPU and memory constraint on a client and avail the speed-up of transmission, we employ a network environment and build a digital ink recognition server. This paper describes network data flow transmission, client/server architecture and presents performance.**

Recognition Server; Online Handwriting; Client/Server; Japanese Text; Digital Ink

I. INTRODUCTION

In recent years, due to the development of pen input devices such as Tablet PCs, electronic whiteboards, PDAs, digital pens (like the Anoto pen) and so on, handwritten text recognition as an input method has been given a considerable attention.

However, PDA and other small portable devices give an unsatisfied performance due to limited processing performance and memory limitation. As a result, accurate processing in real-time recognition is difficult, especially, in continuous writing with less constraint for handwritten text recognition [1]. In addition, desktop systems and electronic whiteboards as sufficient hardware resources are available, the cost to maintain accurate handwriting recognition engine cannot be ignored.

At present, there are so many servers for the speech recognition, [2] and [3] both had given a good construction model. However, for online handwritten character recognition server, the research is scarce, and most systems are based on application program. Sakurada et al. [4] refers to the applications of an online handwritten character recognition server, but it was just tailored for the limited applications.

In this paper, to provide an online handwritten character recognition service and to solve problems existing in the small portable devices, we propose building an online handwritten character recognition server system based on Linux to offer an online recognition service and present a construction model to allow rich interaction with users. Consequently, small devices can provide the best performance in real-time.

Section 2 discusses the client/server architecture, and describes the handling procedure in the server. Section 3 describes our online handwritten character recognition engine using the latest technology. Section 4 shows a client application. Section 5 present experiments and results to show that building an online handwritten character recognition server is feasible. Section 6 draws conclusion.

II. CLIENT/SERVER ARCHITECTURE

The design of client/server architecture arises from common net structure where a server offers online handwritten character recognition for client's programs. We can learn some strategies and techniques from [5]. In our experiment, sampling of digital ink (a series of pen-tip coordinates form pen-down to pen-up) and preprocessing run only on client's side. These ink data are transferred to the recognition server via Internet. Finally, the recognized text is sent back to client's writing application. The architecture of system is shown in Fig. 1.

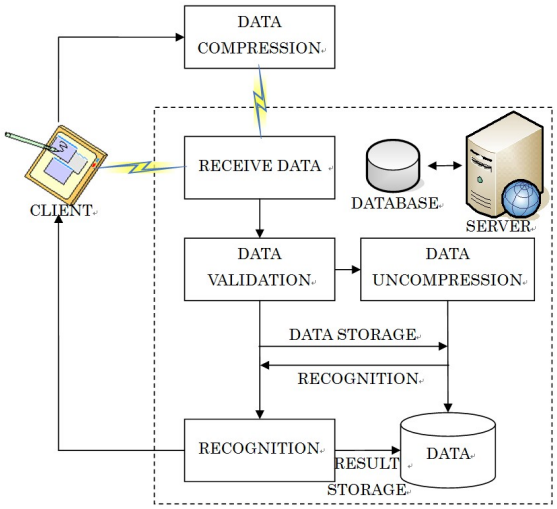

Figure 1. System Architecture

Network transmission is based on TCP / IP protocol, the entire transfer process is as follows:

Firstly, the client collects ink data, and sends them to the server. Data transmission can be divided into non-compressed format and compressed format. Compressed format also can be divided into lossless compression and lossy compression. We employ a lossy compression which is used for character recognition so that it does not degrade recognition performance. The client takes that role from the server since it decreases the transmission cost.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.38

It extracts feature points in the ink data. The method is often applied in on-line handwritten character recognition field [6]. As shown in Fig. 2, take the number "4" for example, first, the starting point and the end of every stroke are picked up as feature points. Then, the most distance point from the straight line between adjacent feature points is selected as a feature point if the distance to the straight line is greater than a threshold value. This is applied recursively until no more feature point is selected. The feature points extracting process is shown in Fig. 3. Due to the fact that unselected points cannot be restored, this is lossy compression. We also call a sequence of selected feature points as digital ink.

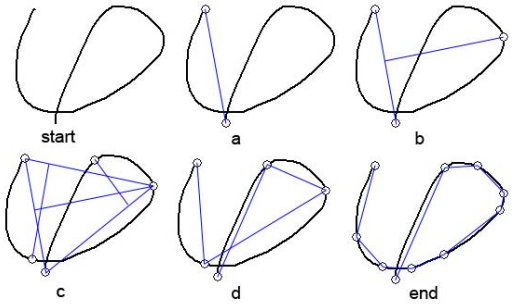

Figure 2. Feature Points Extraction

Before sending digital ink, it is encoded with header data, so that it can be handled by the Server. Besides, the Server supports HTTP protocol via a CGI program, offering an interface of XML data format. XML is a textual data format, and is widely used for the representation of arbitrary data structure, In [7], Jon Bosak gives a description of it, and the SVG image is also organized by XML.

Secondly, on the server side, when a user was authorized, the server creates a connection, receives data, and checks the data; only the correct data format can be recognized. After data validation, according to the data header information it decompresses the data, which directly goes to be recognized. According to user needs, the server determines to store the data or not. Finally, send back the recognition results to the client application. In addition, the server also supports the keep-alive and non-alive connection. The Server is a shared resource, and the client without response for a long time can take the non-alive connection in order to save the Server resources, while the frequent interaction client can take keep-alive connection to improve the response speed.

III. RECOGNITION ENGINE

Our online handwritten Japanese text recognition engine provides on-line handwritten Japanese text recognition without writing box constraints. It evaluates the likelihood of candidate segmentation, character recognition, geometric and linguistic contexts [8]. There are three major steps, over-segmentation, candidate lattice construction and string recognition. The character recognizer and geometric scoring functions is trained by a Japanese on-line handwriting database Nakayosi [9, 10], and the recognition rate is about 93% for the TUAT Kondate database.

For the geometric scores, four quadratic discriminant function (QDF) classifiers are trained for $p(bi\ |C_i)$, $p(q_i\ |C_i)$, $p(pu_i|C_i)$ and $p(pb_i|C_{i-1},C_i)$, respectively.

The character recognizer combines off-line and on-line recognition methods to allow both cursive handwriting and stroke order free recognition by normalizing the recognition scores to conditional probabilities $p(z_i\ |C_i)$ [9] as shown in Fig. 3.

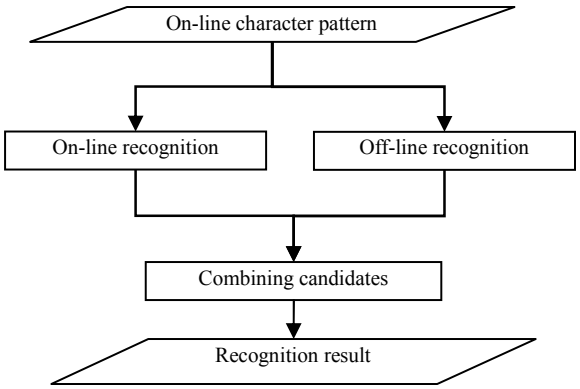

Figure 3. Combined Recognition

For the on-line recognizer, we extract feature points along the pen-tip trace from pen-down to pen-up. We employ the coordinates of feature points as unary features and the displacements in coordinates between the neighboring feature points as binary features. Then we use a MRF model to match the feature points with the states of each character class and obtain similarity for each character class. We then select the top character categories with the largest similarities as the output candidates of the fine classifier [6].

For the off-line recognizer, from on-line character patterns (sequences of stroke coordinates) we extract direction features: histograms of normalized stroke direction [11]. For the coordinate normalization methods we apply pseudo 2D bi-moment normalization (P2DBMN) [12]. The local stroke direction is decomposed into 8 directions and from the feature map of each direction, 8x8 values are extracted by Gaussian blurring. So, the dimensionality of feature vectors is 512. Then, to improve the Gaussianity of feature distribution, each value of the 518 features is transformed by Box-Cox transformation (also called variable transformation). The input feature vector is first reduced from 512D to 90D by Fisher linear discriminant analysis (FLDA) [13]. Then we use the nD feature vectors to create a modified quadratic discriminant function (MQDF) recognizer [14].

IV. A CLIENT APPLICATION

Fig. 4 shows an example of a client application. Other applications and interfaces are possible as far as they follow the protocol with the server.

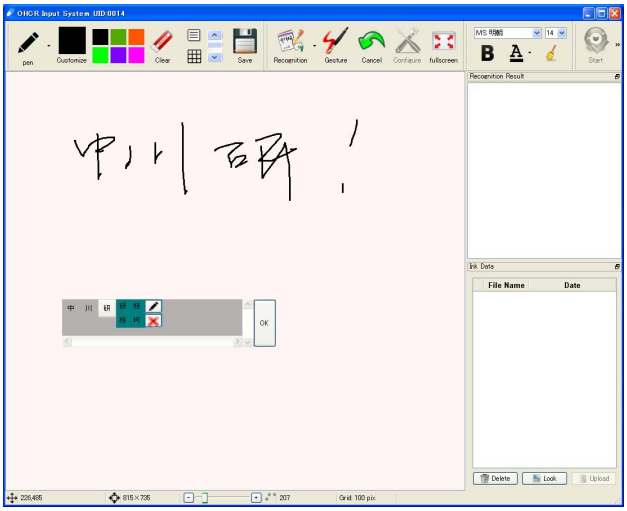

Figure 4. User Interface of a Client Application

Our recognizer combines an on-line recognizer and an off-line recognizer to allow both cursive handwriting and stroke order free recognition, but it is for each character pattern. If a user adds some strokes after writing another character, they are not correctly recognized. This happens when a user add missing strokes later or overwrite strokes on characters written previously. This system allows the user to teach correct segmentation into each character pattern by encircling its strokes as shown in Fig. 5. To allow this interaction, the client application reorders the strokes according to encircling information, and then the server re-recognizes the strokes again to get the correct result.

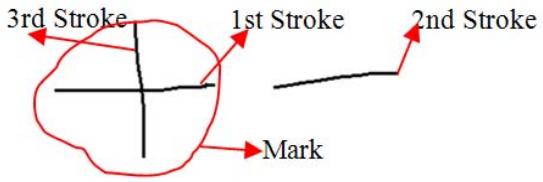

Figure 5: Writing not in Order

V. EXPERIMENTS AND RESULTS

In the actual experiments with a 10M/100M wireless switch, we have verified that our system supports continuous writing with less constraints and returns recognition result in real-time. The test pattern we used is shown in Fig. 6, D1, D2, D3, D4, D5, and the data size of each pattern after encoding is shown under each pattern. Here we just present the experimental results for a few test patterns.

In Experiment 1, we send each test pattern 20 times to the server. The response time is shown in Fig. 7. From the figure we can see that for a certain size of data, the response time is similar. As data size becomes larger, however, the response time increases, because the recognition handling time depends on the number of pen points, that is, as the character becomes more complex or the string becomes longer, the recognition time becomes longer too.

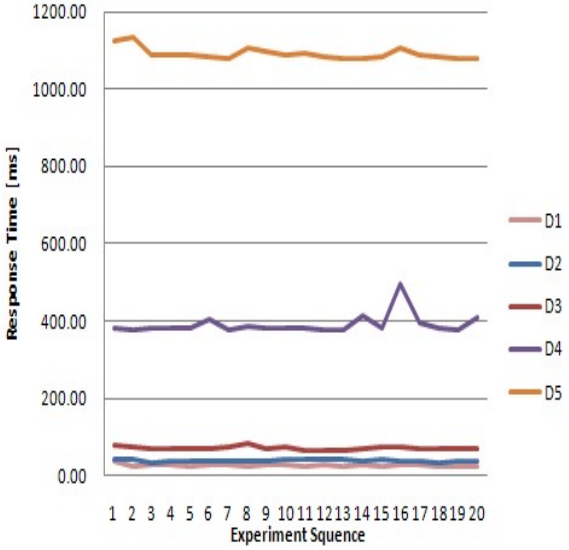

Figure 7. Response Time of Different Size Data

In Experiment 2, we test the data on different client computers and get the average response time. The experimental result is shown in Fig. 8, Exp-A was tested on the Server machine so that it doesn't involve the network transmission. Exp-B was tested on a poor client computer while Exp-C, Exp-D, Exp-E and Exp-F were tested on different client computers of ordinary configuration. From the figure, we can see:

1) The response time is similar to a stand-alone system where character recognition applications run on a computer with more or less processing power. The reason is that our recognition program is running on a high-performance server computer, and the response time is determined by a network environment, rather than handling speed of the clients (From Exp-A, Exp-C, Exp-D, Exp-E and Exp-F).

2) Handling time on different client computers is similar, no matter on higher or lower processing power computer (From Exp-C, Exp-D, Exp-E and Exp-F).

3) Our client/server architecture processes handwritten patterns at much higher speed than many of the clients running on their own client-side character recognition (From Exp-B and Exp-C).

In Experiment 3, we use different number of clients to access our server system. In Fig. 9, we test 15 concurrency clients, and get the average response time of each client. We will see that because of waiting time, each client's response time is different, especially when a few of clients send requests in the same time, and the server will make some clients wait. But each client's average response time is similar, because the waiting chance is same. In Fig. 10, we make the clients number 5, 10, 15, 30 and 50, we can see that the response time of different clients is similar when the number is not big, but as the number is increasing, because the waiting chance is increasing too, so the response time is also increasing.

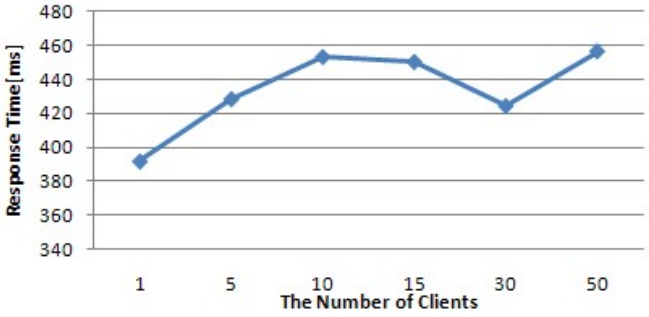

Figure 9. Average Response Time to a Different Number of Clients

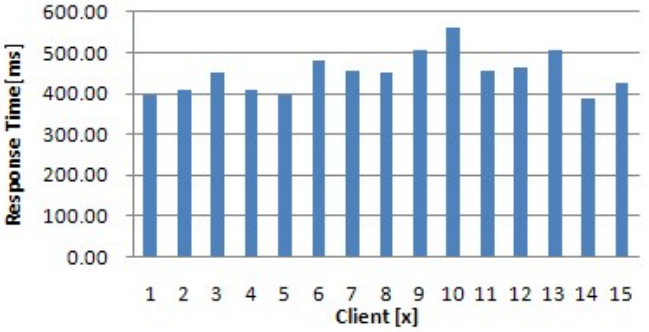

Figure 10. Average Response Time of Each Client with Sending a Same String in 10 Times

VI. CONCLUSION

This paper described a digital ink recognition server for handwritten Japanese text. The server provides on-line handwritten character recognition via network for a common PC, or a small portable device without high-performance hardware requirements. It is expected to widen the application area of pen-based, paper-based or handwriting-based interfaces. We have confirmed that two or more character recognition applications can use a single recognition server at the same time. The response time is similar to a stand-alone system where character recognition applications run on a computer with more or less processing power. The response time strongly depends on connection speed. In the future we are going to propose higher data flow transmission, and provide an interface of other wireless transmission, and support even more devices. Also from our experiments, we know that with the string becomes longer, the handling time increases, so that we will introduce background recognition into the server system (avoid from big data transferring) and provide much faster and more reliable recognition service.

ACKNOWLEDGMENT

This work is being partially supported by the R&D fund for "development of pen & paper based user interaction" under Japan Science and Technology Agency.

REFERENCES

[1] B. Zhu, X. D. Zhou, C. L. Liu, M. Nakagawa: Effect of Improved Path Evaluation for On-line Handwritten Japanese Text Recognition, 10th International Conference on Document Analysis and Recognition, 2009

[2] M. Holada: Internet Speech Recognition Server. SCI 2003, Orlando, Florida, USA, pp.388-391, 2003

[3] R. C. Rose, I. Arizmedi: Efficient Client-server based Implementations of Mobile Speech Recognition Services, Speech Communication, vol.48, no.11, pp.1573-1589, 2006

[4] T. Sakurada, M. Yorifuji, M. Onuma and M. Nakagawa: Web-Based Applications Using Pen-Based Interfaces and Network-Based On-line Handwriting Recognition, Proceeding of 10th International Conference on Human-Computer Interaction, Crete, Greece, vol.2, pp.268-272, 2003

[5] M. C. Lacity, et al: A Strategic Client/server Implementation: New Technology, Lessons from History, Journal of Strategic Information Systems (6), pp.95-128, 1997

[6] B. Zhu, M. Nakagawa, A MRF Model with Parameters Optimization by CRF for On-line Recognition of Handwritten Japanese Characters, Document Recognition and Retrieval XVIII, USA, 2011

[7] J. Bosak, T. Bray, XML and the Second-Generation Web, Scientific American, pp.89-93, 1999

[8] B. Zhu, X. D. Zhou, C. L. Liu, M. Nakagawa, A robust model for on-line handwritten Japanese text recognition, Document Recognition and Retrieval XVI, 2009

[9] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, M. Nakagawa: A Compact On-line and Off-line Combined Recognizer, Proc. IWFHR-10, La Baule, France, pp.133-138, 2006

[10] M. Nakagawa and K. Matsumoto: Collection of On-line Handwritten Japanese Character Pattern Databases and Their Analysis, IJDAR, vol.7, no.1, pp.69-81, 2004

[11] C. L. Liu, X. D. Zhou: Online Japanese caracter recognition using trajectory-based normalization and direction feature extraction, Proc. 10th IWFHR, pp.217-222, 2006

[12] C. L. Liu, K. Marukawa: Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, Pattern Recognition, vol.38(12), pp.2242-2255, 2005

[13] M. Cheriet, N. Kharma, C. L. Liu, C. Y. Suen: Character Recognition Systems, A Guide for Students and Practioners, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007

[14] F. Kimura: Modified quadratic discriminant function and the application to Chinese characters, IEEE PAMI, vol.9 (1), pp.149-153, 1987

Figure 6. Response Time to a Different Size of Data

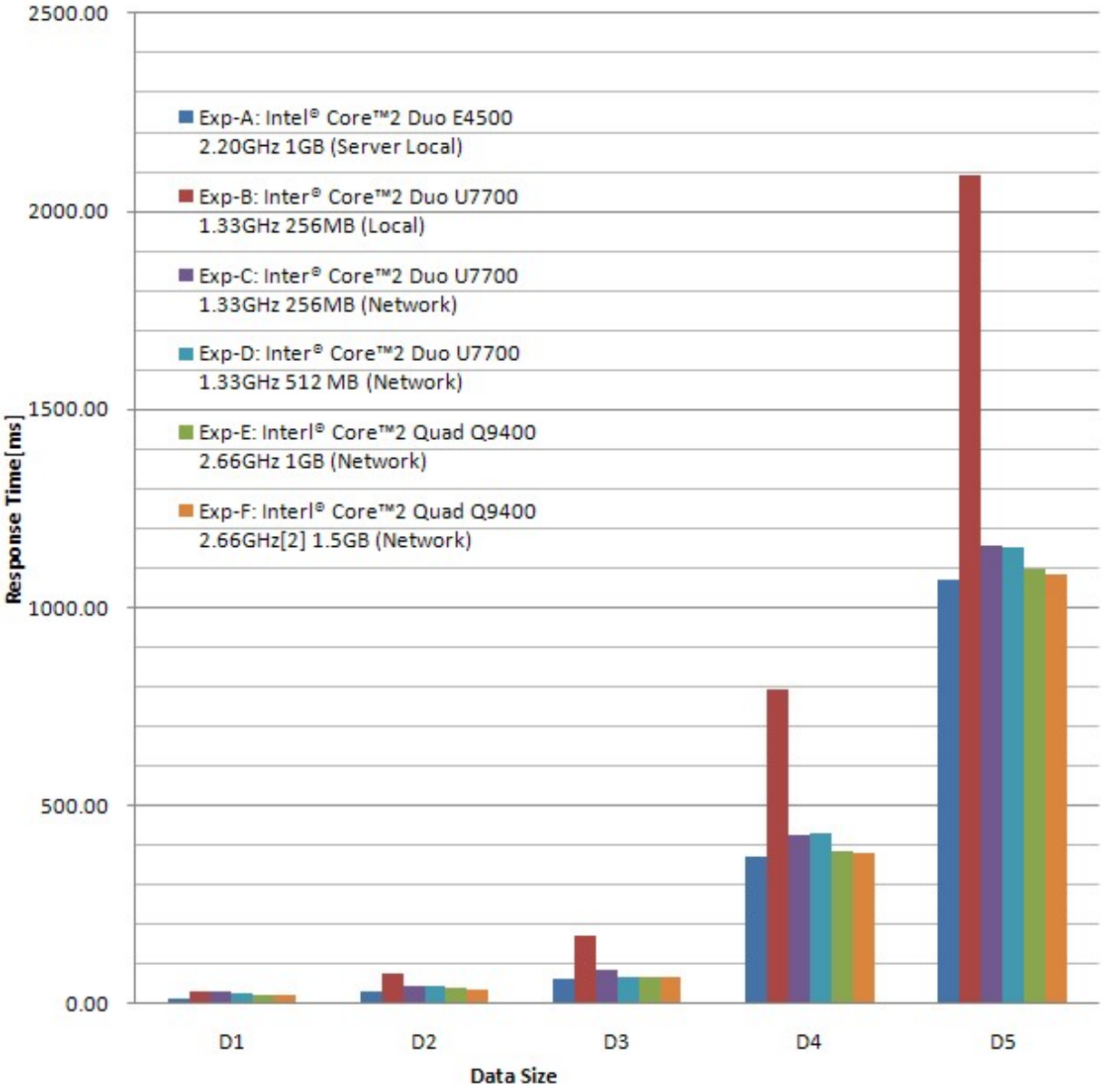

Figure 8. Response Time to a Different Size of Data on Different Platforms