Vision-Based Continuous Graffiti-like Text Entry System

İ. Aykut Erdem[1], M. Erkut Erdem[1], Volkan Atalay[1], A. Enis Çetin[2]

[1]Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey

[2]Dept. of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

ABSTRACT

It is now possible to design real-time, low-cost computer vision systems even in personal computers due to the recent advances in electronics and computer industry. Due to this reason, it is feasible to develop computer vision based human-computer interaction systems. In this paper, a vision-based continuous Grafiti^{TM}-like text entry system is presented. The user sketches characters in a Grafiti^{TM}-like alphabet in a continuous manner on a flat surface using a laser pointer. The beam of the laser pointer is tracked on the image sequences captured by a camera and the corresponding written word is recognized from the extracted trace of the laser beam.

Keywords: Computer vision, human-computer interaction, wearable computing, graffiti recognition, tracking

1. INTRODUCTION

In this paper, we address the problem of entering ASCII text to a wearable computer or a mobile communication device. Mobile communication and computing devices currently have tiny keyboards which are not easy to use. Furthermore such keyboards occupy a large part of the screen in the tablet computers and the touch screen systems. Computer vision may provide alternative, flexible and versatile ways for humans to communicate with the computers. In this approach, the key idea is to monitor the actions of the user by a camera and interpret

Further author information: (Send correspondence to İ.A.E.)

İ.A.E.: E-mail: aykut@ceng.metu.edu.tr, Telephone: +90 312 210 5552, Fax: +90 312 210 1259

M.E.E.: E-mail: erkut@ceng.metu.edu.tr, Telephone: +90 312 210 5552, Fax: +90 312 210 1259

V.A.: E-mail: volkan@ceng.metu.edu.tr, Telephone: +90 312 210 5576, Fax: +90 312 210 1259

A.E.Ç.: E-mail: cetin@ee.bilkent.edu.tr, Telephone: +90 312 290 1477, Fax: +90 312 266 4126

them in real-time. For example, character recognition techniques developed in document analysis[1–3] can be used to recognize hand-writing or sketching. In a previous study by Ozer *et al.*[1] , a vision based system for recognizing isolated characters is developed where users draw by a pointer or a stylus on a flat surface or the forearm of a person. The user's actions are captured by a head mounted camera. To achieve very high recognition rates characters are restricted to a single stroke alphabet like Graffiti^{TM}alphabet.

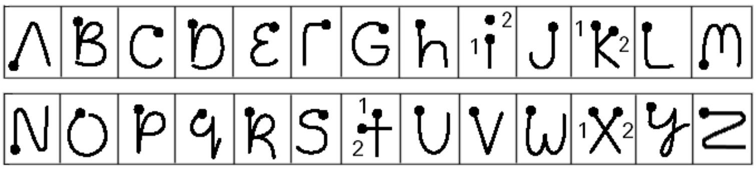
In this paper, we develop Vision-Based Continuous Graffiti^{TM}-like Text Entry System as an extension of.[1] In this system, instead of drawing isolated characters the user sketches the Graffiti^{TM} alphabet in a continuous manner on his or her left arm or on a flat surface using a pointer or a stylus or a finger. In this approach, the alphabet is also based on Graffiti^{TM} alphabet. However, some letters of the Graffiti^{TM} alphabet have to be modified to increase the recognition accuracy. By restricting the alphabet to Graffiti^{TM}-like characters, very high recognition rates can be achieved.

The proposed continuous Graffiti^{TM} recognition system can be incorporated into a presentation system as well. In many large auditoriums, the computer containing the presentation material is not on the stage. It is usually very difficult for the speaker to jump to previous or future slides or to extract another document from the computer. The user can mark some keywords or some slides before the presentation. During the presentation he or she can write the keyword on the screen using the laser pointer and then the system brings the pre-marked slide or the requested document to the screen.

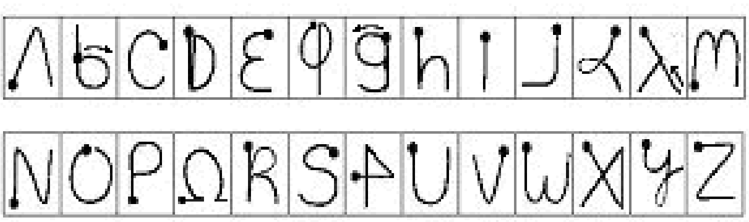
The organization of the paper is as follows: In Section II, the basics of the overall text entry system is presented. The details of tracking and recognition phases are described in Section III and IV, respectively. The experimental results are given in Section V. The paper concludes with Section VI in which the presented study is discussed and the future work is stated.

2. VISION-BASED CONTINUOUS GRAFFITI^{TM}-LIKE TEXT ENTRY SYSTEM

Unistroke isolated character recognition systems are successfully used in personal digital assistants in which people feel easier to write rather than type on a small size keyboard[4, 5] . In this approach, it is assumed that each character is drawn by a single stroke as an isolated character. One of the alphabets that has this property is the Graffiti^{TM}. In a study by Ozer *et al.*[1] , a vision-based system for recognizing isolated Graffiti^{TM} characters is proposed. In this system, the user draws characters by a pointer or a stylus on a flat surface or the forearm of a person. In our study, we extend the work, isolated Graffiti^{TM} recognition problem, to continuous Graffiti^{TM} recognition. To increase the recognition accuracy of the system, we have modified the original Graffiti^{TM} alphabet. The original Graffiti^{TM} and our modified alphabets can be seen in Figure 1(a) and (b), respectively.



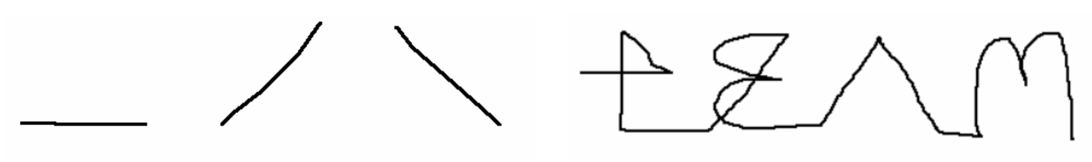
(a)



(b)

Figure 1. (a) Original Graffiti^{TM} alphabet, (b) Modified alphabet. Heavy dot indicates starting point.

In this handwriting method, the transitions from a character to another are also restricted to the three possible strokes shown in Figure 2(a). Transition from one character to another can be done with a horizontal line segment, a monotonically increasing convex curve or a monotonically decreasing convex curve. An example word "team" is written in continuous Graffiti^{TM} in Figure 2(b).



(a) (b)

Figure 2. (a) Character to character transition strokes, (b) Word "team" written in continuous Graffiti^{TM}-like alphabet.

In the current system, the user writes in continuous Graffiti^{TM} using a laser pointer on the forearm captured

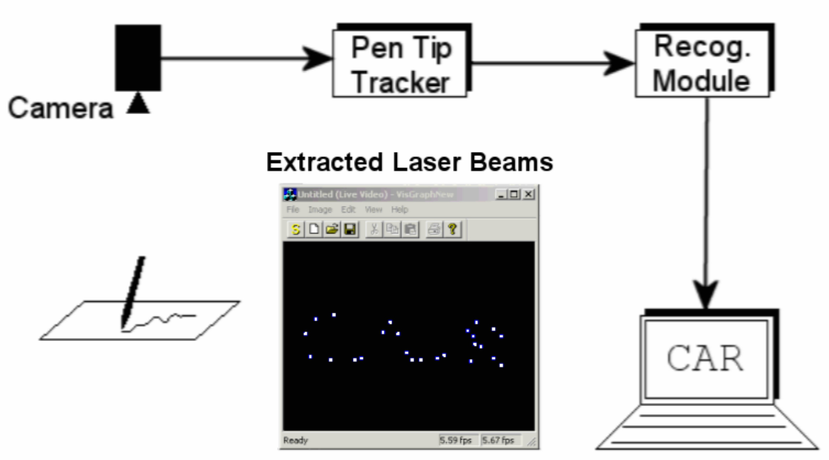


Figure 3. Overall system architecture of Vision-Based Continuous Graffiti^{TM}-like Text Entry System.

by a camera mounted on the forehead or a shirt pocket. The video is segmented to image sequences corresponding to each written word. The image sequence starts with a laser pointer turn-on action and terminates when the user turns off the laser pointer. In each image in this sequence, the beam of the laser pointer is located by the tracker module and after obtaining these sample points, the recognition module outputs the recognized word. As the overall system architecture is shown in Figure 3, the system is composed of tracking and recognition phases.

The advantages of our vision-based text entry system compared to other vision-based systems[6–8] are the following:

- The background is controlled by the forearm of the user. Furthermore, if the user wears a unicolor fabric then the tip of the finger or the beam of the pointer can be detected in each image of the video by a simple image processing operation such as thresholding.

- It is very easy to learn a Graffiti^{TM}-like alphabet. Only a few characters are different from the regular Latin alphabet. Although it may be easy to learn other text entry systems such as[6, 7, 9] , some people are reluctant to spend a few hours to learn unconventional text entry systems. Furthermore, in addition to the regular characters other single stroke characters can be defined by the user to be used as bookmarks, pointers to databases etc.

- Computationally efficient, low power consuming algorithms exist for the recognition of unistroke characters and they can be implemented in real time with very high recognition accuracy. After a few minutes of studying the Graffiti^{TM}-like alphabet, recognition accuracy is very high compared to the regular handwriting recognition method developed by Fink *et al.*[8] .

- Computer vision-based text entry systems are almost weightless.

3. TRACKING

The beam of the laser pointer is located by detecting the moving pixels in the current image of the video and from the color information. Moving pixels are estimated by taking the image difference of two consecutive image frames. Then by using the fact that the beam of the laser pointer is brighter than its neighbor pixels, the tracking process can be performed in a robust way. By calculating the center of the mass of the bright red pixels among the moving pixels, the position of the beam of the laser pointer is determined. The overall process is shown in Algorithm 1.

Algorithm 1 Finding the position of the beam of the laser pointer.

Given two consecutive camera images I_j and I_{j-1} and lines l_{AB}, l_{BC} and l_{CD},

(i) Determine the binary difference image I_{diff} between I_j and I_{j-1}.

(ii) By masking I_{diff} over I_j, form the image I_{mask}.

(iii) Determine the maximum intensity value i_{max} over the pixels in I_{mask}.

(iv) Set the intensity threshold t to $0.9 \times i_{max}$.

(v) For all pixels p_j where $i_{p_j} > t$, calculate the position of the beam of the laser pointer by taking center of mass as follows:

$$c_x = \frac{1}{n}\sum_{j=0}^{n} p_{j_x}, \quad c_y = \frac{1}{n}\sum_{j=0}^{n} p_{j_y}$$

4. RECOGNITION

As shown in Figure 4, the position of the pen tip and pen up/down information extracted in tracking phase is applied as an input to the recognition system. First, the chain code is extracted from the relative motion

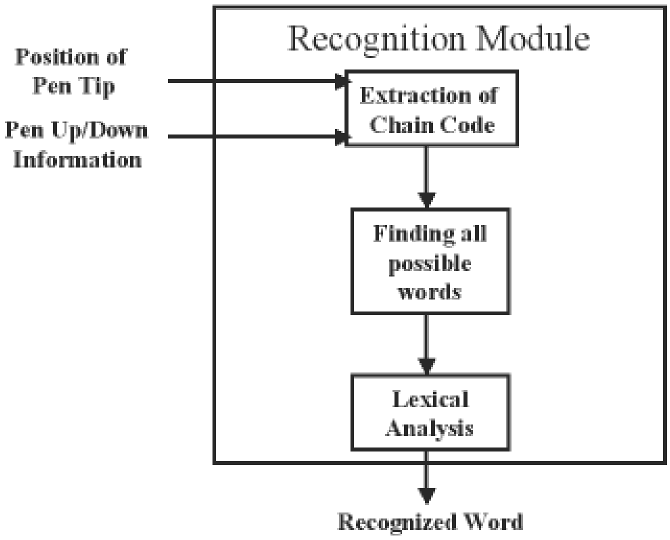


Figure 4. The inner structure of Recognition Module.

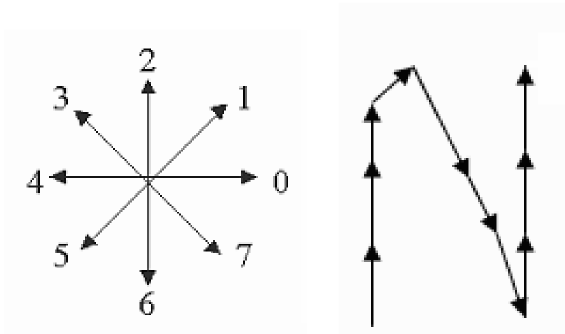
of the beam of the laser pointer between consecutive camera images. Then, the extracted chain code of the word is analyzed and all possible words conforming the extracted chain code are determined. At the end by performing a lexical analysis, the recognized word(s) are displayed on the screen.

4.1. Extraction of chain code

In our system, the unistroke characters are described using a chain code which is a sequence of numbers between 0 and 7 obtained from the quantized angle of the beam of the laser pointer in an equally time sampled manner as shown in Figure 5(a). Chain coded representation of characters are generated according to the angle between two consecutive positions of the beam of the laser pointer. A sample chain coded representation of the word “N” is shown in Figure 5(b).

4.2. Finding all possible words

Each character in the alphabet and transition strokes are all represented by a distinct finite state machine (FSM) (see Table 1). If we have an extracted chain code of a character, we can recognize that character by examining it according to the FSMs representing each character in the alphabet. As an example, in Figure 5(b), the character “N” is characterized by the chain code [2,2,2,1,7,7,6,2,2,2] where the finite state machine for the character “N” is shown in Figure 6. The first three inputs, 2,2 and 2, do not produce any



(a) (b)

Figure 5. (a) Chain code values for the angles, (b) A sample chain coded representation of the character "N" is [2,2,2,1,7,7,6,2,2,2].

error when applied to the first state of the FSM representing the character "N". The forth number of the chain code, 1, leads to an error and an increase in the error counter by 1. The next input 7 makes the FSM to go to the next state and the subsequent 7, and 6 let the machine to remain there. Whenever the input becomes 2, the FSM moves to the third state. The machine stays in this state until the end of the chaincode and the FSM terminates with an error value of 1. When we extend this analysis over all FSMs, we come up with the character recognition algorithm shown in Algorithm 2.

As it can be observed from Table 1, FSMs are different for each character in the alphabet. However, for some extracted chain codes of the written characters, some FSMs can output close error counts. For example, for the input chain code [6,6,4], while the FSM for character "J" outputs an error count 0, the FSM for character "I" outputs an error count 1. This may generate a confusion between characters "J"

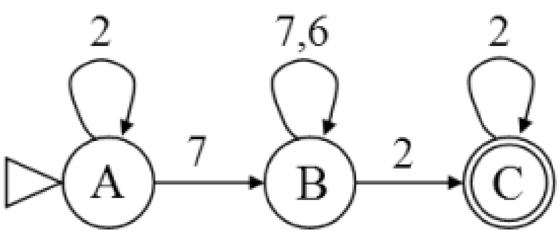


Figure 6. Finite State Machine for the character "N".

Algorithm 2 Character Recognition Algorithm based on analysis using FSMs.

Given the extracted chain code of a character

(i) The chain code is applied as input to all FSMs representing each character.

(ii) State changes are determined and additionally an error counter is increased by one if a change is not possible according to the current FSM.

(iii) If a chain code does not terminate in the final state, the corresponding character is eliminated.

(iv) Errors in each state are added up to find the final error for each character.

(v) Character with the minimum error is the recognized one.

---

and "I". Similarly, while writing the character "E", the FSM for the character "G" outputs a low error count. This is also the case for the characters "J" and "I", "S" and "C", "U" and "W", "W" and "Y", and "X"and "L". The main reason for these confusions is that the FSMs are constructed to be tolerant enough to different user writings for alternative chaincodes. However, this is corrected by introducing a lexical analysis step at the end.

It is preferred that a word can be segmented into characters by examining the transition strokes. But in general, this may not be possible since these detected transition strokes can also be a sub-stroke of a character. Therefore, our recognition module works in a recursive manner and outputs all possible words of the extracted chain code. As described before, each FSM representing a character returns an error value, the ones having minimum errors are selected and for each one, the next chaincode inputs will be passed to all the FSM's for the next character. This process continues until the end of chaincode is reached. The segmentation problem can also be solved at the lexical analysis step similar to the confusion issue discussed in the previous paragraph.

It is observed that the FSM based recognition algorithm is robust as long as the user does not move his arm or the camera during the writing process of a letter. Characters can be also modeled by Hidden Markov Models which are stochastic FSM's instead of the deterministic FSM's to further increase the robustness of the system at the expense of higher computational cost. In addition, in order to prevent noisy state

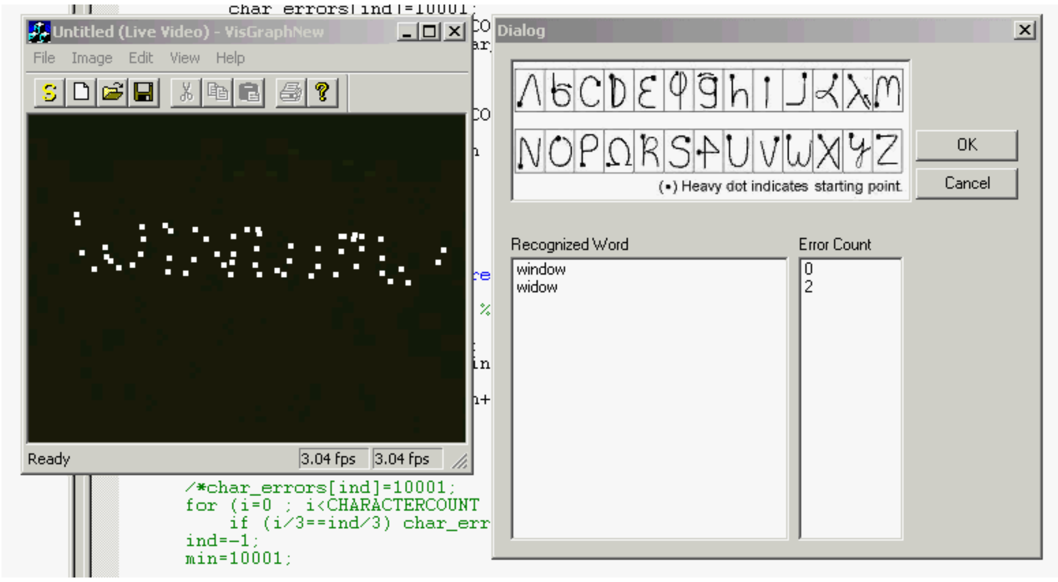


Figure 7. The result of lexical analysis for the written word “window”.

changes, look-ahead tokens can be used which acts as a smoothing filter on the chain code.

4.3. Lexical Analysis

At the end of the step described in Section 4.2, a list of all possible words is obtained. In lexical analysis step, the meaningless words are eliminated by looking up a 18000 word dictionary which is composed of most common English words. At the end, only the words found in the dictionary are displayed as the recognized ones in sorted order according to their total error count. This can be seen in Figure 7.

5. EXPERIMENTAL RESULTS

In our experiments, we have a computer with Intel Pentium IV-1.7Ghz processor with 512 GB memory, a webcam producing 320x240 pixels color images at 13.3 fps and an ordinary laser pointer. The user draws continuous Graffiti^{TM} characters using the laser pointer on the dark background material. In Graffiti^{TM}-like recognition systems, very high recognition rates are possible[5] .

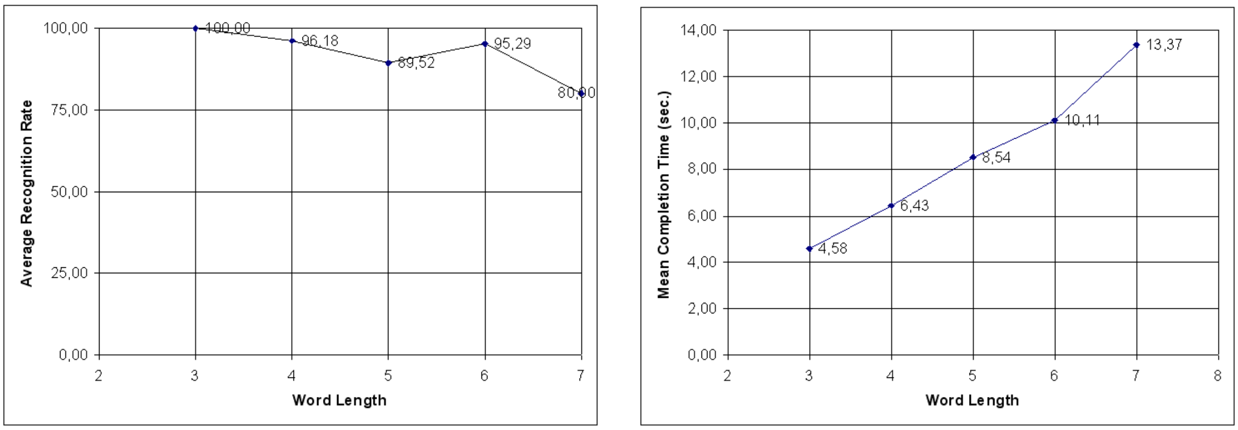
To examine the performance of our system, the system is tested with a word dataset consisting of 30 words in various lengths and these words are written at least 15 times by different people. In our system, in spite of the existence of perspective distortion, it is possible to attain a recognition rate of 93% at the word level. The words in the test set and corresponding recognition rates are listed in Table 2. Additionally, according to experiments the average writing speed is calculated as 8 words per minute (wpm). Actually, there is a trade-off between writing speed and the recognition rate. Since the whole process depends on the CPU power of the computer and the frame rate of the webcam, if the user writes quickly, the extracted chaincode may not be fully correct due to the frame losses and consequently this directly affects the recognition. Due to this trade-off, the size of written characters and therefore the written word must be big enough. In this case, only 2-3 words can be written in the viewing area of the camera. However, we believe that the effect of this trade-off can be minimized with the improvements on current hardware.

In addition, when we examine recognition rate versus word length graph shown in Figure 8(a), we can infer that although the word length has an importance, the recognition rate is not directly related with word length. Furthermore, the mean completion time of the written word versus word length graph which is given in Figure 8(b) shows that the writing time increases linearly with the increase in word length.

It is also observed that the recognition process is writer independent with little training and we believe that we can achieve higher writing speed rates with the advances in digital camera and wearable computer technology. The perspective distortion plays some role in the recognition accuracy of the system. In our experiments, we have observed that the degradation in recognition is at most 10% around 30 degree difference between the plane on the which writing is performed and the camera. Several tests are also carried out under different lighting conditions. In day/incandescent/fluorescent light, the average intensity of the background is about 50/180/100 whereas the intensity value of the beam of the laser pointer is about 240/250/240. In all cases, the beam of the laser pointer can be easily identified from the dark background.

6. CONCLUSION

In this study, we present a vision-based continuous Graffiti^{TM}-like text entry system. A Graffiti^{TM}-like alphabet is developed where the users can write characters in a continuous manner on a flat surface using the laser pointer. This alphabet can be easily extended by defining finite state machines for each newly added character. The video



(a) (b)

Figure 8. (a) Recognition rate versus word length. (b) Mean completion time versus word length.

is segmented to image sequences corresponding to each written word. Every image sequence starts with a laser pointer turn-on action and ends when the user turns off the laser pointer. In each image in this sequence, the beam of the laser pointer is tracked and the written word is recognized from the extracted trace of the laser beam. Recognition is based on finite state machine representations of characters in the alphabet.

According to the experiments, the recognition rate of Vision-Based Graffiti^{TM}-like Text Entry System is measured as 93% at the word level and the writing speed as around 8 wpm. It is also observed that the system is writer-independent and requires little training for learning the alphabet. Also, the writing time increases linearly with the increase in word length.

Since we use the laser pointer as the pointing device, tracking the beam in real-time is not a complicated process. As future work, possibility of using some other pointing devices (e.g. finger, an ordinary pen, etc.) can be investigated. But this time to track the tips of these pointers, some complex feature trackers (e.g. Kanade-Lucas-Tomasi (KLT) point-based feature tracker[10]) in a combination with a Kalman filter[11] can be used.

REFERENCES

1. O. F. Ozer, O. Ozun, V. Atalay, and A. E. Cetin, "Visgraph: Vision based single stroke character recognition for wearable computing," *IEEE Intelligent Systems and Applications* **16**, pp. 33–37, May-June 2001.

2. O. Gerek, A. Cetin, A. Tewfik, and V. Atalay, "Subband domain coding of binary textual images for document archiving," *IEEE Transactions on Image Processing* **8**, pp. 1438–1446, October 1999.
3. M. Munich and P. Perona, "Visual input for pen-based computers," in *Proceedings of 13th Internation Conference on Pattern Recognition*, pp. 33–37, 1996.
4. D. Goldberg and C. Richardson, "Touch-typing with a stylus," in *Proceedings of the INTERCHI'93 Conference on Human Factors in Computing Systems*, pp. 80–87, 1993.
5. I. MacKenzie and S. Zhang, "The immediate usability of graffiti," in *Proceedings of Graphics Interface'97*, pp. 129–137, (Toronto), 1997.
6. J. A. R. A. Vardy and L.-T. Cheng, "The wristcam as input device," in *Proceedings of the Third International Symposium on Wearable Computers*, pp. 199–202, October 1999.
7. T. Starner, J. Weaver, and A. Pentland, "A wearable computing based american sign language recognizer," in *Proceedings of the 1st International Symposium on Wearable Computers*, October 1997.
8. G. A. Fink, M. Wienecke, and G. Sagerer, "Video-based on-line handwriting recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 226–230, IEEE, 2001.
9. "http://www.handykey.com as accessed on," July 2003.
10. C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep. CMU-CS-91132, Carnegie Mellon University School of Computer Science, Pittsburgh, 1991.
11. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering* **82**(Series D), pp. 35–45, 1960.

Table 1. The table of FSMs for each character in the alphabet.

Character	*Corresponding FSM*	*Character*	*Corresponding FSM*
A	12 76	N	12 7 12
B	107 654 32	O	45 67 012 34
C	345 67	P	2 01 76 54
D	67 12 076 54	Q	234 10 765
E	345 670 345 670	R	6 12 076 54 076
F	456 07 12 345 6	S	345 607 543
G	345 67 012 654 3210	T	0 23 56
H	6 6 210 76	U	6 071 2
I	6 6	V	65 32
J	6 54	W	67 012 670 12
K	65 4321 07	X	7 12 56
L	7 23 5	Y	67 012 654 321
M	21 67 210 76	Z	0 5 0

Table 2. The words in the test set and corresponding recognition rates.

Word	*Recognition Rate(%)*	*Word*	*Recognition Rate(%)*
she	100.00	agree	94.44
car	100.00	queue	93.75
tin	100.00	three	90.00
road	100.00	money	92.00
kind	90.00	model	84.00
bird	100.00	future	95.00
them	89.47	vision	85.00
word	100.00	window	100.00
book	93.75	liquid	100.00
sand	100.00	engine	100.00
jazz	93.75	desire	100.00
nine	100.00	problem	75.00
find	93.75	science	80.00
twin	100.00	subject	80.00
crazy	8.005	lexical	90.00